IAENG International Journal of Applied Mathematics

L21 Norm Regularization ELM with p-Huber Loss Function for Multi-target Regression

Wenhao Sun, Kuaini Wang*, Mingming Wang and Qiang Lin

Abstract—**Extreme Learning Machine (ELM), as a fast and easy to implement model, has been widely used in multiple fields in recent years. In practice, it has been found that ELM has limited application in the field of multi-target data problems. Moreover, ELM lacks resistance to outliers, which may lead to a lack of robustness in multi-target problems. In response to the above issues, this article proposes an improved methods to enhance the robustness of multi-target ELM from the perspective of L21 norm regularization and loss function. The L21 norm regularization can adaptively eliminate redundant neurons in ELM, reduce the complexity of model learning, and thus achieve optimization of ELM. Based on the L21 norm regularization, a p-Huber loss function is introduced to multi-target robust ELM model, referred to as L21pHELM. The p-Huber loss function can control the impact of outliers on the model through the parameter p, thereby improving the robustness of the model. To better validate the effectiveness of the proposed L21pHELM in terms of model robustness and generalization performance, we conducted experiments on both artificial datasets and 14 benchmark datasets using an iteratively reweighted algorithm, comparing it with four other algorithms. The evaluation was performed under different outlier conditions using two metrics: aRRMSE and average rank. The experimental results demonstrate that L21pHELM achieves relatively smaller aRRMSE values, particularly in high-outlier scenarios, indicating stronger noise resistance capabilities.**

Index Terms—**Multi-target, Extreme Learning Machine, L21 Norm Regularization, p-Huber Loss, Robustness.**

I. Introduction

MULTI-target regression [1], [2] is a machine learning task where the model needs to predict the values of multi-target variables simultaneously. In this type of problem, each sample has multi-target variables associated with it, typically representing properties or features of different aspects in the system[3]. The multi-target regression task requires the model to consider the relationships between target variables, which may have positive, negative, or nonlinear relationships. The choice of model depends on the nature of the data and the complex relationship between the target variables [4]. Feature selection is equally important in multi-target regression, requiring selecting appropriate features, performing normalization, and handling missing values. In addition, to solve the problem of multi-target regression, researchers have proposed various algorithm improvements and techniques. Reference [5] solves the regression problem of multi-target data by using different penalty functions for error values falling in different intervals in the loss function. Researchers continuously strive to improve the model to handle the problems in multi-target regression tasks better. In this field, it is not only necessary to focus on the prediction accuracy of a single target but also to consider the synergistic effects between multiple targets to improve the performance of the model in complex tasks.

Multi-target regression involves developing predictive models for problems with multiple continuous targets. One challenge in constructing a multi-target model lies in capturing the relationships between multi-target variables during the training process, as these relationships are intertwined with the input and target variables of the training set. Support Vector Regression Correlation Chain [6] addresses this challenge by establishing a maximum correlation chain to capture the optimal correlations between target variables, thereby enhancing the predictive performance of the model. This method introduces a correlation regression subchain within the framework of Support Vector Regression to address the complexities of multi-target regression. By leveraging the correlations between targets, the correlation regression chain enhances the regression performance, consequently improving the accuracy of multi-target regression.

The L1 norm is introduced into the Outlier Robust ELM(OR-ELM) [7], which is applied to a single-target. The Generalized Outlier Robust ELM (GOR-ELM) [8] proposes to extend OR-ELM to handle multi-target regression problems. GOR-ELM extends OR-ELM to multi-target regression problems, using the model proposed in GOR-ELM, replacing the F norm with the L21 norm of prediction error, which can be explained as an extended form of L1 norm. When there is only one-dimensional target and only ridge regression is used, this method is the same as OR-ELM. GOR-ELM uses the alternating direction multiplier method [9] to solve optimization problems. The alternating direction multiplier method is an efficient optimization method mainly used to solve separable convex optimization problems. It performs well in processing speed and convergence performance, and is therefore widely used in fields such as statistical learning and machine learning. Multi-target regression is widely used in practical applications, including financial prediction [10], mechanical design [11], and other fields.

Manuscript received January 22, 2025; revised July 18, 2025.

The work was supported by the National Natural Science Foundation of China under Grant No. 61907033, and the Postgraduate Innovation Fund project of Xi'an Shiyou University under Grant No. YCX2413142.

Wenhao Sun is a postgraduate student at the School of Computer Science, Xi'an Shiyou University, Shaanxi 710065, P. R. China (e-mail: sunwenhao1023@163.com).

Kuaini Wang is an associate professor at the School of Science, Xi'an Shiyou University, Shaanxi 710065, P. R. China (Corresponding author, e-mail:wangkuaini1219@sina.com).

Mingming Wang is a postgraduate student at the School of Computer Science, Xi'an Shiyou University, Shaanxi 710065, P. R. China (e-mail: ming126916@163.com).

Qiang Lin is a lecturer at the School of Business, Jiangnan University, Wuxi 214122, China (e-mail:8202208015@jiangnan.edu.cn).

Whether it is a ELM model for multi-target data or single-target data, the objective function comprises a regularization and a loss term. Below is the development history of the regularization term and the loss function.

A. Regularization

While traditional ELM may exhibit small training errors, the primary objective extends beyond merely minimizing training error. The ultimate aim is to ensure that the model accurately predicts new samples during the testing phase, thereby minimizing testing error. Regularizing target weights serves the purpose of preventing the model from overfitting to the training data and controlling model complexity. The determination of target layer weights is influenced by factors such as the number of hidden nodes in the input layer and various parameters. Excessive model parameters can lead to increased model complexity, rendering it prone to overfitting [12]. Hence,adjusting model complexity can effectively reduce test errors, and model complexity is usually controlled by regularization terms, which are typically implemented through various norm forms.

L2 norm regularization [13] imposes constraints on all target weights, potentially making it difficult to distinguish target weights representing distinct model features. This heightened sensitivity could negatively impact the model's performance. By amplifying the influence of irrelevant features and introducing noise, the model may struggle to extract meaningful feature-related information [14]. Consequently, some researchers have proposed enhanced regularization techniques, such as L1 norm regularization ,also known as Lasso regularization [13], to improve the model's generalization performance. An L1 norm regularization-based ELM [15] combines L1 norm with ELM and utilizes the Newton iteration method for optimization. This approach tends to drive certain fitting weights of the model towards zero, resulting in a sparse model that is more interpretable. Experimental results demonstrate that, compared to conventional ELM, L1 norm regularization-based ELM can achieve comparable performance with fewer hidden layer nodes.

In contrast to L2 norm regularization, L1 norm regularization constrains the model by incorporating a penalty term that represents the sum of the absolute values of the target weight parameters. This regularization technique facilitates setting certain target weight values to zero, thereby effectively reducing the impact of specific features on model predictions [16]. However, when dealing with highly correlated features, L1 norm regularization may encounter non-uniqueness issues. These challenges have spurred a demand for regularization methods with enhanced robustness and flexibility. To address these challenges, the elastic net [17] introduces a novel hyperparameter to balance the penalty terms of L1 and L2 norms. This enables the elastic net to simultaneously select important features and handle scenarios involving highly correlated features. As a result, the elastic net provides a more stable solution, effectively mitigating the issues arising from correlated features.

The L21 norm regularization is a novel regularization technique that combines the characteristics of both L2 norm and L1 norm regularization, similar to elastic networks. This combination aims to strike a balance between promoting smoothness and facilitating feature selection [18], [19], [20], thereby effectively addressing the challenges posed by complex data scenarios [21]. In 2018, Rui et al. proposed a robust ELM model (L21ELM) that simultaneously applies the L21 norm to both loss functions and regularization [22]. Additionally, the Sparse Multi-target ELM [23] incorporates L21 norm regularization to enhance the sparsity of model parameters when solving multi-target regression problems. This regularization technique helps automatically select input features that are strongly correlated with the target variable, thus improving the model's generalization performance. Moreover, L21 norm-based regularization is also used for feature selection in high-dimensional data processing. By penalizing the L21 norm of features, this approach enables the model to identify and select the most informative features, thereby enhancing the model's robustness and generalization capabilities.

The online sequential ELM with L21 norm regularization [24] aims to improve performance in online sequential modeling tasks that involve processing sequential data. This method helps handle noise and redundant information inherent in online sequential data. By integrating L21 norm regularization, these methods continuously adjust the sparsity of model parameters and the smoothness of weights, thereby enhancing the performance of the ELM model in various tasks. In grouped variable regression [25], the model selection and estimation algorithm incorporates L21 norm regularization as part of the Group Lasso method for model selection and evaluation. This approach introduces regularization of feature groups during variable selection, enabling the simultaneous selection of relevant features. Zhang et al. [26] proposed a technique for inducing weight sparsity across multiple tasks using L21 norm regularization, aiming to achieve feature selection within the framework of multi-task learning. Similarly, multi-task feature learning [23] employs L21 norm regularization as a sparse representation technique, providing an overview of a sparse representation method. These methods are applied across various domains, such as signal processing, image analysis, and pattern recognition, significantly improving the ability to handle high-dimensional data effectively. The diverse applications of L21 norm regularization in ELM methods highlight its flexibility and efficacy, offering an efficient solution for addressing a wide range of complex problems.

B. Loss functions

In ELM, due to the simplicity and directness of the L2 loss, it is commonly chosen as a performance metric [27]. This metric quantifies the average squared difference between predicted and true values, guiding the model to minimize the overall prediction error. However, the L2 loss is highly sensitive to outliers and noise, which may compromise the model's robustness in complex scenarios. On the other hand, the L1 loss [28] measures the absolute difference between predicted values and true values. The use of absolute errors provides robustness against outliers and helps mitigate the impact of errors. Nevertheless, the L1 loss function is non-smooth and non-differentiable at zero, requiring additional processing when used in optimization algorithms. The L21ELM [22] adopts a loss function based on the L21 norm. Compared to the L2 loss, the L21 norm loss function can reduce the improper influence of outliers in data points, thereby enhancing the robustness and stability of the learning process.

The correntropy loss function [29] achieves robustness to outliers and noise by capturing statistical correlations between samples. Outliers within the sample can lead to amplified errors, resulting in poor generalization performance of the model. The robust one-class ELM based on correntropy and kernel learning [30] addresses one-class problems using elastic networks and correntropy loss functions, demonstrating the model's capability in boundary construction and noise resistance. By minimizing the correntropy loss, the model can better capture the nonlinear relationship between input and target variables, thereby enhancing its robustness to noise and outliers. Furthermore, a robust extreme learning machine based on truncating the maximum correntropy criterion loss function [31] improves upon the correntropy loss function by truncating the Maximum Correntropy Criterion (MCC). This truncated loss function limits the maximum correntropy loss to a constant, effectively suppressing the influence of noise and outliers on the model. Additionally, ELM introduces a regularization correntropy criterion suitable for extreme learning machines [32], which is based on the regularization correntropy criterion. This enhancement aims to improve the noise resistance of ELM. The introduction of correntropy loss enhances the model's ability to capture complex data distributions and relationships between labels.

The L2 loss function is commonly employed for evaluating regression performance in regression problems. However, regression algorithms based on L2 loss suffer from a notable drawback: their optimality heavily relies on the Gaussian assumption. In practical applications, data often exhibits non-Gaussian noise or outliers, and using a loss function that lacks robustness to outliers can compromise the entire statistical analysis. Reference [33] introduces a robust regression technique called Regression Robust Extreme Learning Machine, which leverages the differentiability, non-convexity, and boundedness of the exponential Laplace loss function. This approach significantly enhances the robustness of the ELM model. Similarly, Reference [34] proposes a Robust Regularized Extreme Learning Machine based on non-convex loss functions. By replacing the L2 loss function with a non-convex alternative, this method imposes a constant penalty for larger outliers, thereby mitigating their negative impact. Additionally, the optimization problem of the model is solved using difference of convex functions programming.

The Huber loss function [35] stands out as a smooth loss function that bridges the gap between L1 and L2 loss, offering robustness against outliers. By blending the strengths of both L1 and L2 loss functions, Huber loss proves to be more resilient to outliers compared to L2 loss while remaining differentiable at the center point. The robust regularization ELM for iterative reweighted least squares regression [36] adopts four different loss functions (L1 norm, Huber, Bissquare, and Welsch) within the ELM model. This approach also incorporates L1 and L2 regularization strategies to mitigate overfitting. Huber loss, in particular, strikes a balance between L2 and L1 loss, adjusting its behavior based on the error magnitude. It behaves similarly to L2 loss when the error is small and transitions to L1 loss when the error is large.

The p-Huber loss function [37] is an extension of the Huber loss function that allows sensitivity to outliers to be adjusted through the parameter p. It combines the L2 loss function and the L1 loss function, controlled by two parameters. When the residual is less than the given parameter, the computationally convenient L2 loss is still used. After comparing the robustness and fitting effects of the L1 loss function, Huber loss function, and correntropy loss function, it was verified that the p-Huber loss function is more robust to outliers and can better fit data [38]. Therefore, the p-Huber loss function is more robust in handling outliers and can reduce the impact of outliers on the model, thereby improving the stability and predictive performance of the model.

The remainder of this paper is organized as follows. Section 2 presents related work concerning the symbols and background of ELM. In Section 3, we describe the L21 norm regularization ELM with the p-Huber loss function for multi-target regression and provide the algorithmic process. Section 4 reports the experimental results by comparing our model with several other ELM models on 14 benchmark datasets and 1 artificial dataset. Finally, Section 5 draws conclusions.

II. RELATED WORKS

A. Notations and definitions

This section provides an overview of the symbols and concepts utilized. The L$_p$-norm of a vector v belonging to the real number set R^n is expressed as $\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p\right)^{1/p}$. In the case of a matrix $M \in R^{m\times n}$, and apply it to a well-known vector norm. For example, if we employ the L$_p$-norm for a vector, then can obtain the L$_p$-norm of the matrix M.

$$\|M\|_p = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |m_{ij}|^p\right)^{1/p} \tag{1}$$

The L$_{21}$-norm of a matrix, which is also called the rotational invariant L$_1$-norm and used in various applications [12-15], which is defined as:

$$\|M\|_{21} = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} {m_{ij}}^2\right)^{1/2} \tag{2}$$

B. Brief introduction of extreme learning machine

ELM [2] is a single-hidden-layer feedforward neural network, in which the parameters of the hidden layer are randomly initialized and remain unchanged without iterative adjustment. Given a multi-target regression problem involving N training samples$\left\{(x_i,t_i)\,|x_i \in R^d, t_i \in R^m\right\}_{i=1}^{N}$, where each training sample with m targets. H can be written as:

$$H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_N) \end{bmatrix} \begin{bmatrix} g({w_1}^T x_1 + b_1) & g({w_2}^T x_1 + b_2) & \cdots & g({w_L}^T x_1 + b_L) \\ g({w_1}^T x_2 + b_1) & g({w_2}^T x_2 + b_2) & \cdots & g({w_L}^T x_2 + b_L) \\ \vdots & \vdots & \vdots & \vdots \\ g({w_1}^T x_N + b_1) & g({w_2}^T x_N + b_2) & \cdots & g({w_L}^T x_N + b_L) \end{bmatrix} \quad (3)$$

Within the framework of the ELM, the j-th node is defined by parameters (w_j, b_j), and the hidden layer output matrix H is represented as a function of input weights w and biases b. These parameters are randomly obtained within a specific range. The activation function $g(\cdot)$ in ELM is typically a nonlinear piecewise continuous function.

ELM aims to represent these N samples with zero error, which can be expressed in matrix form as:

$$H\beta = T \quad (4)$$

where $\beta = [\beta_1, \beta_2, ..., \beta_L]^T \in R^{L\times m}$ is the output weight matrix, and $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the weight vector connecting the i-th hidden neurons and the output neurons. T is the target matrix of training samples.

$$T = \begin{bmatrix} {t_1}^T \\ {t_2}^T \\ \vdots \\ {t_N}^T \end{bmatrix}_{N\times m} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & t_{21} & \cdots & t_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ t_{N1} & t_{N2} & \cdots & t_{Nm} \end{bmatrix} \quad (5)$$

The regularization ELM enhances stability and generalization capability by minimizing both the training errors and the norm of output weights.

$$Minimize: \mathrm{L}_{\mathrm{ELM}} = \frac{C}{2}\|T - H\beta\|_2^2 + \frac{1}{2}\|\beta\|_2^2 \quad (6)$$

where C is the regularization parameter that governs the trade-off between training error and model complexity. By setting the gradient of $\mathrm{L}_{\mathrm{ELM}}$ with respect to β to 0, the analytical expression for β is given in [2],

$$\beta = \begin{cases} H^T\left(HH^T + \dfrac{I}{C}\right)^{-1}T, & N < L \\ (H^T H + \dfrac{I}{C})^{-1}H^T T, & N \geq L \end{cases} \quad (7)$$

The dimension of the identity matrix I will change with the size relationship between N and L. When $N \geq L$, the dimension is L; otherwise, the dimension is N. The process of ELM training can be summarized into three steps: First, generate w and b randomly within a certain range, and confirm the specific activation function. Second, calculate H through the activation function, w, and b, and finally, calculate β through the above equation. ELM generates random hidden node parameters and analytically determines the output weights β, providing a stable, efficient, and simple deterministic solution.

III. L21 NORM REGULARIZATION ELM WITH p-HUBER LOSS FUNCTION FOR MULTI-TARGET REGRESSION

A. *p-Huber loss function*

The Huber loss function as follows, shown in Fig. 1.

$$\phi^{Huber}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, |y - f(x)| < \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, |y - f(x)| \geq \delta \end{cases} \quad (8)$$

In (8), this parameter is used as a trade-off between quadratic and linear loss functions. When it exceeds or equal δ, the Huber loss function adopts L1 loss; when it is less than δ, L2 loss is used. The design of Huber loss function combines the advantages of L1 and L2 loss functions, making it more anti-interference when dealing with outliers. In contrast, the impact of outliers on it is smaller than that of L2 loss function, and it is still differentiable in the central part.

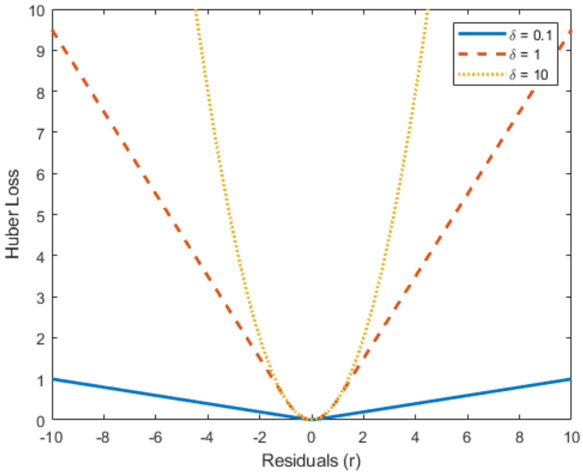


Fig. 1: Huber loss functions with different δ

On the basis of Huber loss function, p-Huber loss is proposed, which is defined as follows:

$$\phi(y, f(x)) = \begin{cases} (y - f(x))^2, |y - f(x)| < \delta, \\ \frac{2\delta^{2-p}}{p}|y - f(x)|^p - \frac{2-p}{p}\delta^2, |y - f(x)| \geq \delta \end{cases} \quad (9)$$

where $p > 0$ and $\delta > 0$. When p =1, the loss function is the same as the Huber loss function. When p =2, the loss function is L2 loss. As shown in Fig. 2 and 3, the parameters p mainly controls the concavity and convexity of the p-Huber loss function. When p <1, the image is non convex, and when p >1, the image is convex. The main focus is on studying the non-convex case when p <1. The parameters mainly control the inflection point of the loss function. The p-Huber loss function combines the advantages of two different loss functions, which are controlled by two parameters. When the error between the predicted value and the true value is less than the given parameter, the computationally convenient L2 loss is still used. Conversely, a more robust function is used. To verify the regression prediction performance of the model, the regression performance of the proposed model was compared with other ELM models on different datasets, and the experimental results were compared and analyzed to draw conclusions.

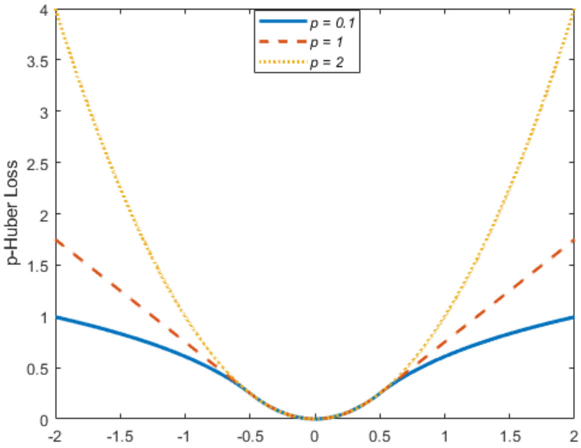


Fig. 2: Different parameters p when fixed δ =0.5

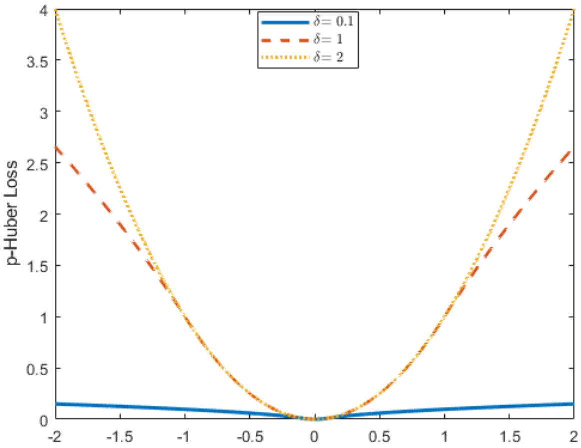


Fig. 3: Different parameters δ when fixed p =0.5

B. *Robust ELM model based on L21 norm regularization and p-Huber loss function*

A multi-target robust extreme learning machine model with L21 norm regularization and p-Huber loss function was constructed. Firstly, the model is established:

$$\begin{aligned} &\min_{\beta,\xi,p} \frac{C}{2}\sum_{i=1}^{N}\phi(\xi_i)+\|\beta\|_{2,1} \\ &s.t. \quad h(x_i)\beta = t_i^T - \xi_i^T, i=1,2,...,N \end{aligned} \tag{10}$$

Construct the Lagrange function as follows:

$$\begin{aligned} \mathrm{L}(\beta,\xi_i,\alpha_{ij}) &= \frac{C}{2}\sum_{i=1}^{N}\phi(\xi_i)+\|\beta\|_{2,1} \\ &-\sum_{\mathrm{i}=1}^{N}\sum_{j=1}^{m}\alpha_{ij}(h(x_i)\beta_{.j}-t_{ij}+\xi_{ij}) \end{aligned} \tag{11}$$

According to the optimality conditions, take partial derivatives for α_i , β_j , and ξ_i to zero respectively

$$\frac{\partial \mathrm{L}}{\partial \alpha_i} = 0 \rightarrow h(x_i)\beta - t_i^T + \xi_i^T = 0 \rightarrow H\beta - T + \xi = 0 \tag{12}$$

$$\frac{\partial \mathrm{L}}{\partial \beta_{.j}} = 0 \rightarrow D\beta_{.j} = \sum_{i=1}^{N}\alpha_{ij}h(x_i)^T \rightarrow D\beta = H^T\alpha \tag{13}$$

$$\frac{\partial \mathrm{L}}{\partial \xi_i} = 0 \rightarrow \alpha = C\phi' \tag{14}$$

where

$$\phi' = \begin{cases} \xi, |\xi| < \delta, \\ \delta^{2-p}\|\xi\|_1^{p-1}\frac{1}{\max(|\xi|,10^{-6})}, |\xi| \geq \delta \end{cases} \tag{15}$$

The optimal solution β can be calculated as

$$\beta = \begin{cases} D^{-1}H^T\left(HD^{-1}H^T+\frac{W_N{}^{-1}}{C}\right)^{-1}T, N<L, \\ (H^TW_NH+\frac{D}{C})^{-1}H^TW_NT, N \geq L \end{cases} \tag{16}$$

C. *Algorithm description and analysis*

The training steps of a multi-target extreme learning machine based on L21 norm regularization and p-Huber loss function are described in Algorithm 1. The training process uses an iterative reweighting algorithm to solve the weight matrix until it reaches the maximum number of iterations, and targets of optimal solution. Throughout the entire algorithm execution process, from the initial stage to the convergence result, the model needs to solve multiple matrix equations for each iteration. Each iteration of the algorithm is equivalent to solving an independent ELM model. The algorithm iteratively updates the two weighted matrices D and W_N in the model based on β_0 , ξ. As the target weights are continuously updated, the prediction error gradually decreases until it reaches its minimum.

IV. EXPERIMENTS AND ANALYSIS

To evaluate the effectiveness of the proposed L21pHELM, experiments are conducted on various multi-target datasets, comprising one artificial dataset and 14 benchmark datasets. The comparison includes ELM, WELM, IRWELM, and L21ELM. ELM serves as the base model, employing only the L2 loss term. IRWELM enhances robustness through an iterative reweighting algorithm. L21ELM incorporates the L21 norm into both the regularization term and the loss function of ELM, endowing the model with robustness and sparsity. Regularization parameter C used in all methods is selected from a set $\{2^i|i=-19,-18,-17,...,17,18,19,20\}$. The σ introduced in the proposed L21pHELM is from the range $\{0.1,0.2,0.3,0.4,0.5,...,1.7,1.8,1.9,2.0\}$. We set the number of hidden nodes to L = 1000. The experiments are conducted in Matlab R2021a on a system with 4GB of memory and an i5-7300 2.50-GHz processor.

To assess the efficacy of these algorithms, we utilize a commonly used regression estimation measure. Assuming that $\mathrm{t}^{(l)} = (\mathrm{t_1}^{(l)},...,\mathrm{t_N}^{(l)})^T$, where $l = 1,2,...,m$, represents the l-th target of the m targets in the actual values of testing samples,aRRMSE [29] is defined as the mean of each target's relative root mean square error (RRMSE):

$$\begin{aligned} \mathrm{aRRMSE}\left(\widehat{T},T\right) &= \frac{1}{m}\sum_{l=1}^{m}\mathrm{RRMSE}\left(\widehat{\mathrm{t}}^{(l)}-\mathrm{t}^{(l)}\right) \\ &= \frac{1}{m}\sum_{l=1}^{m}\sqrt{\frac{\sum\limits_{\mathrm{i}=1}^{\mathrm{N}}\left(\widehat{\mathrm{t_i}}^{(l)}-\mathrm{t_i}^{(l)}\right)^2}{\sum\limits_{\mathrm{i}=1}^{\mathrm{N}}\left(\bar{\mathrm{t}}^{(l)}-\mathrm{t_i}^{(l)}\right)^2}} \end{aligned} \tag{17}$$

Algorithm 1 L21pHELM

```
Input: Training set $X$ and corresponding target matrix $T$, penalty coefficient $C$, and small-scale predefined parameters $\varepsilon > 0$, $p$-Huber loss parameter $\delta$, $p$. Number of hidden layer nodes $L$.
Output: Output weight $\beta$
1: Transform the samples into the random feature space $H$ according to (5).
2: Initialize $t = 1$ and $D \in R^{L\times L}$ , $W_N \in R^{N\times N}$ as an identity matrix.
3: Repeat
4: Calculate the output weight $\beta^t$ according to (16)
5: Update $D$ as:
       $$D^{(t+1)} = diag\left\{1/(2\|\beta_1\|_2), 1/(2\|\beta_2\|_2), ..., 1/(2\|\beta_L\|_2)\right\}$$
6: Update $\xi$ as:
       $$T - H\beta = \xi$$
7: Update $D_1$ as:
       $${W_N}^{(t+1)} = diag\left\{w(\xi_1), w(\xi_2), ..., w(\xi_N)\right\}$$
   and
       $$w(\xi_i) = \min(\xi_i, \delta^{2-p}\|\xi_i\|_1^{p-1}/\max(|\xi_i|, 10^{-6}))$$
8: $t = t + 1$
9:
       $$\beta^{(t+1)} = \begin{cases} D^{-1}H^T\left(HD^{-1}H^T + \frac{{W_N}^{-1}}{C}\right)^{-1}T, N < L, \\ (H^T W_N H + \frac{D}{C})^{-1}H^T W_N T, \quad N \geq L \end{cases}$$
10: If $t > t_{max}$ or $\left\|\beta^{(t+1)} - \beta^{(t)}\right\| \leq 10^{-3}$, stop, else, go to step 4.
```

where $\widehat{\mathrm{t}}^{(l)} = (\widehat{\mathrm{t}_1}^{(l)}, \widehat{\mathrm{t}_2}^{(l)}, ..., \widehat{\mathrm{t}_\mathrm{N}}^{(l)})^T$ is the l-th targets of predicted values, and $\bar{\mathrm{t}}^{(l)} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{t_i}^{(l)}$.

A. Experiments on artificial datasets

We first conduct experiments on an artificial dataset to prove the robustness of the algorithm. Reference [39] provides the simulated datasets used in our experiments. The outlier-free dataset $S = \{(x,y) \in X \times Y \subset R \times R^3\}$, $x \in (0,10)$ is the function defined as follows:

$$\begin{aligned} &\mathrm{y1} = x^{-1}\sin x \\ &\mathrm{y2} = |x-1|/8 + |\sin(0.75 + 0.25x)\pi|/2 + 0.5 \qquad (18) \\ &\mathrm{y3} = |x-1|(1 + 10|\sin(x+1)|)/100 \end{aligned}$$

We initially generate a dataset comprising 200 samples, which are then randomly partitioned into 150 training samples and 50 testing samples. Additionally, to simulate varying degrees of outliers, we added outliers to the target values of the training samples generated in the second step. The outliers were randomly selected from the range [0, 0.1] and added to 0%, 10%, 20%, 30%, and 40% of the training samples, respectively. Notably, the testing samples were obtained from the outlier-free function to ensure consistency. To guarantee the reliability of our results, we conducted ten independent experiments for each outlier distribution.

TABLE I: aRRMSE under different levels of outliers in 5 models

Level	ELM	WELM	IRWELM	L21ELM	L21pHELM
0%	1.1925	1.2019	1.2033	1.4500	**0.9715**
10%	0.9829	1.2741	1.2495	1.3829	**0.8758**
20%	1.0271	1.1237	1.1298	1.1672	**0.9490**
30%	0.9748	0.9988	1.1151	1.1241	**0.9348**
40%	0.9502	0.9482	0.9524	1.0251	**0.9192**

TABLE I summarizes the prediction accuracy of the artificial datasets under varying outlier levels. Notably, L21pHELM exhibits the lowest aRRMSE values across different outlier level. When the outlier level increases from 0% to 40%, the change in aRRMSE for L21ELM is the most significant, while the changes for other models, namely WELM, IRWELM, ELM, and L21pHELM are relatively small. In particular, for the L21pHELM model, the aRRMSE even decreases by 0.0523 as the proportion of outliers increases. This indicates that L21pHELM not only achieves superior performance compared to other models, but also exhibits greater stability.

To further assess stability, Fig. 4 utilizes a multi-target artificial dataset with a 40% outlier level to compare five models. Overall, the curves of each model remain consistent with the original sample points. Initially, the y1 curve shows that all five models maintain a similar proportion between the predicted and original values during function prediction. However, upon analyzing the y2 function curve, it becomes evident that the latter half of the curves for ELM, WELM, and IRWELM exhibit excessive smoothness compared to the original y2 curve. Conversely, the curves of L21ELM and L21pHELM are notably closer to the trajectory of the y2 function curve.

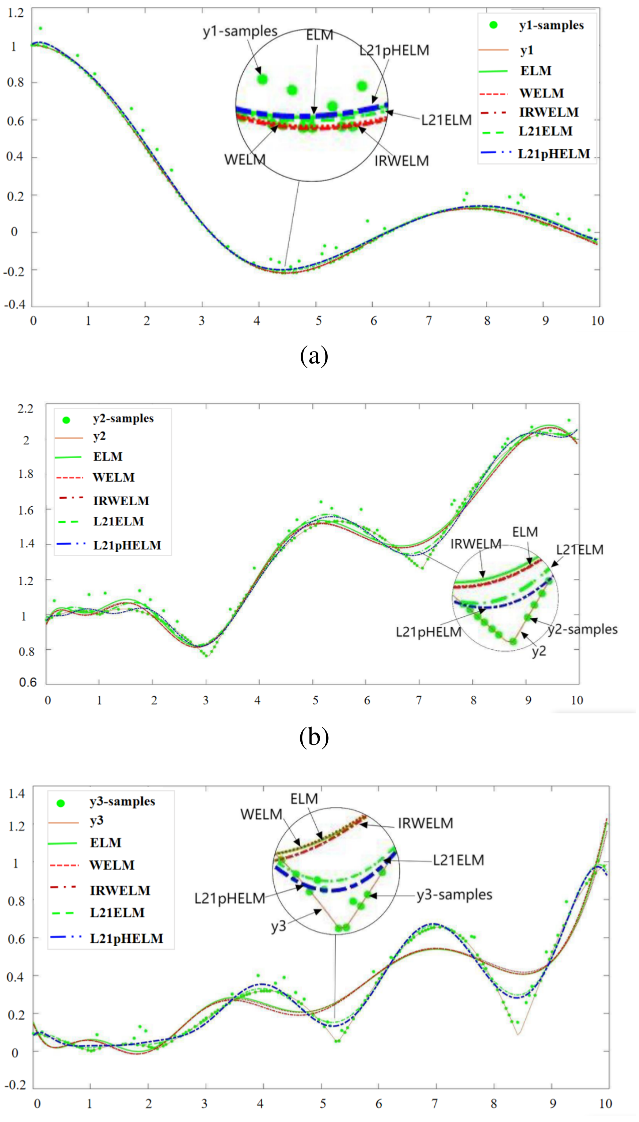
Finally, comparing the y3 function curves reveals a significant discrepancy between the function images generated by ELM, WELM, and IRWELM and the original function images. This indicates that these models are more sensitive to outliers. In contrast, the curves produced by L21ELM and L21pHELM closely match the original images, demonstrating their robustness to outliers. These results further confirm that the proposed L21pHELM maintains strong robustness even under high outlier levels.

B. Experiments on benchmark datasets

1) Robustness Analysis

To thoroughly assess the robustness of L21pHELM, this section presents further experiments on benchmark datasets. The experimental data were obtained from the Mulan datasets [40] and other benchmark datasets. First, the datasets were preprocessed and divided into training and testing sets for multi-target prediction. TABLE II summarizes the key characteristics of the 14 datasets, including the number of training samples, test samples, attributes, and targets.

To evaluate the robustness of L21pHELM in the presence of outliers, five different outlier levels—0%, 10%, 20%, 30%, and 40%—were introduced to simulate varying noise levels. All datasets were normalized to the range [0, 1]. To ensure consistency, each model was tested on all 14



(a)

(b)

(c)

Fig. 4: Prediction Objectives of Different Models on Artificial Datasets with 40% Outliers.

datasets. The samples were randomly allocated according to the specified number of training and testing samples, and the experiments were repeated ten times using cross-validation to ensure reliability.

TABLE III presents a comparison of aRRMSE without outliers across 14 benchmark datasets. The results show that L21pHELM achieved the smallest aRRMSE on 4 datasets and ranked second on the enb and sf1 datasets. Similarly, WELM and L21ELM attained the minimum aRRMSE on four datasets, while IRWELM achieved the optimal aRRMSE on the andro, sf1, and sf2 datasets. This suggests that these models exhibit better robustness compared to the standard ELM. Although L21pHELM outperforms the standard ELM, the improvements are not significant. This is because, whether L1 loss, L21 norm, or *p*-Huber loss functions are employed, when outliers are absent, the differences between model predictions and actual targets are primarily influenced by the model's complexity and parameters rather than random outliers in the data. Therefore, the impact of the loss function is relatively minor in the absence of outliers.

TABLE II: Information about multi-target regression datasets

Dataset	Training samples	Test samples	Attributes	Targets
andro	33	16	30	6
atp1d	225	112	411	6
arp7d	197	99	411	6
edm	103	51	16	2
enb	512	256	8	2
jura	239	120	15	3
oes10	269	134	298	16
oes97	223	111	263	16
ef	400	368	8	2
slump	69	34	7	3
sf1	200	123	10	3
wq	707	353	16	4
scpf	95	48	23	3
sf2	700	366	10	3

TABLE III: aRRMSE for 5 models and 14 benchmark datasets without outliers

0% level	ELM	WELM	IRWELM	L21ELM	L21pHELM
andro	0.6400	0.6375	**0.6350**	0.6741	0.6612
atp1d	0.4739	**0.4724**	0.4797	0.4849	0.4844
atp7d	0.4951	**0.4912**	0.5022	0.5098	0.5061
edm	0.9157	0.9092	0.9146	0.8985	**0.8823**
enb	0.1319	0.1357	0.1456	**0.1267**	0.1296
jura	**0.6736**	0.6914	0.7068	0.6918	0.7510
oes10	0.4580	**0.4579**	0.4580	0.4872	0.4893
oes97	**0.6675**	0.6690	0.6687	0.7146	0.7155
ef	0.1574	0.1632	0.1761	0.1440	**0.1413**
slump	**0.6722**	0.6920	0.6748	0.6915	0.6791
sf1	1.0753	**1.0611**	**1.0611**	**1.0611**	1.0659
wq	0.9999	1.0002	1.0009	**0.9998**	**0.9998**
scpf	1.0004	1.0036	1.0068	0.9996	**0.9990**
sf2	1.0305	1.0616	**1.0163**	**1.0163**	1.0763

Based on the data presented in TABLE IV, it can be inferred that when the outlier level is set to 10% and 20%, the proposed method L21pHELM achieves the best results in 25 out of 28 cases. Next, we analyze the datasets with 10% and 20% outliers. TABLE IV shows that L21pHELM achieved the best aRRMSE on 12 datasets. When the outlier level was 10%, the aRRMSE ranked second on the sf1 dataset. On the slump dataset with a 20% outlier proportion, the aRRMSE was only surpassed by the L21ELM algorithm. This indicates that L21pHELM can achieve more optimal aRRMSE compared to the scenario where the levels of outliers is 0%. A comparison with TABLE III reveals that the optimal solutions tend to cluster around the L21pHELM model, validating its ability to resist outliers. However, due to the random generation of outliers, the aRRMSE of some models may increase as outliers appear, further confirming their sensitivity to outliers. We can measure the extent to which each model is affected by outliers by examining the changes in aRRMSE of different models under different datasets. The growth rate of aRRMSE for the proposed model in this section is approximately one-tenth that of the standard ELM, demonstrating that L21pHELM exhibits excellent robustness in the presence of outliers. By calculating the increment from 10% outliers to 20% outliers, it was found that the average increment of aRRMSE for L21pHELM on each dataset was 0.0265, while for ELM, it was 0.0561, for WELM, it was

TABLE IV: aRRMSE for 5 models and 14 benchmark datasets with 10% ,20%outliers

Dataset	Outlier levels	ELM	WELM	IRWELM	L21ELM	L21pHELM
andro	10%	0.7725	0.7302	0.7398	0.7295	**0.6860**
	20%	0.8483	0.8332	0.8162	0.8328	**0.7881**
atp1d	10%	0.5923	0.5158	0.5235	0.5009	**0.4918**
	20%	0.7005	0.5850	0.5488	0.5115	**0.4928**
atp7d	10%	0.5859	0.5082	0.5088	0.5028	**0.4955**
	20%	0.7046	0.5634	0.5257	0.5262	**0.5006**
edm	10%	0.9798	0.9453	0.9297	0.9332	**0.9266**
	20%	0.9943	0.9808	0.9388	0.9569	**0.9326**
enb	10%	0.3827	0.2567	0.1849	0.1813	**0.1639**
	20%	0.5074	0.3536	0.2534	0.2864	**0.1891**
jura	10%	0.8172	0.7326	0.7343	0.7114	**0.7033**
	20%	0.8673	0.8201	0.8262	0.7455	**0.7324**
oes10	10%	0.7681	0.5178	0.5045	0.4852	**0.4729**
	20%	0.8597	0.6528	0.5927	0.5230	**0.4753**
oes97	10%	0.8692	0.6609	0.6226	0.6520	**0.6057**
	20%	0.8985	0.7248	0.5977	0.6490	**0.5772**
ef	10%	0.3894	0.2988	0.2071	0.1880	**0.1648**
	20%	0.5097	0.3595	0.2983	0.2738	**0.2057**
slump	10%	0.8812	0.8438	0.8149	0.7498	**0.7331**
	20%	0.9182	0.8931	0.9170	**0.8417**	0.8454
sf1	10%	1.0207	1.0391	1.0452	**1.0105**	1.0109
	20%	1.0073	1.0249	1.0423	1.0061	**1.0053**
wq	10%	1.0000	0.9999	0.9999	1.0000	**0.9998**
	20%	1.0001	1.000	1.0000	**0.9999**	**0.9999**
scpf	10%	0.9997	0.9988	0.9981	0.9990	**0.9944**
	20%	1.0000	0.9953	0.9884	0.9889	**0.9838**
sf2	10%	1.0024	1.0029	1.0090	1.0036	**1.0017**
	20%	1.0013	1.0012	1.0078	1.0017	**0.9996**

TABLE V: aRRMSE for 5 models and 14 benchmark datasets with 30% ,40%outliers

Dataset	Outlier levels	ELM	WELM	IRWELM	L21ELM	L21pHELM
andro	30%	0.8859	0.8413	0.8552	0.8467	**0.8135**
	40%	0.8937	0.8862	0.8903	0.9134	**0.8753**
atp1d	30%	0.7929	0.7126	0.6503	0.5616	**0.5079**
	40%	0.8263	0.8192	0.7987	0.6077	**0.5133**
atp7d	30%	0.7855	0.7198	0.6624	0.5641	**0.5112**
	40%	0.8166	0.7980	0.7700	0.6159	**0.5173**
edm	30%	0.9941	0.9854	0.9299	0.9575	**0.9289**
	40%	0.9801	0.9773	0.9608	0.9645	**0.9571**
enb	30%	0.6070	0.4578	0.3928	0.3404	**0.2652**
	40%	0.6671	0.6399	0.6167	0.3834	**0.3158**
jura	30%	0.9142	0.8915	0.8594	0.7924	**0.7906**
	40%	0.9280	0.9262	0.9268	0.8433	**0.8047**
oes10	30%	0.9159	0.7824	0.7047	0.5411	**0.4818**
	40%	0.9526	0.9387	0.9128	0.6051	**0.5079**
oes97	30%	0.9455	0.8628	0.7336	0.6786	**0.6039**
	40%	0.9734	0.9650	0.9463	0.7639	**0.5926**
ef	30%	0.6082	0.4656	0.4206	0.3427	**0.2920**
	40%	0.6630	0.6437	0.6349	0.3833	**0.3191**
slump	30%	0.9458	0.9584	0.9694	0.9327	**0.9084**
	40%	0.9499	0.9466	0.9650	0.9193	**0.9143**
sf1	30%	1.0042	1.0094	1.0376	1.0015	**1.0000**
	40%	1.0059	1.0044	1.0206	1.0030	**1.0029**
wq	30%	1.0000	0.9999	0.9999	1.0000	**0.9998**
	40%	0.9999	1.0000	1.0001	0.9999	**0.9998**
scpf	30%	1.0000	0.9964	0.9877	0.9753	**0.9715**
	40%	0.9997	0.9985	0.9946	**0.9575**	0.9603
sf2	30%	1.0012	**0.9998**	1.0005	1.0004	**0.9998**
	40%	1.0022	1.0018	1.0020	1.0007	**0.9998**

0.0554, for IRWELM, it was 0.0435, and for L21ELM, it was 0.0382. From the perspective of the incremental mean, it was evident that the incremental mean of L21pHELM was significantly smaller compared to other models, with ELM exhibiting the largest incremental mean. This can be attributed to the use of the L2 loss function in the first three models, which amplifies the impact of outliers, thereby increasing the models sensitivity to outliers. On the other hand, the p-Huber loss function employed by L21pHELM reduces sensitivity to outliers by adjusting parameters to control the range of L1 and L2 losses. Consequently, as the levels of outliers increases, L21pHELM demonstrates robust resistance to outliers.

TABLE V presents a comparison of different models at outlier levels of 30% and 40%. It is evident that the L21pHELM model consistently achieved the optimal aRRMSE across all 14 datasets under the 30% outlier condition, surpassing the optimal aRRMSE attained under a 20% outlier level by one. Moreover, in the sf2 dataset, WELM also attained a comparable optimal aRRMSE. Comparing the incremental changes in aRRMSE from 20% to 30% outliers, the results show that L21pHELM has the smallest change in aRRMSE, indicating its strong robustness against outliers and minimal susceptibility to their influence.

As depicted in TABLE V, under a 40% outlier level, L21pHELM achieved the optimal aRRMSE in 13 out of 14 datasets, securing the second position in the scpf dataset. When analyzing the growth rate of outlier levels, it was observed that L21pHELM maintained the smallest growth rate, while the first three models exhibited significantly higher growth rates in aRRMSE compared to L21pHELM. This phenomenon can be attributed to the linear increase in L2 loss with error escalation, particularly in environments with high levels of outliers. In contrast, the p-Huber loss function does not exhibit linear growth similar to L2 loss in such outlier-rich environments; instead, it reduces model sensitivity to errors through the parameter p. Overall, even under the highest levels of outliers, the L21pHELM model demonstrates exceptional robustness. Therefore, based on the Mulan dataset experiment, it is evident that the models proposed in this paper exhibit strong performance in terms of aRRMSE, even with varying levels of outliers. As outlier levels continuously increase, the growth rate of aRRMSE for the models proposed in this section remains the lowest compared to other models. This highlights that through iterative optimization of model parameters, the trained model exhibits significant robustness against outliers.

2) Sparsity Analysis

As previously mentioned, the L21pHELM proposed in this section enhances sparsity through the introduction of L21 norm regularization. In this context, the sparsity of specific components of β is determined by setting a threshold, where components with absolute values below this threshold are considered sparse. To validate the capability of L21pHELM in learning sparse representations of regression problems, this section conducted an analysis involving five models: ELM, WELM, IRWELM, L21ELM, and L21pHELM.

Following the outlined procedure, a column vector is derived by computing the number of elements in each row of the target weight matrix that satisfy the sparsity condition. Subsequently, these column vectors corresponding to the five models are combined into a matrix, and the results for each row are ranked. Finally, the average ranking of each column serves as the sparsity ranking indicator for this section. Presented in TABLE VI are the average rankings of row sparsity across the edm, jura, oes97, and sf2 benchmark datasets at a 40% outlier level. Notably, both the L21ELM and L21pHELM models, which employ the L21 norm, secured top positions among the five models. This further validates the effectiveness of the L21 norm in achieving row sparsity by eliminating redundant neurons. The sparsity ranking results highlight the superior sparsity achieved through the application of the L21 norm in the models.

TABLE VI: Average rank of row sparsity under 40% outliers

Dataset	ELM	WELM	IRWELM	L21ELM	L21pHELM
edm	1.8100	1.8100	1.8100	**1.1550**	1.1770
jura	2.8960	2.8010	2.7910	1.4390	**1.0490**
oes97	2.6170	2.6210	2.6210	1.1670	**1.1010**
sf2	2.4810	2.4800	2.4810	**1.0440**	1.0500

3) *Friedman Test*

To more comprehensively assess the effectiveness of the five algorithms, we performed a statistical evaluation of the obtained results. The Friedman test [41] is a widely used statistical method for comparing algorithm performance across different datasets and under varying outlier conditions. Our null hypothesis states that there is no significant difference in performance among all the algorithms. The calculation of the Friedman test involves the following statistical measure:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{19}$$

where the test statistic is χ_F^2 distributed with k-1 degrees of freedom, with N as the dataset count, k as the algorithm count, and R_j as the algorithm's mean rank. Furthermore, the Friedman test statistic approximates an F distribution under the null hypothesis:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{20}$$

with $(k-1)(N-1)$ degrees of freedom.

TABLE VII: average ranks of aRRMSE for the five algorithms at different outliers levels on the Benchmark datasets

Algorithm	0%	10%	20%	30%	40%
L21pHELM	3.0714	1.0714	1.0714	1.0714	1.0714
ELM	2.8571	4.6429	4.6429	4.6429	4.5714
L21ELM	3.3571	3.4286	3.1429	3.2143	3.5
WELM	3.0714	2.5	2.4286	2.4286	2.2857
IRWELM	2.6429	3.3571	3.7143	3.6429	3.5714

TABLE VIII: Related information about the Friedman test at different outliers levels

Outlier	X_F^2	F_F	CD	d(L21pHELM–)			
				ELM	L21ELM	WELM	IRWELM
0%	1.6	0.3824	1.4695	0.2143	0	0.4286	0.2857
10%	39.0857	30.0405	1.4695	**3.5714**	1.4286	**2.2857**	**2.3571**
20%	33.6571	40.7429	1.4695	**3.5714**	1.3571	**2.6429**	**2.0714**
30%	40.3429	33.4964	1.4695	**3.5714**	1.3571	**2.5714**	**2.1429**
40%	40.7429	34.7154	1.4695	**3.5**	1.2143	**2.5**	**2.4286**

In the experiments, N = 14 and k = 5. The comparative performance rankings of the algorithms, as measured by aRRMSE under increasing outlier levels, are summarized in Table VII through their average rank values R_j. The Friedman test statistics were calculated based on the average algorithm rankings from Table VII, with χ_F^2 and F_F values subsequently reported in Table VIII. For α = 0.05, $F_\alpha(4,52)$ = 2.550. As evident from Table VIII, the condition $F_F > F_\alpha$ holds across outlier levels from 10% to 40%, leading us to reject the null hypothesis of equivalent algorithmic performance. For an in depth performance evaluation of L21pHELM against competing algorithms, the Nemenyi test [42] serves as a commonly employed post hoc analysis method. The critical difference (CD) is mathematically defined as:

$$\mathrm{CD} = q_\alpha\sqrt{\frac{k(k+1)}{6N}} = 2.459 \times \sqrt{\frac{5\times(5+1)}{6\times 14}} = 1.4695 \tag{21}$$

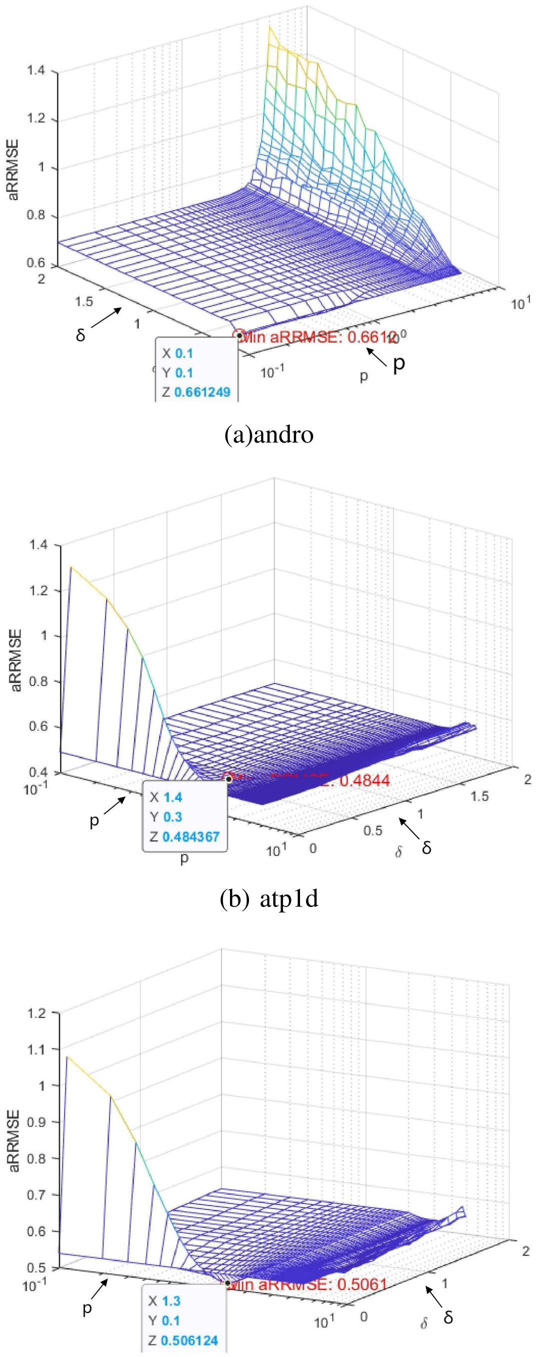
where $q_{0.1}$= 2.459 represents the critical value at α = 0.1 significance level. Performance differentials between the proposed L21pHELM method and comparative algorithms were statistically evaluated using Nemenyi's test on mean rank distributions. A statistically significant performance difference between algorithms is established when their mean rank difference surpasses the CD threshold. In our analysis, significant comparisons are highlighted in boldface in Table VIII.

Table VIII reveals no statistically significant differences between L21pHELM and the four algorithms in outlier free conditions, indicating comparable baseline performance. However, at outlier contamination levels between 10% and 40%, L21pHELM demonstrates statistically superior performance compared to ELM, WELM, and IRWELM, while showing no significant improvement over L21ELM according to the Nemenyi test results. Although the mean rank difference between L21pHELM and L21ELM does not exceed the CD threshold at 10%-40% outlier levels, the observed difference approaches statistical significance. This observed performance similarity may be attributed to their shared utilization of L21 norm regularization, which inherently limits substantial performance divergence between the two algorithms. Collectively, the experimental results demonstrate that L21pHELM exhibits superior generalization capability and exceptional outlier robustness, particularly under high outlier contamination scenarios.

4) *Parameter Analysis*

To further investigate the influence of various parameters on experimental outcomes, an analysis was conducted to explore their impact on model performance. Initially, a fixed optimal regularization parameter C was chosen, and the effects of different values of δ and p on model performance were examined. Three datasets—andro, atp1d, and atp7d—were selected for analysis. The traversal range for δ was [0, 2] with a step size of 0.1, while p ranged from [0.1, 5] with the same step size. In the accompanying figures, points denote the optimal aRRMSE and the corresponding coordinates where this optimal value is achieved. The X-axis represents p, the Y-axis represents δ, and the Z-axis represents aRRMSE. The parameters under scrutiny are p and δ.

As depicted in Fig. 5(a), upon analyzing the traversal



(a)andro

(b) atp1d

(c) atp7d

Fig. 5: aRRMSE values of different parameters p and δ without outliers

data from the andro dataset, it is evident that when the value of p becomes too large, the rate of aRRMSE increase also escalates rapidly, peaking around 1.3. This peak occurs when both δ and p values are very small, indicating that within a certain range, smaller exponents p in the p-Huber loss function allow L21pHELM to achieve relatively low aRRMSE across all δ ranges. Moving to the analysis of the atp1d dataset, illustrated in Fig. 5(b), it is notable that the change in aRRMSE remains relatively modest across most parameter conditions. However, significant changes in aRRMSE begin to manifest when p exceeds a certain threshold. A similar condition is observed in the analysis of the atp7d dataset, as shown in Fig. 5(c). Overall, when p is relatively large, the aRRMSE of the three datasets will experience violent fluctuations, significantly deviating from the minimum aRRMSE. This trend is also evident in the variations of δ. Therefore, to achieve optimal aRRMSE, it's crucial for the selection range of both parameters to be appropriate, allowing the model to attain peak performance.

V. Conclusion

This section primarily addresses the challenges of robustness and insufficient generalization performance encountered by traditional regression ELM when applied to multi-target datasets. It introduces an L21 norm regularization ELM combined with a p-Huber loss function to tackle regression problems with outliers. L21 norm regularization is an effective technique for inducing row sparsity, which can dynamically eliminate potential noise and autocorrelated neurons in ELM. The p-Huber loss function regulates the variability of the loss function through parameter p and exhibits strong robustness against outliers. The optimal target weight matrix is derived through an iterative reweighting method. Experimental findings on 14 multi-target benchmark datasets and artificial datasets demonstrate that L21pHELM exhibits superior robustness and row sparsity compared to other models, delivering commendable performance across benchmark datasets.

Due to the wide range of traversal required for the two parameters p and δ in the p-Huber loss function, the process tends to extend the model training time to identify the optimal parameters. Consequently, the principal enhancement introduced by the L21pHELM model in this article focuses on refining parameter selection.

References

[1] Aho T, Zenko B, Dzzeroski S, et al. Multi target regression with rule ensembles[J]. Journal of machine learning research, 2012, 13(8).

[2] Borchani H, Varando G, Bielza C, et al. A survey on multi-output regression[J]. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2015, 5(5): 216-233.

[3] Inaba F K, Salles E O T, Perron S, et al. DGR ELM distributed generalized regularized ELM for classification[J]. Neurocomputing, 2018, 275: 1522-1530.

[4] Kocev D, Džeroski S, White M D, et al. Using single and multi target regression trees and ensembles to model a compound index of vegetation condition[J]. Ecological modelling, 2009, 220(8): 1159-1168.

[5] Yang B, Wang Z, Guan X. Optimal Operation of Integrated Energy Systems Under Uncertainties: Distributionally Robust and Stochastic Methods[M]. Elsevier, 2023.

[6] Melki G, Cano A, Kecman V, et al. Multi target support vector regression via correlation regressor chains[J]. Information sciences, 2017, 415: 53-69.

[7] Zhang K, Luo M. Outlier-robust extreme learning machine for regression problems[J]. Neurocomputing, 2015, 151: 1519-1527.

[8] da Silva B L S, Inaba F K, Salles E O T, et al. Outlier robust extreme machine learning for multi target regression[J]. Expert systems with applications, 2020, 140: 112877.

[9] Chen C, He B, Ye Y, et al. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent[J]. Mathematical programming, 2016, 155(1): 57-79.

[10] Xiong T, Bao Y, Hu Z. Multiple output support vector regression with a firefly algorithm for interval-valued stock price index forecasting[J]. Knowledge-Based Systems, 2014, 55: 87-100.

[11] Wu Y, Zhao D, Peng J, et al. Hybrid Electric Vehicle Powertrain Mounting System Optimization Based on Cross Industry Standard Process for Data Mining[J]. Electronics, 2024, 13(6): 1117.

[12] Deng W, Zheng Q, Chen L. Regularized extreme learning machine[C] //2009 IEEE symposium on computational intelligence and data mining. IEEE, 2009: 389-395.

[13] Hastie T, Tibshirani R, Friedman J, et al. Linear methods for regression[J]. The elements of statistical learning: data mining, inference, and prediction, 2009: 43-99.

[14] Shalev Shwartz S, Ben David S. Understanding machine learning: From theory to algorithms[M]. Cambridge university press, 2014.

[15] Deng C, Huang G, Xu J, et al. Extreme learning machines: new trends and applications[J]. Science China. information sciences, 2015, 58(2): 1-16.

[16] Zou H, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the royal statistical society series B: statistical methodology, 2005, 67(2): 301-320.

[17] Zhang Z, Lai Z, Xu Y, et al. Discriminative elastic net regularized linear regression[J]. IEEE transactions on image processing, 2017, 26(3): 1466-1481.

[18] Nie F, Huang H, Cai X, et al. Efficient and robust feature selection via joint L21 norm minimization[J]. Advances in neural information processing systems, 2010, 23: 1813-1821.

[19] Kong D, Liu J, Liu B, et al. Uncorrelated group lasso[C] //Proceedings of the AAAI conference on artificial intelligence. 2016, 30(1).

[20] Luo M, Nie F, Chang X, et al. Adaptive unsupervised feature selection with structure regularization[J]. IEEE transactions on neural networks and learning systems, 2017, 29(4): 944-956.

[21] Li Y, Mark B, Raskutti G, et al. Graph based regularization for regression problems with highly correlated designs[C] //2018 IEEE global conference on signal and information processing (GlobalSIP). IEEE, 2018: 740-742.

[22] Li R, Wang X, Lei L, et al. L21 Norm Based Loss Function and Regularization Extreme Learning Machine[J]. IEEE access, 2018, 7: 6575-6586.

[23] Solorio Fernández S, Carrasco Ochoa J A, Martínez Trinidad J F. A review of unsupervised feature selection methods[J]. Artificial intelligence review, 2020, 53(2): 907-948.

[24] Preeti, Bala R, Dagar A, et al. A novel online sequential extreme learning machine with L21 norm regularization for prediction problems[J]. Applied intelligence, 2021, 51: 1669-1689.

[25] Yuan M, Lin Y. Model selection and estimation in regression with grouped variables[J]. Journal of the royal statistical society series B: statistical methodology, 2006, 68(1): 49-67.

[26] Zhang Z, Xu Y, Yang J, et al. A survey of sparse representation: algorithms and applications[J]. IEEE access, 2015, 3: 490-530.

[27] Bishop C M. Pattern recognition and machine learning[J]. Springer google schola, 2006, 2: 5-43.

[28] Zhang T. Statistical behavior and consistency of classification methods based on convex risk minimization[J]. The annals of statistics, 2004, 32(1): 56-85.

[29] Liu W, Pokharel P, Principe J. Correntropy: Properties and applications in non Gaussian signal processing[J]. IEEE transactions on signal processing, 2007, 55(11): 5286-5298.

[30] Zhan W, Wang K, Cao J. Elastic net based robust extreme learning machine for one-class classification[J]. Signal processing, 2023, 211: 109101.

[31] Yuan C, Yang L. Robust twin extreme learning machines with correntropy based metric[J]. Knowledge Based Systems, 2021, 214: 106707.

[32] Xing H, Wang X. Training extreme learning machine via regularized correntropy criterion[J]. Neural computing and applications, 2013, 23: 1977-1986.

[33] Luo L, Wang K, Lin Q. Robust Extreme Learning Machine Based on p order Laplace Kernel Induced Loss Function[J]. International Journal of Advanced Computer Science and Applications, 2024, 15(4): 1281-1291.

[34] Wang K, Pei H, Cao J, et al. Robust regularized extreme learning machine for regression with non convex loss function via DC program[J]. Journal of the franklin institute, 2020, 357(11): 7069-7091.

[35] Gupta D, Hazarika B, Berlin M. Robust regularized extreme learning machine with asymmetric Huber loss function[J]. Neural computing and applications, 2020, 32(16): 12971-12998.

[36] Chen K, Lv Q, Lu Y, et al. Robust regularized extreme learning machine for regression using iteratively reweighted least squares[J]. Neurocomputing, 2017, 230: 345-358.

[37] Khrapov A, Popov V, Sadekova T, et al. Improving Diffusion Models's Data Corruption Resistance using Scheduled Pseudo Huber Loss[J]. arXiv preprint arXiv:2403.16728, 2024.

[38] Zhou X, Xiao D, Fu Y. ”Research on Incremental Huber Support Vector Regression Algorithm.” Operations Research and Management Science, 2022, 31(08): 137-142.

[39] Chen Z, Gao H, Wang K. A motion based object detection method [C] //2020 2nd international conference on information technology and computer application (ITCA). IEEE, 2020: 280-283.

[40] Tsoumakas G, Spyromitros Xioufis E, Vilcek J, et al. Mulan: A java library for multi label learning[J]. The journal of machine learning research, 2011, 12: 2411-2414

[41] Demšar J. Statistical comparisons of classifiers over multiple data sets[J]. The Journal of Machine learning research, 2006, 7: 1-30.

[42] Benavoli A, Corani G, Mangili F. Should we really use post-hoc tests based on mean-ranks?[J]. The Journal of Machine Learning Research, 2016, 17(1): 152-161.