Applied Intelligence (2024) 54:8745–8760
https://doi.org/10.1007/s10489-024-05651-3

Pinball-Huber boosted extreme learning machine regression: a multiobjective approach to accurate power load forecasting

Yang Yang[1] · Hao Lou[1] · Zijin Wang[1] · Jinran Wu[2]

Accepted: 25 June 2024 / Published online: 3 July 2024

Abstract

Power load data frequently display outliers and an uneven distribution of noise. To tackle this issue, we present a forecasting model based on an improved extreme learning machine (ELM). Specifically, we introduce the novel Pinball-Huber robust loss function as the objective function in training. The loss function enhances the precision by assigning distinct penalties to errors based on their directions. We employ a genetic algorithm, combined with a swift nondominated sorting technique, for multiobjective optimization in the ELM-Pinball-Huber context. This method simultaneously reduces training errors while streamlining model structure. We practically apply the integrated model to forecast power load data in Taixing City, which is situated in the southern part of Jiangsu Province. The empirical findings confirm the method's effectiveness.

Keywords Load forecasting · Robust loss function · Multi-objective optimization · Neural networks · Extreme learning machine

1 Introduction

Power load forecasting forms the foundation of power system scheduling and operation [1, 2]. Ensuring accurate power load forecasting is crucial for the stable operation and economic efficiency of a power system [3, 4]. Power load data often encompass anomalies and asymmetric noise because of factors such as climate fluctuations and changes in market demands [5]. These data elements can impede model training, hence affecting the accuracy of power load forecasting and making the subject of mitigating asymmetric noise a significant concern in the field of power load forecasting.

Despite significant advancements in power load forecasting methods, there is a need for further improvement to meet the current demands. Existing loss functions have made substantial progress in enhancing robustness and accuracy. However, their symmetric nature renders them inadequate for effectively addressing the deviations caused by outliers in power load datasets. Thus, there is a need for a comprehensive and robust loss function that can consider both the magnitude and direction of errors within the machine learning framework. Additionally, the number of hidden layer nodes in ELM significantly impacts the complexity of model training and final predictive accuracy. The lack of reliable benchmarks to balance the structural parameters and prediction accuracy also poses a challenge. Therefore, integrating novel and effective optimization techniques is essential for enhancing the ELM model and determining the optimal parameters for precise predictive modeling.

The present paper focuses on two key aspects: asymmetric loss function and multi-objective optimization. To address the aforementioned challenges, the paper contributes the following:

(1) An asymmetric Pinball-Huber loss function for more effective data handling is developed. Because of its superior characteristics compared with other loss functions, it has been incorporated into the training objective of the ELM model.
(2) The multiobjective optimization algorithm NSGA-II has been used to optimize two critical objectives of the ELM model: training error and output weight. By considering

✉ Jinran Wu
ryan.wu@acu.edu.au

Yang Yang
yyang@njupt.edu.cn

Zijin Wang
1321058617@njupt.edu.cn

[1] Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, PR China

[2] Australian Catholic University, North Sydney 2060, NSW, Australia

the input weights, hidden layer thresholds, and hidden node numbers as input parameters, the optimized ELM model can achieve minimized training errors and a more streamlined network structure.

(3) The superiorty of proposed NSGA-II-ELM-Pinball-Huber model is validated by comparing it with benchmarks (LSTM, GRU, CNN-BiLSTM-Attention) using power load data from Taixing City. This validation has emphasized the effectiveness of the Pinball-Huber loss, highlighting the enhanced performance of the multiobjective optimization algorithm NSGA-II.

The rest of the current paper is organized as follows: Section 2 contains the literature review. Section 3 presents definitions of some terms. Section 4 gives the methodology (method). Section 5 goes over the results. Section 6 discusses the conclusions and future research areas.

2 Literature review

Over the past few decades, researchers have proposed numerous short-term load forecasting methods [6], which can be broadly categorized into physical, statistical, and intelligent methods [7]. Physical methods establish the mathematical relationships between historical data and physical characteristics to achieve power load forecasting. Statistical methods perform mathematical statistics on historical data, establishing the correlations between load and time to make predictions [8]. These models typically include linear regression (LR) [9], autoregressive integrated moving average (ARIMA) [10], gray models (GM) [11], and seasonal exponential smoothing(SEs) [7]. However, these methods fail to capture the nonlinear characteristics present in load data.

Compared with traditional physical and statistical methods, intelligent methods exhibit greater potential in handling the nonlinear fluctuations and complex relationships within power load data, hence demonstrating higher accuracy in the field of power load forecasting [12, 13]. Intelligent methods such as artificial neural networks (ANN) [14], support vector regression (SVR) [15], and ELM [16] have found extensive applications in recent power forecasting studies. Among these, ANN is adept at modeling more intricate relationships between the power load and correlated variables compared with other methods, hence leading to its widespread usage in power load forecasting [2, 17]. ANN, which is akin to the structure of the human brain, can interpret vast amounts of data and transform it into actionable knowledge [18]. ELM, an enhanced single-hidden-layer feedforward neural network, has been widely employed in forecasting tasks [19, 20].

Unlike traditional artificial neural networks, ELM's input weights and biases in the hidden layer are randomly assigned. ELM derives hidden weights through the least squares method, eliminating the need for adjusting hidden layer weights through iterative backpropagation [21]. As a result, the ELM model demonstrates faster learning and more pronounced generalization with minimal preset parameters [22]. Numerous ELM-based predictive models have been proposed, showcasing their exceptional regression capabilities in forecasting. Ni et al. [23] employed an ensemble method using ELM and lower upper bound estimation (LUBE) for short-term power prediction. Han et al. [24] developed seasonal multimodels based on ELM by considering the seasonal distribution of power features. The effectiveness of the proposed methods was validated through a comparison with other approaches. Thus, compared with shallow learning systems, ELM exhibits higher efficiency, lower computational costs, and stronger generalization.

The loss function reflects the disparity between the predicted values and actual values during the optimization process, significantly impacting the learning model's generalization and accuracy [25]. Chen et al. [26] utilized ELM enhanced with an L2-norm loss function for feature selection. Most neural network methods adopt mean squared error (MSE) or L2 loss function. Unfortunately, MSE loss function relies on Gaussian assumptions, making it sensitive to outliers and challenging to precisely evaluate nonlinear errors. Yang et al. [27] suggested employing the Huber loss function as the model's training objective. The Huber loss treats errors of different magnitudes differently. However, it lacks consideration for the direction of errors. Power load data are nonlinear and often exhibit various asymmetric noise distributions [5], necessitating the development of a new loss function that comprehensively considers both error magnitude and direction.

In conventional ANNs, including ELM, certain parameters are set randomly, leading to a degree of error and variability in the predictive outcomes. Artificial intelligence also exhibits drawbacks such as slow convergence, susceptibility to local optima, and overfitting [11, 28]. Hence, several intelligent optimization algorithms have been proposed to alleviate these limitations. Optimization algorithms applied to machine learning algorithms have further improved their regression capabilities to some extent [22]. For instance, Niu et al. [29] utilized a cooperative search algorithm that can explore the optimal hyperparameters of support vector machines (SVM), using this algorithm to predict electricity consumption in four Chinese provinces. Niu et al. [29] optimized BPNN parameters using a genetic algorithm (GA). Shang et al. [30] established a prediction model combining least squares support vector machines (LSSVM) with generalized regression neural networks and optimized the weight coefficients by using the whale optimization algorithm (WOA). Xie et al. [31] proposed a short-term power load forecasting method combining Elman neural network (ENN) and particle swarm optimization (PSO). Differing from traditional random initialization, PSO was employed to search for the optimal learning rate for ENN. Addressing the issue of model parameter determination, arithmetic optimization algorithms (AOA) [32], gene expression programming (GEP) [33], and chimpanzee optimization algorithm (ChOA) [34], among others, have been utilized. Many studies on power load forecasting solely employed single-objective algorithms to optimize a criterion. However, in practical applications, meeting multiple constraints is often necessary [7, 35].

The present paper introduces a novel power load forecasting model to address the aforementioned issues. Named NSGA-II-ELM-Pinball-Huber, this model is based on an enhanced Pinball-Huber loss function and multiobjective optimization algorithm NSGA-II. To effectively handle errors and anomalies in power load data, we introduced an asymmetric and robust Pinball-Huber loss function. Within the ELM framework, this loss function is employed as the objective, and the iteratively reweighted least squares (IRLS) method is utilized to determine the output weight vector. The present paper conducts global multiobjective optimization of the ELM model by employing the NSGA-II algorithm to simultaneously optimize training errors and output weights. The experimental results demonstrate that the proposed load forecasting model significantly enhances predictive performanc

3 The preliminaries

3.1 Regression loss function

Within various enhanced algorithms, the role of the loss function is to assess the merits and drawbacks of the improved model by computing its minimum value within the improved function. Yet during practical application, because of factors such as the loss function's objective, the nature of the application, data attributes, and the desired level of confidence in the forecasted values, a single loss function cannot be universally applied to all model experiments. Thus, a range of loss functions is required to be explored to optimize the treatment of target-type data and achieve optimal evaluation results [36].

3.1.1 L2-norm loss

L2-norm loss is a smooth function that is derivable in the whole domain and simplifies the calculation. When the error increases, the error is squared because of L2-norm loss, so that the error obtained is amplified. The L2-norm loss function can be described as follows:

$$L2(r) = \frac{1}{2}r^2, \tag{1}$$

where $r = y - \hat{y}$ is the residual, y represents the expected results, and $\hat{y}$ represents the forecasting results.

3.1.2 L1-norm loss

In the regression problem, L1-norm loss measures the absolute value of the difference between the forecasting value and true value. The L1-norm loss function can be described as follows:

$$L1(r) =\mid r \mid . \tag{2}$$

The L1-norm loss function is a function commonly used in regression problems.

3.1.3 Huber loss

The Huber loss function was proposed in 1964. It absorbs the advantages of L1-norm and L2-norm loss functions and makes up for their shortcomings. Concerning outliers in the data, Huber loss can perform more robustly. Not only is it more robust to outliers, but Huber is also derivable in the whole domain, greatly simplifying the calculation difficulty. Huber loss function can be described as follows:

$$H_\delta(r) = \begin{cases} \frac{1}{2}r^2, |r| \le \delta \\ \mid r \mid \delta - \frac{\delta^2}{2}, |r| > \delta, \end{cases} \tag{3}$$

where parameter δ represents tuning parameters, which control the quadratic and linear range. It is recommended to set the parameter δ to 1.345 [37].

3.1.4 Pinball loss

The Pinball loss function is asymmetric. It not only imposes certain penalties on outliers in data, but it also imposes additional penalties according to different situations of outliers. In addition, because of the introduction of quantile distance, the Pinball loss function improves the insensitivity to characteristic noise and resampling. The expression of the Pinball loss function is as follows:

$$P_\tau(r) = \begin{cases} r\tau, r \ge 0 \\ r(1-\tau), r < 0. \end{cases} \tag{4}$$

The parameter $\tau \in [0, 1]$. When parameter $\tau = 1$, Pinball loss is the same as the L1-norm loss in function, so the Pinball loss function can be considered a generalized L1-norm loss function. In addition, Pinball loss also absorbs the advantages of L1-norm loss and can handle the deviation of outliers.

3.1.5 Biweight loss

Tukey's Biweight loss function is also a non-convex loss function, which can overcome the interference and influence caused by outlier samples and noise samples in regression tasks, hence showing strong robustness in regression tasks [38, 39]. The Biweight loss function is defined as follows:

$$B_c(r) = \begin{cases} \frac{c^2}{6}[1-(1-(\frac{r}{c})^2)], \mid r \mid \leq c \\ \frac{c^2}{6}, \text{ otherwise}, \end{cases} \tag{5}$$

where c is a tuning constant, which is generally specified as 4.685. At this time, Tukey's Biweight can achieve a regression effect like that of the L2-norm loss function (95% progressive) in minimizing the variance consistent with the normal distribution [40]. Tukey's Biweight suppresses the influence of outliers during backpropagation by reducing the gradient size to near zero. Another interesting feature of this loss function is that it imposes a soft constraint between the inner layer and outlier without setting a hard threshold for the residual.

3.1.6 Lncosh loss

Lncosh is a loss function commonly used in regression tasks, with high smoothness. Define it as [41, 42]:

$$L(r) = ln(cosh(r)). \tag{6}$$

For the smaller residual r, $ln(cosh(r))$ is approximately equal to $\frac{r^2}{2}$; for the larger residual r, it is roughly equal to $\mid r \mid -ln2$. This means that the working principle of Lncosh is very similar to the mean square error to a large extent, but it is not greatly affected by the occasional wrong forecasting. It has all the advantages of the Huber loss function, but unlike Huber loss, it is quadratically differentiable everywhere.

3.2 Extreme learning machine

Unlike traditional feedforward neural networks, such as BP, ELM is a single hidden layer feedforward neural network that removes the requirement to set an excessive number of node parameters. ELM was introduced by Huang et al. [16] and has demonstrated substantial progress in the realm of artificial intelligence algorithms.

ELM stands out from conventional neural networks because of its distinctive approach. ELM selects its weight matrix between the input and hidden layers randomly, along with hidden layer thresholds, without any further adjustments during algorithm execution. With no need for additional parameter settings, ELM offers simplicity in its usage. As highlighted by Huang et al. [43], ELM commonly employs the Moore-Penrose generalized inverse to determine key node weights. This methodology involves only a single calculation step (linear equation operation) to establish the weight matrix between the hidden and output layers [44]. Unlike backpropagation, there's no gradient operation, significantly reducing computational demands and enhancing speed. Furthermore, ELM demonstrates superior generalization compared with alternative algorithms. The structural diagram of ELM is shown in Fig. 1.

A typical ELM network structure consists of an input layer, a hidden layer, and an output layer, with n, L, and m nodes, respectively. For data set $(x_i, y_i)(i = 1, 2, ..., N)$ with N samples, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T$ is the input vector, $y_i = [y_{i1}, y_{i2}, ..., y_{im}]^T$ is output vector, and the output of ELM can be described as:

$$y = \sum_{j=1}^{L} \beta_j G(w_j \cdot x_i + b_j) \quad (i = 1, 2, ..., n) \tag{7}$$

where w_j is the weight from the input layer to the j-th hidden layer node, b_j is the threshold of the j-th hidden layer node, β_j is the output layer weight connecting the j-th hidden layer node, and $G(\cdot)$ represents the activation function.

Equation (7) can be simplified as $H\beta = Y$. The objective function of the ELM model can be written as follows:

$$min \quad ||H\beta - Y||. \tag{8}$$

Using the least square method to solve the (8), the solution β is the following:

$$\beta = (H^T H)^{-1} H^T Y = H^+ Y, \tag{9}$$

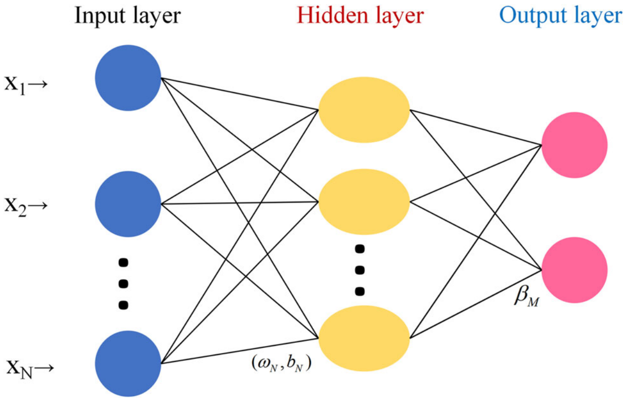


Fig. 1 The structural diagram of ELM. ($(x_1, x_2, ..., x_N)$ is the input of ELM. (w_N, b_N) is the weight and threshold of hidden layer. β_M is the output layer weight.)

where H^+ is the Moore-Penrose generalized inverse of the hidden layer output matrix H.

3.3 Multiobjective optimization

The concept behind the multiobjective optimization algorithm is to identify a collection of optimal Pareto solutions, where each solution fulfills the fundamental criteria of multiple optimization objectives and showcases an optimal state holistically. Within the optimal Pareto solution set, no other solution surpasses itself in all optimization objectives [45]. Achieving this demands that the optimization algorithm extensively explores the solution set, guarantees a global optimization outcome, and prevents being trapped in local optimization.

Derived from biological genetic theory, the genetic algorithm has evolved and found applications across diverse domains [46]. By incorporating the genetic algorithm, the drawbacks associated with traditional multiobjective optimization approaches, such as the risk of converging to local optima, are circumvented. This integration ensures that the solutions' diversity is effectively maintained.

The general multiobjective optimization problem can be described as follows:

$$\begin{aligned} min \quad & F(x) = [f_1(x), f_2(x), ..., f_n(x)] \\ & s.t. \quad x \in C, \end{aligned} \tag{10}$$

where $f_i(x)$ is the optimization objective, x is the solution, and C is the constraint.

NSGA-II is an advanced multiobjective optimization algorithm that was improved by Deb et al. [47]. NSGA-II introduces the concepts of fast nondominated sorting, crowding-distance sorting, and elitist strategy, which greatly enhance the practical application of NSGA-II. In NSGA-II, we can initialize a certain population P and use the genetic algorithm to select, cross, and mutate the parent population P to produce the offspring population Q. After fast nondominated sorting and crowding distance sorting of the combined population $R = P \bigcup Q$, the new population and its Pareto optimal solution set are obtained by using the elitist strategy. The outflow diagram of NSGA-II is shown in Fig. 2. The specific steps are as follows:

Step 1: Set population and iteration times and initialize the parent population P.

Step 2: For the parent population P, conduct fast nondominated sorting and crowding distance sorting and assign each individual the rank.

Step 3: Generate the offspring population Q_0 through tournament selection, simulated binary crossover, and polynomial mutation.

Step 4: Combine the parent P_t and offspring Q_t to get the population $R_t = P_t \bigcup Q_t$, where t is the number of iterations; a new parent population P_{t+1} is selected through elitist strategy.

Step 5: When the iteration reaches the specific number or the termination condition is met, the final population and its Pareto solution set will be obtained; otherwise, let the number of iterations $t = t + 1$ and go to Step 2.

4 The proposed method

4.1 Proposed Pinball-Huber loss

Section 3.1 provides an overview of six fundamental loss functions: L2-norm, L1-norm, Huber, Pinball, Biweight, and Lncosh. The strengths, weaknesses, and suitable application contexts for each loss function are analyzed. Upon examination and consolidation, it is evident that these loss functions often lack compatibility with robustness and accuracy in diverse evaluation models, measurement approaches, and forecasting experiments. Additionally, they tend to inadequately address standard positive and negative errors and outliers in machine learning challenges. As a result, this may lead to suboptimal evaluation levels and reduced accuracy in forecasting outcomes.

To address the aforementioned challenges, we propose a solution by merging the Pinball loss with the Huber loss. The Pinball loss function offers the ability to adapt to positive and negative errors during forecasting computations, displaying self-adjusting asymmetry. On the other hand, the Huber loss function demonstrates remarkable robustness and effectively handles outliers; however, it treats both positive and negative issues concurrently in the algorithmic process, leading to a reduction in forecasting precision. Our innovation lies in the development of novel loss functions, combining the attributes of Huber and Pinball. This allows for distinct measures to be applied to diverse errors within the training procedure, significantly enhancing the model's performance. The proposed Pinball-Huber loss function is presented as follows:

$$PH_{\delta,\tau}(r) = \begin{cases} \frac{1}{2}r^2\tau, 0 \le r \le \delta \\ \frac{1}{2}r^2(1-\tau), -\delta \le r \le 0 \\ (| r | \delta - \frac{\delta^2}{2})\tau, r > \delta \\ (| r | \delta - \frac{\delta^2}{2})(1-\tau), r < -\delta. \end{cases} \tag{11}$$

The newly introduced Pinball-Huber loss function comprises two adjustable parameters: δ and τ. Notably, these parameters originate from the Huber and Pinball loss functions and are skillfully merged to leverage their distinct roles. Their combined utilization allows for tailored actions based

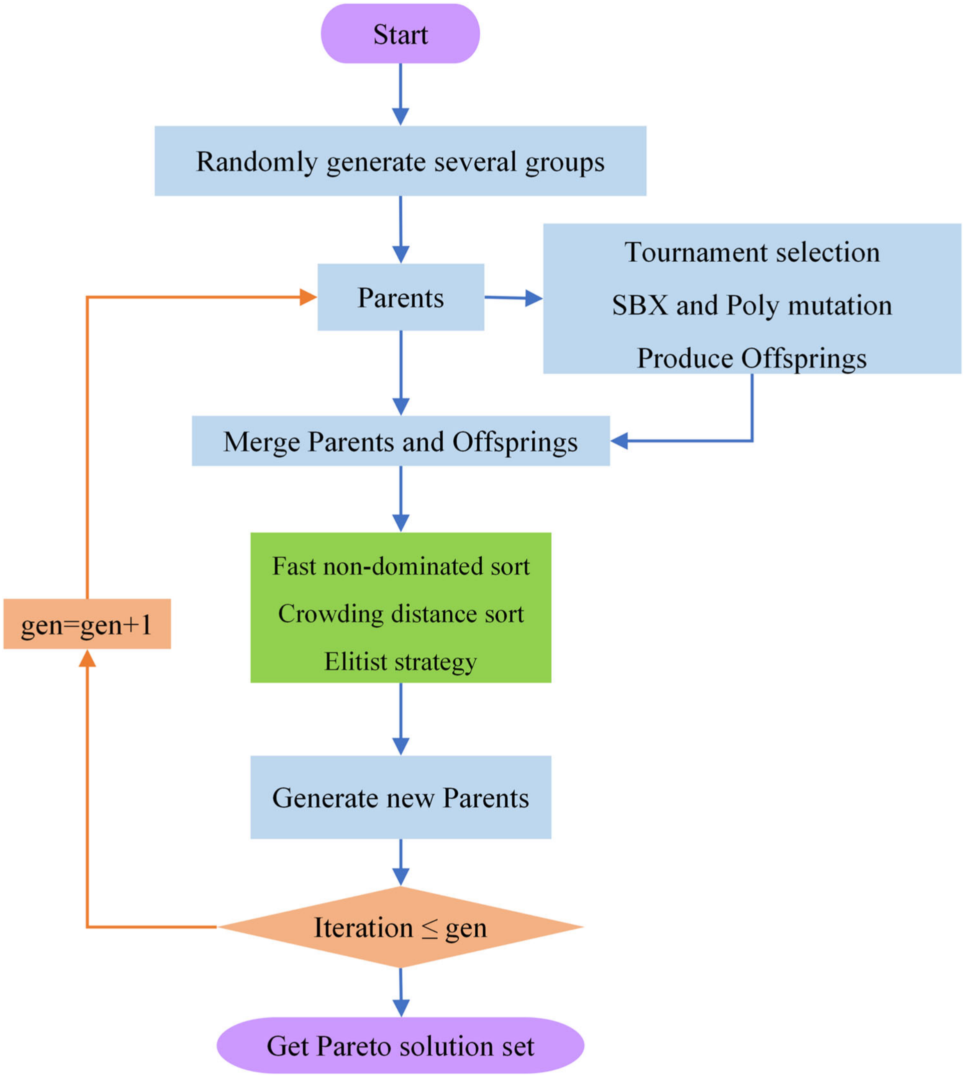


Fig. 2 The outflow diagram of NSGA-II

on the magnitude and direction of training errors. Beyond refining the accuracy of the foundational loss function, the Pinball-Huber approach introduces a novel perspective by categorizing training errors based on their directional attributes. This innovative method presents a fresh approach for addressing outliers. In the context of the power system, where power load data are influenced by variables like weather, season, and market demand, volatility and the presence of outliers and asymmetric noise are common. Our proposed Pinball-Huber loss function addresses these intricacies by meticulously dissecting errors and handling positive and negative scenarios in distinct ways.

4.2 NSGA-II-ELM-Pinball-Huber

In the practical application of ELM, there exists a fundamental trade-off between forecasting accuracy and network structure complexity. Achieving higher accuracy demands a network that can personalize its modeling to the data, which often results in an intricate network structure, particularly within the hidden layer, potentially harboring numerous unnecessary nodes. While pursuing a simplified neural network structure, it is not prudent to directly designate a minimal number of nodes and related parameters. Subjectively determining the appropriate count of neurons for the network to accurately capture the input-to-output relationship is not a feasible approach. What is required is a rational and efficient algorithm to assist in identifying the optimal number of nodes for ELM.

Utilizing the minimization of training error and output weight within the ELM-Pinball-Huber model as the dual optimization objectives, we employ the multiobjective optimization algorithm NSGA-II to enhance the ELM model. From the derived set of Pareto front solutions, we carefully choose the most suitable solution to execute the forecasting task. The objective function for this multiobjective optimization endeavor is presented as follows:

$$\begin{cases} min & \sum_{i=1}^{N} PH(r_i) \\ min & \sum_{i=1}^{L} \mid \beta_i \mid, \end{cases} \tag{12}$$

where r_i is the training error and N represents the number of samples; we use the training error based on the new proposed Pinball-Huber loss function as one optimization objective. β is the output weight vector of the output layer in the ELM model, and L is the number of hidden layer nodes; in addition, we take the L1 norm of it as the other optimization objective.

Following the multiobjective optimization process, we arrive at the set of Pareto solutions. By representing the two optimization objectives along the horizontal and vertical axe, respectively, we observe that the solutions within the Pareto set form a U-shaped distribution. This pattern highlights the inherent trade-off between training error and output weight as optimization objectives. The solution situated at the inflection point simultaneously possesses a lower training error and output weight norm, rendering it the optimal choice for our multiobjective optimization.

Furthermore, to validate the effectiveness of the Pinball-Huber loss function, we conducted a separate comparative test. Employing various loss functions in conjunction with ELM and integrating a lasso penalty term, the composite ELM-loss function models were employed for power load forecasting within identical experimental parameters. Elaborate insights into the model's objective function and its solution procedure are provided in Appendix A.

4.2.1 The overall steps

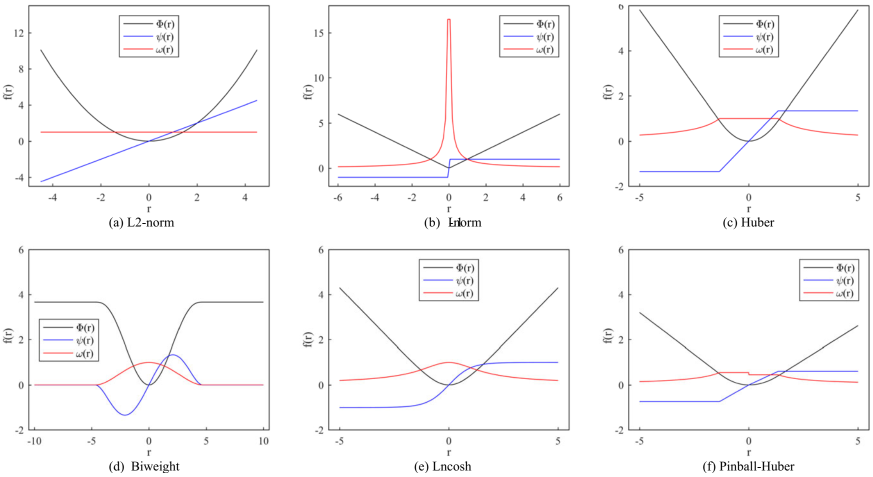
In this section, we provide a combined load forecasting model NSGA-II-ELM-Pinball-Huber. The model is ELM based on the Pinball-Huber loss function and then optimized by the multiobjective optimization algorithm NSGA-II. The steps of the model can be found in pseudocode Algorithm 1.

Algorithm 1 The NSGA-II-ELM-Pinball-Huber model.

```
Require: Hidden layer nodes $L$, input weight $\omega$, and hidden layer threshold $b$
Ensure: Training error and output weight of ELM
Set the population and iteration times
Initialize the population:
Randomly generate multiple groups of ELMs with different numbers of hidden layer nodes
for each ELM
    Select the input weight $\omega$ and hidden layer threshold $b$ randomly
    Take the proposed Pinball-Huber loss function as the objective function and solve the output weight vector $\beta = (H^T WH)^{-1} H^T WY$ by IRLS
    Get the training error $\sum_{i=1}^{N} PH(r_i)$ based on Pinball-Huber loss function and output weight $\sum_{i=1}^{L} \mid \beta_i \mid$ of ELM as the two optimization objectives
end for
Fast nondominated sort and crowding distance sort on the population and take it as the Parent
for each $i = 1, 2, 3, \cdots, gen$ do
    Tournament selection, Simulated Binary Crossover, and Polynomial mutation to produce the Offspring
    Merge Parent and Offspring
    Fast nondominated sort and crowding distance sort
    Generate new Parent by Elitist strategy
end for
Get optimized Pareto solution set of all the groups
Select the best sample from the set, that is, the ELM model with the best parameters $L, \omega, b$
```

5 Case study

This section employs the power load dataset from Taixing City in southern Jiangsu Province to validate the efficacy of the integrated NSGA-II-ELM-Pinball-Huber forecasting model within the power load system.



(a) L2-norm (b) L1-norm (c) Huber

(d) Biweight (e) Lncosh (f) Pinball-Huber

Fig. 3 The loss functions

Table 1 Loss functions and their weight functions

Loss	Loss function $\Psi(r)$	Derivative function $\phi(r) = \frac{\partial(\Psi(r))}{\partial r)}$	Weight function $\omega(r) = \frac{\phi(r)}{r}$	Default
L2 loss	$\frac{1}{2}r^2$	r	1	–
L1 loss	$\mid r \mid$	sign(r)	$\frac{1}{max(\mid r \mid,\epsilon)}, \epsilon = 10^-6$	–
Huber loss	$\begin{cases} \frac{1}{2}r^2, \mid r \mid \leq \delta \\ \mid r \mid \delta - \frac{\delta^2}{2}, \mid r \mid > \delta \end{cases}$	$\begin{cases} r, \mid r \mid \leq \delta \\ \delta sign(r), \mid r \mid > \delta \end{cases}$	$min(1, \frac{\delta}{\mid r \mid})$	1.345
Biweight loss	$\begin{cases} \frac{c^2}{6}[1-(1-(\frac{r}{c})^2)^3], \mid r \mid \leq c \\ \frac{c^2}{6}, otherwise \end{cases}$	$r(1-(\frac{r}{c})^2)^2, \mid r \mid \leq c$	$(1-(\frac{r}{c})^2)^2, \mid r \mid \leq c$	4.685
Lncosh loss	ln(cosh(r))	tanh(r)	$\frac{tanh(r)}{r}$	–
Pinball-Huber loss	$\begin{cases} \frac{1}{2}r^2\tau, 0 \leq r \leq \delta \\ \frac{1}{2}r^2(1-\tau), -\delta \leq r \leq 0 \\ (\mid r \mid \delta - \frac{\delta^2}{2})\tau, r > \delta \\ (\mid r \mid \delta - \frac{\delta^2}{2})(1-\tau), r < -\delta \end{cases}$	$\begin{cases} \tau r, 0 \leq r \leq \delta \\ (1-\tau)r, -\delta \leq r \leq 0 \\ \delta\tau sign(r), r > \delta \\ \delta(1-\tau)sign(r), r < -\delta \end{cases}$	$\begin{cases} \tau, 0 \leq r \leq \delta \\ 1-\tau, -\delta \leq r \leq 0 \\ \frac{\delta\tau}{\mid r \mid}, r > \delta \\ \frac{\delta(1-\tau)}{\mid r \mid}, r < -\delta \end{cases}$	δ, τ

5.1 Experimental setup and evaluation criteria

In this section, alongside the newly proposed Pinball-Huber loss function, various common loss functions are integrated with ELM for comparison, highlighting the performance of the novel loss function. Six loss functions, namely L2-norm, L1-norm, Huber, Biweight, Lncosh, and Pinball-Huber, are employed as the objective functions for ELM, enabling a comparison of their distinct effects.

The specific experiments involve forecasting the 49th observation based on the preceding 48 observations. Multi-step experiments, encompassing three-step, five-step, and seven-step forecasts, are conducted under consistent conditions. A choice of 200 is made for the number of hidden layer nodes in ELM, allowing for the demonstration of the compression effect from the lasso penalty within the ELM-loss function model. The entire experimentation is conducted in Matlab, utilizing Matlab2016 to compile the experimental code (Fig. 3).

We use root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) to measure the forecasting effect of the model, as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} \mid r_i \mid, \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {r_i}^2}, \tag{14}$$

and

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N} \mid \frac{r_i}{y_i} \mid. \tag{15}$$

In the three formulas, $r_i = y_i - \hat{y}_i \quad (i = 1, 2, ..., N)$ is the residual, y_i is the actual power load value, and $\hat{y}_i$ is the forecasting value, representing the forecasting value of the i-th sample (Table 1).

5.2 Taixing power load data

For the Taixing electric power data set, we carry out power load forecasting. From 2018.5.13 to 2021.8.2, the data set records the power load data every other day. In the power data set of Taixing City, there are 1,175 data points. We divide these into the training set and test set by a ratio of 8:2, in which the former includes 940 points and the latter 235 points. The specific characteristics of the data are shown in Table 2.

In Table 1, some of the six loss functions listed need to set parameters. The δ in Huber loss represents the tuning parameters, which determines how to deal with outliers.

Table 2 The descriptive statistics of Taixing data set

Taixing data set	Size	Min.	Max.	Median	Mean	Std. Dev.
Data set	1175	1210.872	2875.318	1893.728	1902.926	249.691
Training set	940	1210.872	2578.343	1842.336	1843.150	216.938
Testing set	235	1424.918	2875.318	2137.340	2142.030	228.071

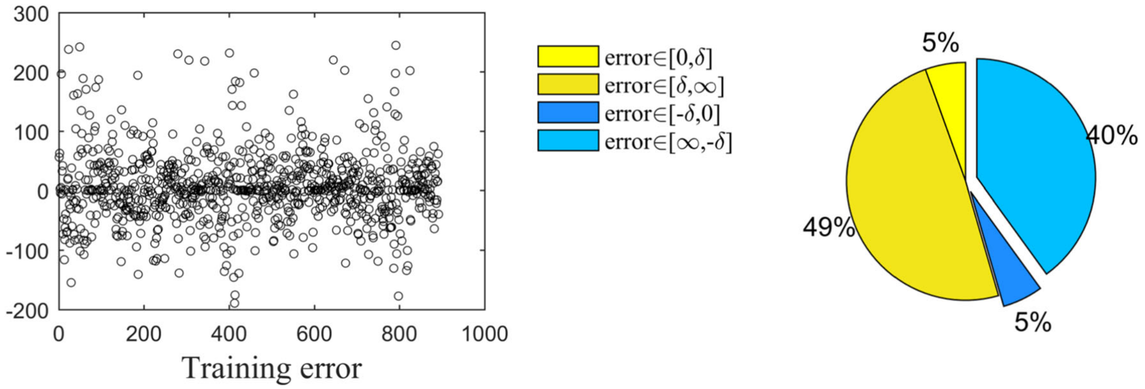


Fig. 4 Training error distribution diagram of ELM-Pinball-Huber in the Taixing data set

The hyperparameter τ in Pinball loss is the target quantile, which is used to handle the direction errors in forecasting. For the robust Pinball-Huber loss function proposed by us, both parameters δ and τ need to be set. The proper values are determined using the time series cross-validation approach, with δ and τ each choosing a random value between [0,1] and [1,2]. Taking the training error in single-step forecasting as an example for analysis, as shown in Fig. 4, the training errors have deviations and are asymmetrically distributed. The hyperparameters of the Pinball-Huber loss function obtained through the time series cross-validation method are shown in Table 3. The experimental analysis is as follows:

5.2.1 Comparisons among ELM-Pinball-Huber and ELM with other loss functions

This section presents the comparative experiments conducted on the ELM model using six distinct loss functions that are aimed at substantiating the benefits of the newly introduced Pinball-Huber loss function. Based on the three evaluation metrics-RMSE, MSE, and MAPE-outlined in Table 4, it can be deduced that the forecasting outcomes of ELM utilizing our Pinball-Huber loss function surpass those achieved with other loss functions, both in single-step and multistep forecasting scenarios. Figure 5 provides a visual representation of the ELM's performance in multistep forecasting across the six different loss functions. Hence, adopting the proposed Pinball-Huber loss function as the objective function for ELM can lead to enhanced forecasting capabilities within the power load prediction.

Table 3 The hyperparameters of Pinball-Huber loss function

Steps	Single-step	Three-step	Five-step	Seven-step
δ	1.25	1.60	1.20	1.50
τ	0.45	0.30	0.50	0.30

Furthermore, an intriguing observation emerged from the comparison among the six loss functions. The L2-norm and L1-norm loss functions exhibited subpar and unstable performance in multistep forecasting. Huber, Biweight, and Lncosh loss functions demonstrated favorable performance, but their stability in multistep experiments displayed notable fluctuations. Conversely, the ELM-Pinball-Huber consistently demonstrated the most optimal forecasting results while maintaining a relatively stable performance throughout the experiments.

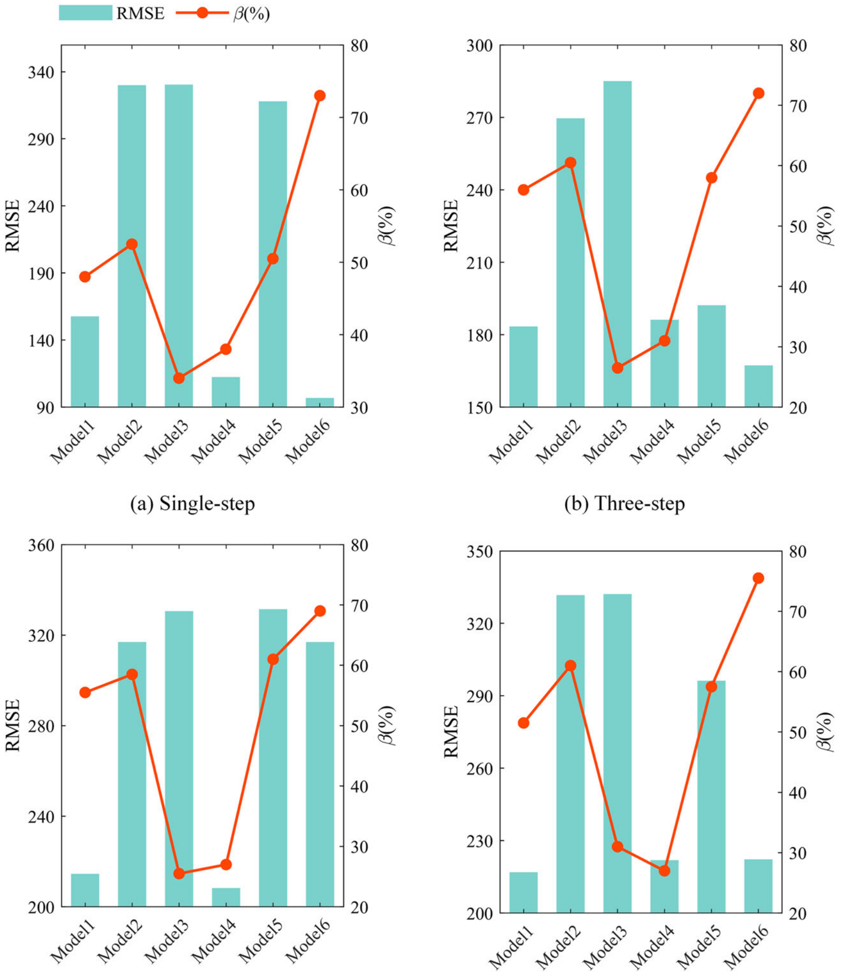
5.2.2 Comparisons of ELM with loss functions with/without multiobjective optimization

In the preceding section's comparative experiments, ELM utilizing the Pinball-Huber loss function exhibited consistent advantages in forecasting accuracy. Nonetheless, a noteworthy observation was that achieving higher precision often resulted in elevated output weights within the ELM model. This outcome could lead to intricate network structures and even overfitting issues. To ascertain the enhanced forecasting performance of ELM-Pinball-Huber through NSGA-II optimization, a comparative validation was conducted.

Table 5 presents the outcomes of multiobjective optimization for ELM using diverse loss functions. Upon comparison with the results from the pre-multiobjective optimization experiments illustrated in Table 4, the ELM, post-multiobjective optimization not only attains heightened forecasting precision, but it also substantially diminishes the output weight within the ELM model. The distribution of solutions within the Pareto solution set, along with the curve showcasing the alteration in the two optimization objectives with the number of iterations, is depicted in Fig. 6. Notably, as the number of iterations increases, the values of the two optimization objectives consistently decrease. Ultimately, within the figure, a Pareto solution is discernible, maintaining commendable values for both optimization objectives, thereby achieving elevated forecasting accuracy while concurrently

Table 4 Multistep forecasting results of ELM-loss function models in the Taixing data set

Models	Taixing data set							
	MAE	RMSE	MAPE	$\beta(\%)$	MAE	RMSE	MAPE	$\beta(\%)$
	Single-step				*Three-step*			
ELM-L2	115.337	157.701	0.055	48.0	130.577	183.425	0.063	56.0
ELM-L1	274.207	329.957	0.125	52.5	233.836	269.626	0.107	60.5
ELM-Huber	274.858	330.423	0.126	34.0	222.722	285.020	0.103	26.5
ELM-Biweight	85.986	112.454	0.041	38.0	132.345	186.186	0.064	31.0
ELM-Lncosh	258.347	317.961	0.119	50.5	143.788	192.219	0.069	58.0
ELM-Pinball-Huber	73.946	96.879	0.036	73.0	121.818	167.292	0.060	72.0
	Five-step				*Seven-step*			
ELM-L2	161.549	214.583	0.077	55.5	163.955	216.922	0.079	51.5
ELM-L1	258.779	316.949	0.119	58.5	273.857	331.715	0.125	61.0
ELM-Huber	273.836	330.600	0.125	25.5	274.930	332.128	0.126	31.0
ELM-Biweight	162.277	208.288	0.077	27.0	168.125	221.927	0.081	27.0
ELM-Lncosh	273.831	331.458	0.125	61.0	234.569	296.230	0.108	57.5
ELM-Pinball-Huber	258.779	316.949	0.119	69.0	163.492	222.261	0.077	75.5



(a) Single-step

(b) Three-step

(c) Five-step

(d) Seven-step

Fig. 5 Multistep forecasting results of ELM-Pinball-Huber and ELM with other loss functions in the Taixing data set

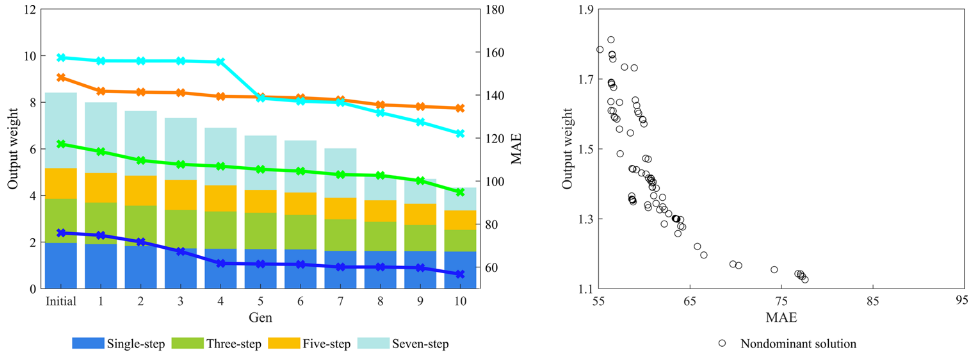
Table 5 Multistep forecasting results of NSGA-II-ELM-loss function models in the Taixing data set

Steps	Models	RMSE		Output weight	Hidden neurons
		Training	Test		
Single-step	NSGA-II-ELM-L2	67.993	96.143	7.974	60
	NSGA-II-ELM-L1	67.499	98.078	7.622	50
	NSGA-II-ELM-Huber	68.335	109.475	7.556	55
	NSGA-II-ELM-Biweight	69.071	114.421	7.644	55
	NSGA-II-ELM-Lncosh	71.925	86.592	6.402	45
	NSGA-II-ELM-PH	66.545	84.273	7.427	50
Three-step	NSGA-II-ELM-L2	98.761	139.295	7.957	60
	NSGA-II-ELM-L1	117.152	146.849	3.770	25
	NSGA-II-ELM-Huber	120.028	150.411	2.491	15
	NSGA-II-ELM-Biweight	117.582	163.121	3.215	20
	NSGA-II-ELM-Lncosh	124.808	199.248	2.957	20
	NSGA-II-ELM-PH	113.192	129.229	6.934	45
Five-step	NSGA-II-ELM-L2	123.072	181.270	4.929	35
	NSGA-II-ELM-L1	121.165	179.277	7.597	35
	NSGA-II-ELM-Huber	124.281	171.076	6.407	50
	NSGA-II-ELM-Biweight	122.325	177.307	5.344	50
	NSGA-II-ELM-Lncosh	120.771	185.114	6.386	50
	NSGA-II-ELM-PH	120.812	169.957	6.235	45
Seven-step	NSGA-II-ELM-L2	126.301	206.447	12.118	70
	NSGA-II-ELM-L1	146.658	210.009	3.016	20
	NSGA-II-ELM-Huber	147.564	202.533	2.511	20
	NSGA-II-ELM-Biweight	145.008	213.595	3.167	20
	NSGA-II-ELM-Lncosh	147.561	219.394	2.700	20
	NSGA-II-ELM-PH	143.977	196.647	8.706	50

preserving smaller output weights. This simplification considerably reduces the intricacies of the model network. The streamlined ELM necessitates fewer hidden layer nodes, enhancing its generalization capabilities. Moreover, we have also observed that NSGA-II can enhance the performance of various ELM-loss function combinations, indicating its wide applicability for ELM. Notably, the amalgamation of the Pinball-Huber loss function and ELM, following NSGA-II optimization, demonstrates the most optimal performance. The multistep ahead forecasting curves for the NSGA-II-ELM-Pinball-Huber model are displayed in Fig. 7.

5.2.3 Comparisons among NGSA-II-ELM-Pinball-Huber and comparative models

To assess the predictive performance of the NSGA-II-ELM-Pinball-Huber model, we conduct comparative experiments



(a) The iteration curve of NSGA-II

(b) The distribution of nondominant solutions

Fig. 6 Distribution of Pareto solution set and optimization iteration curves of NSGA-II-ELM-Pinball-Huber in the Taixing data set

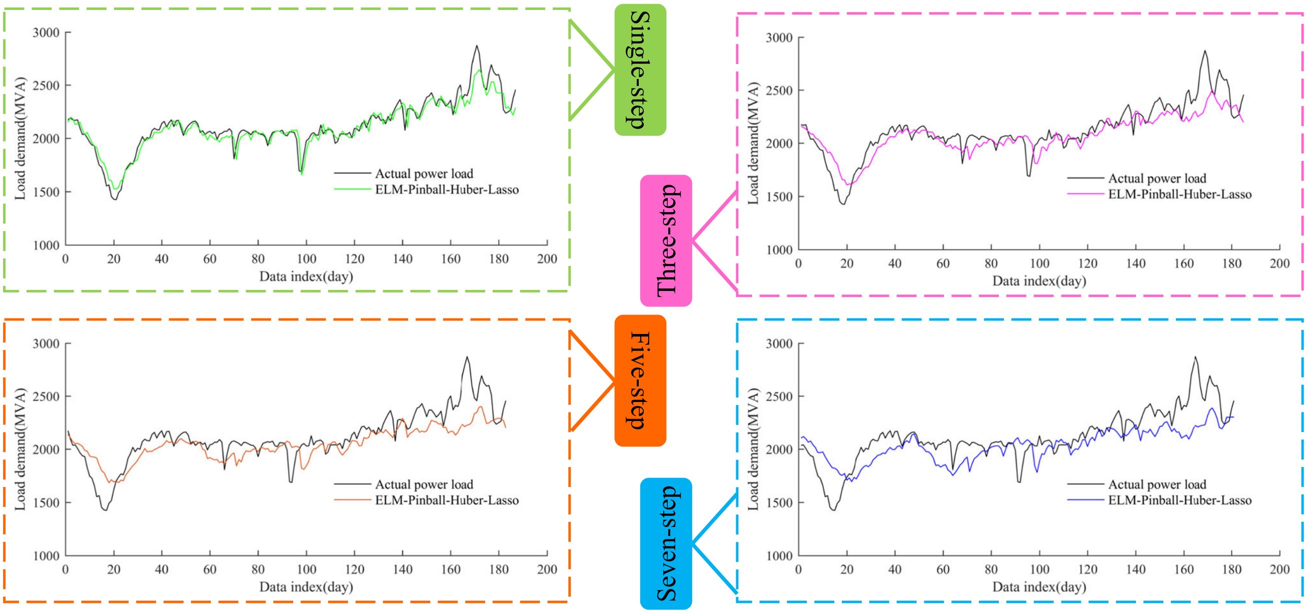


Fig. 7 Multi-step forecasting curves of NSGA-II-ELM-Pinball-Huber in Taixing data set

Table 6 Parameter of comparative models

Model	Parameter name	Parameter value
LSTM	number of layers, units	[3, 64]
GRU	number of layers, units	[3, 64]
CNN-BiLSTM-Attention	number of layers, kernel size, units	[9, 1, 64]

Table 7 Comparisons among NGSA-II-ELM-Pinball-Huber and other models

Steps	Models	RMSE	Output weight
Single-step	LSTM	184.97	11.06
	GRU	184.58	11.15
	CNN-BiLSTM-Attention	273.91	96.45
	NSGA-II-ELM-PH	84.27	7.43
Three-step	LSTM	221.56	29.94
	GRU	228.79	31.56
	CNN-BiLSTM-Attention	257.66	287.01
	NSGA-II-ELM-PH	129.23	6.93
Five-step	LSTM	229.74	47.26
	GRU	256.36	49.84
	CNN-BiLSTM-Attention	259.29	480.26
	NSGA-II-ELM-PH	169.96	6.24
Seven-step	LSTM	256.58	67.89
	GRU	314.94	63.85
	CNN-BiLSTM-Attention	258.21	669.97
	NSGA-II-ELM-PH	196.65	8.71

Table 8 The Wilcoxon signed-rank test of the compared models

Steps	NSGA-II-ELM-PH vs LSTM		NSGA-II-ELM-PH vs GRU		NSGA-II-ELM-PH vs CNN-BiLSTM-Attention	
	p-value	h	p-value	h	p-value	h
Single-step	<0.05	1	<0.05	1	<0.05	1
Three-step	<0.05	1	<0.05	1	<0.05	1
Five-step	<0.05	1	<0.05	1	>0.05	0
Seven-step	<0.05	1	<0.05	1	>0.05	0

with three models: LSTM, GRU, and CNN-BiLSTM-Attention. Brief descriptions of these models are as follows:

(1) LSTM model: LSTM, an improved variant of traditional RNN, effectively captures the semantic relationships in long sequences, mitigating gradient vanishing or exploding issues. LSTM features a more complex structure [48].
(2) GRU model: Introduced by Cho et al. [49] in 2014, the GRU neural network addresses the gradient vanishing problem in standard recurrent neural networks and shares a similar design philosophy with LSTM.
(3) CNN-BiLSTM-Attention model [50]: This model employs complex mathematical operations in the convolutional and pooling layers of the convolutional neural networks (CNN) to extract the spatial features of the input variables. The multi-head attention layer minimizes irrelevant feature impact, enhancing the extracted features. The BiLSTM layer models trend information in time series, generating a probability model for prediction distribution.

We adjust the hyperparameters of each model to achieve optimal performance, as shown in Table 6. The evaluation metrics for forecasting performance are presented in Table 7.

The predictive performance of LSTM and GRU is similar, showing close values for RMSE and the output weight. Comparing the predictive performance evaluation metrics of LSTM, GRU, and CNN-BiLSTM-Attention with NSGA-II-ELM-Pinball-Huber, the RMSE of the proposed model was always lower than that of the three comparison models. Particularly in single-step forecasting, the prediction error of NSGA-II-ELM-Pinball-Huber (RMSE=84.27) was significantly smaller than that of the three comparative models (RMSE=184.87, 184.58, 273.91). These results indicate that the proposed model effectively captured the changing trends in power load data in both the spatial and temporal dimensions. Furthermore, the proposed model maintained a stable structure in multistep forecasting.

Finally, to verify whether the NSGA-II-ELM-Pinball-Huber model significantly improves predictive accuracy in power load forecasting compared with other models, we conducted the Wilcoxon signed-rank test [51]. The significance level for the one-tailed test was set at $\alpha = 0.05$. The original hypothesis posited that there would be no significant difference in the predictive results between our model and the comparative models in power load forecasting. If the p-value is less than 0.05, the original hypothesis will be rejected (h=1). The predicted values of the proposed model and three comparison models were turned into Wilcoxon signed-rank tests separately in multistep forecasting from 1 to 7 steps. The results are shown in Table 8.

6 Conclusion and future work

In the current paper, we have introduced a robust Pinball-Huber loss function that demonstrates remarkable resistance to outliers and substantially reduces the likelihood of overfitting. This loss function effectively manages outliers and asymmetrical noise within the dataset, serving as the objective function for training the ELM model. Given the ELM's susceptibility to preset parameters' influence, and aiming to ensure forecasting accuracy while maximizing and simplifying the ELM network structure, as well as preventing the squandering of training time and the emergence of overfitting because of an excessive number of hidden layer nodes, we employed the NSGA-II algorithm for the optimization of both training errors and output weights within the ELM model. The combined NSGA-II-ELM Pinball-Huber model was then employed for power load forecasting in the context of Taixing City. By employing the multi-objective optimization algorithm NSGA-II, we acquired the Pareto optimal solution set for the number of hidden layer nodes in the ELM model, enabling an in-depth analysis of the forecasting outcomes. Our analysis of the experimental outcomes revealed that the performance of the suggested Pinball-Huber loss function within the ELM framework surpassed that of other loss functions. Moreover, the NSGA-II algorithm effectively enhanced the performance of diverse ELM-loss function combinations. The innovative combined approach, NSGA-II-ELM Pinball-Huber model, can be seen as a promising and effective method for power load forecasting, offering a novel solution to this domain.

Multiobjective optimization greatly improves the predictive performance of the model, but it takes up a significant amount of computational resources. In the future, we aim to delve deeper into simplifying the computational resources and time required for training the proposed method, which is crucial for the widespread applicability of the model. Furthermore, this forecasting model can only provide predicted values for future power loads, and recent research has focused on uncertain predictions. Future studies will delve deeper into the probability predictions of the model, which holds significant value for practical applications in power systems [52, 53].

Appendix

A The ELM-loss function model

The combined ELM-loss function is a single objective model, and its mathematical model can be written as follows:

$$\begin{aligned} min \quad & C\sum_{i=1}^{N} PH(r_i) + \sum_{j=1}^{L} \mid \beta_j \mid \\ s.t. \quad & h(x_i) = y_i - r_i, i = 1, 2, ..., N \end{aligned} \tag{16}$$

where r_i represents the training error of the sample and $\sum_{i=1}^{N} PH(r_i)$ is the total error under Pinball-Huber loss function of N different training samples, here representing experience risk. $\sum_{j=1}^{L} \mid \beta_j \mid$ is a lasso penalty term, representing the complexity of the model. $C > 0$ is called the regularization parameter or the penalty parameter and is used to balance empirical risk and model complexity.

Lagrangian multipliers are introduced for each equality constraint condition in the model, and the Lagrangian function is constructed to transform it into an unconstrained optimization problem:

$$\begin{aligned} L(\beta, r, \alpha) = C\sum_{i=1}^{N} PH(r_i) + \sum_{j=1}^{L} \mid \beta_j \mid \\ - \sum_{i=1}^{N} \alpha_i (h(x_i)\beta - y_i - r_i). \end{aligned} \tag{17}$$

where $\alpha = [\alpha_i, \alpha_2, ..., \alpha_N]$ is the Lagrange multiplier vector.

We solve (17), and we obtain the output weight vector β as follows:

$$\beta = \begin{cases} (\frac{W_L}{C} + H^T W_N H)^{-1} H^T W_N Y, & N \geq L \\ W_L^{-1} H^T (\frac{I}{C} + W_N H W_L^{-1} H^T)^{-1} W_N Y, & N < L \end{cases} \tag{18}$$

where W_N is the sample weight matrix and W_L is the weight matrix of hidden nodes. The details of w_i of the loss function can be found in Table 1. Their specific forms are as follows:

$$W_N = \begin{bmatrix} w(r_1) & 0 & \cdots & 0 \\ 0 & w(r_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(r_N) \end{bmatrix}$$

$$W_L = \begin{bmatrix} \frac{1}{max(|\beta_1|, \epsilon)} & 0 & \cdots & 0 \\ 0 & \frac{1}{max(|\beta_2|, \epsilon)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{max(|\beta_L|, \epsilon)} \end{bmatrix}$$

In general, the specific steps of the ELM-Pinball-Huber model are as follows:

Step 1: Initialize the relevant parameters w, b, and L of the ELM-Pinball-Huber model.

Step 2: Calculate the output weight vector β by (18).

Step 3: Update sample weight matrix W_N and hidden nodes' weight matrix W_L.

Step 4: Repeat steps 2-3 until β converges; then, obtain the trained ELM-Pinball-Huber model.

Step 5: Substitute the test set into the trained model to get the forecasting results.

Similar to the above ELM-Pinball-Huber model, we can combine the loss functions in Table 1 with ELM, respectively, to compare their performance.

Acknowledgements The work is supported by the Australian Research Council project (grant number DP160104292), the National Natural Science Foundation of China under Grant 61873130 and Grant 61833011, the Natural Science Foundation of Jiangsu Province under Grant BK20191377, the 1311 Talent Project of Nanjing University of Posts and Telecommunications, and "Chunhui Program" Collaborative Scientific Research Project (202202004).

Author Contributions Yang Yang: Writing - review & editing, Funding acquisition. Hao Lou: Software, Visualization, Formal analysis, Writing-original draft. Zijin Wang: Writing-review & editing. Jinran Wu: Supervision, Formal analysis, Writing-original draft, Writing-review & editing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data Availability The data are available with a reasonable request.

Declarations

Competing of interest The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Ethical approval This article does not contain any studies on human participants or animals.

References

1. Li K, Huang W, Hu G, Li J (2023) Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network. Energy Build 279:112666
2. Wen L, Zhou K, Yang S, Lu X (2019) Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting. Energy 171:1053–1065
3. Lebotsa ME, Sigauke C, Bere A, Fildes R, Boylan JE (2018) Short term electricity demand forecasting using partially linear additive quantile regression with an application to the unit commitment problem. Appl Energy 222:104–118
4. He F, Zhou J, Mo L, Feng K, Liu G, He Z (2020) Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest. Appl Energy 262:114396
5. Gupta D, Hazarika BB, Berlin M (2020) Robust regularized extreme learning machine with asymmetric Huber loss function. Neural Comput Appl 32(16):12971–12998
6. Zhang J, Siya W, Zhongfu T, Anli S (2023) An improved hybrid model for short term power load prediction. Energy 268:126561
7. Wang J, Zhang L, Li Z (2022) Interval forecasting system for electricity load based on data pre-processing strategy and multi-objective optimization algorithm. Appl Energy 305:117911
8. Wu F, Cattani C, Song W, Zio E (2020) Fractional ARIMA with an improved cuckoo search optimization for the efficient Short-term power load forecasting. Alex Eng J 59(5):3111–3118
9. Dudek G (2016) Pattern-based local linear regression models for short-term load forecasting. Electr Power Syst Res 130:139–147
10. Lee CM, Ko CN (2011) Short-term load forecasting using lifting scheme and ARIMA models. Expert Syst Appl 38(5):5902–5911
11. Wang J, Xing Q, Zeng B, Zhao W (2022) An ensemble forecasting system for short-term power load based on multi-objective optimizer and fuzzy granulation. Appl Energy 327:120042
12. Voyant C, Notton G, Kalogirou S, Nivet ML, Paoli C, Motte F et al (2017) Machine learning methods for solar radiation forecasting: A review. Renew Energy 105:569–582
13. Kim MK, Kim YS, Srebric J (2020) Predictions of electricity consumption in a campus building using occupant rates and weather elements with sensitivity analysis: Artificial neural network vs. linear regression. Sustain Cities and Soc 62:102385
14. Hopfield JJ (1988) Artificial neural networks. IEEE Circ Devices Mag 4(5):3–10
15. Awad M, Khanna R, Awad M, Khanna R (2015) Support vector regression. Theories, concepts, and applications for engineers and system designers, Efficient learning machines, pp 67–80
16. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541), vol 2. Ieee; pp 985–990
17. Biswas MR, Robinson MD, Fumo N (2016) Prediction of residential building energy consumption: A neural network approach. Energy 117:84–92
18. Trairat P, Banjerdpongchai D (2022) Multi-objective optimal operation of building energy management systems with thermal and battery energy storage in the presence of load uncertainty. Sustainability 14(19):12717
19. Tian X, Zou Y, Wang X, Tseng M, Li H, Zhang H (2022) Improving the efficiency and sustainability of intelligent electricity inspection: IMFO-ELM Algorithm for Load Forecasting. Sustainability 14(21):13942
20. Sajjadi S, Shamshirband S, Alizamir M, Yee L, Mansor Z, Manaf AA et al (2016) Extreme learning machine for prediction of heat load in district heating systems. Energy Build 122:222–227
21. Ding S, Xu X, Nie R (2014) Extreme learning machine and its applications. Neural Comput Appl 25(3):549–556
22. Zhou Y, Zhou N, Gong L, Jiang M (2020) Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine. Energy 204:117894
23. Ni Q, Zhuang S, Sheng H, Kang G, Xiao J (2017) An ensemble prediction intervals approach for short-term PV power forecasting. Solar Energy 155:1072–1083
24. Han Y, Wang N, Ma M, Zhou H, Dai S, Zhu H (2019) A PV power interval forecasting based on seasonal model and nonparametric estimation algorithm. Solar Energy 184:515–526
25. Chen X, Yu R, Ullah S, Wu D, Li Z, Li Q et al (2022) A novel loss function of deep learning in wind speed forecasting. Energy 238:121808
26. Chen J, Zeng Y, Li Y, Huang GB (2020) Unsupervised feature selection based extreme learning machine for clustering. Neurocomputing 386:198–207
27. Yang Y, Tao Z, Qian C, Gao Y, Zhou H, Ding Z, et al (2022) A hybrid robust system considering outliers for electric load series forecasting. Applied Intelligence, pp 1–23
28. Wang J, Zhu H, Cheng F, Zhou C, Zhang Y, Xu H et al (2023) A novel wind power prediction model improved with feature enhancement and autoregressive error compensation. J Clean Prod 420:138386
29. Wj Niu, Zk Feng, Li Ss Wu, Hj Wang Jy (2021) Short-term electricity load time series prediction by machine learning model via feature selection and parameter optimization using hybrid cooperation search algorithm. Environ Res Lett 16(5):055032
30. Shang Z, He Z, Song Y, Yang Y, Li L, Chen Y (2020) A novel combined model for short-term electric load forecasting based on whale optimization algorithm. Neural Process Lett 52:1207–1232
31. Xie K, Yi H, Hu G, Li L, Fan Z (2020) Short-term power load forecasting based on Elman neural network with particle swarm optimization. Neurocomputing 416:136–142
32. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi A (2020) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
33. Kaboli SHA, Fallahpour A, Selvaraj J, Rahim N (2017) Long-term electrical energy consumption formulating and forecasting via optimized gene expression programming. Energy 126:144–164
34. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338
35. Luo L, Li H, Wang J, Hu J (2021) Design of a combined wind speed forecasting system based on decomposition-ensemble and multi-objective optimization approach. Appl Math Model 89:49–72
36. Yang Y, Zhou H, Gao Y, Wu J, Wang YG, Fu L (2022) Robust penalized extreme learning machine regression with applications in wind speed forecasting. Neural Comput Appl 34(1):391–407
37. Huber PJ (1973) Robust regression: asymptotics, conjectures and Monte Carlo. The Annals of Statistics, pp 799–821

38. Wang K, Zhong P (2014) Robust non-convex least squares loss function for regression with outliers. Knowl-Based Syst 71:290–302
39. Yang X, Tan L, He L (2014) A robust least squares support vector machine for regression and classification with noise. Neurocomputing 140:41–52
40. Beaton AE, Tukey JW (1974) The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. Technometrics 16(2):147–185
41. Wang X, Jiang Y, Huang M, Zhang H (2013) Robust variable selection with exponential squared loss. J Am Stat Assoc 108(502):632–643
42. Karal O (2017) Maximum likelihood optimal and robust Support Vector Regression with lncosh loss function. Neural Netw 94:1–12
43. Huang GB, Siew CK (2005) Extreme learning machine with randomly assigned RBF kernels. Int J Inf Technol 11(1):16–24
44. Huang G, Huang GB, Song S, You K (2015) Trends in extreme learning machines: A review. Neural Netw 61:32–48
45. Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: A tutorial. Reliab Eng & Syst Saf 91(9):992–1007
46. Sampson JR.: Adaptation in natural and artificial systems (John H. Holland). Society for Industrial and Applied Mathematics
47. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: International conference on parallel problem solving from nature. Springer, pp 849–858
48. Wang J, Zhu H, Zhang Y, Cheng F, Zhou C (2023) A novel prediction model for wind power based on improved long short-term memory neural network. Energy 265:126283
49. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078
50. Zhang YM, Wang H (2023) Multi-head attention-based probabilistic CNN-BiLSTM for day-ahead wind speed forecasting. Energy 278:127865
51. Li D, Jiang MR, Li MW, Hong WC, Xu RZ (2023) A floating offshore platform motion forecasting approach based on EEMD hybrid ConvLSTM and chaotic quantum ALO. Applied Soft Computing, pp 110487
52. Hong T, Fan S (2016) Probabilistic electric load forecasting: A tutorial review. Int J Forecast 32(3):914–938
53. Zhang Y, Wang J, Wang X (2014) Review on probabilistic forecasting of wind power generation. Renew Sust Energ Rev 32:255–270

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.