A hybrid robust system considering outliers for electric load series forecasting

Yang, Yang, Tao, Zhenghang, Qian, Chen, Gao, Yuchao, Zhou, Hu, Ding, Zhe and Wu, Jinran

Noname manuscript No.
(will be inserted by the editor)

A hybrid robust system considering outliers for electric load series forecasting

Yang Yang · Zhenghang Tao · Chen Qian · Yuchao Gao · Hu Zhou · Zhe Ding · Jinran Wu

Received: DD Month YEAR / Accepted: DD Month YEAR

Abstract Electric load forecasting has become crucial to the safe operation of power grids and cost reduction in the production of power. Although numerous electric load forecasting models have been proposed, most of them are still limited by poor effectiveness in the model training and a sensitivity to outliers. The limitations of current methods may lead to extra operational costs of a power system or even disrupt its power distribution and network safety. To this end, we propose a new hybrid load-forecasting model, which is based on a robust extreme-learning machine and an improved whale optimization algorithm. Specifically, Huber loss, which is insensitive to outliers, is proposed as the objective function in extreme learning machine (ELM) training. In addition, an improved whale optimization algorithm is designed for the robust ELM training, in which a cellular automaton mechanism is used to enhance the local search. To verify our improved whale optimization algorithm, some experiments were then conducted based on seven benchmark test functions. Due to the enhancement of the local search, the improved optimizer was around 7% superior to the basic. Finally, our proposed hybrid forecasting model was validated by two real electric load datasets (Nanjing and New South Wales), and the experimental results confirmed that the proposed hybrid load-forecasting model could achieve satisfying improvements in both datasets.

Keywords Outilers · Whale optimization algorithm · Cellular automata · Load forecasting · Robust regression

Y. Yang, Z. Tao, C. Qian, Y. Gao, H. Zhou
College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, P. R. China

Z. Ding
School of Computer Science, Queensland University of Technology, Australia

J. Wu
School of Mathematical Science, Queensland University of Technology, Australia
E-mail: jinran.wu@hdr.qut.edu.au;

1 Introduction

The power industry is an important large-scale foundation industry, and it profoundly affects all other industrial areas. Electricity is consumed shortly after it is generated. Over-generated electrical energy results in a waste of resources, while an insufficient amount of generated electrical energy causes electric outages; the consideration of both conditions affect further industrial production [1, 2]. Therefore, it is desirable to effectively minimize or eliminate over- or under-generation, and one way is through electric load forecasting. Moreover, improving the accuracy of electric load forecasting can further reduce electricity production costs. According to [3], for every 1% increase in the accuracy of electric forecasting, production costs can be reduced by 0.35%. However, since power systems are extremely complex and electricity consumption is uncertain [4], implementing accurate electric load forecasting is crucial but tricky [5, 6].

Electric load forecasting can be divided into a very-short-term load forecast, short-term load forecast (STLF) [7, 8, 9], and long-term load forecast [10], according to the length of the relevant forecasting time [11]. Among these, because of the great practicability of STLF, this paper focuses on this type of forecast. Many methods have been put forward to improve load forecasting. Current popular methods can be classified into two main categories [12, 13, 14]: traditional statistical methods

and artificial intelligence methods. Statistical methods include time-series analysis, trend extrapolation, and regression prediction methods, such as quantile regression [15] and fuzzy-based regression [16]. For example, [17] proposed a new autoregressive integrated moving average (ARIMA) model for electric load forecasting. However, traditional methods perform well when fitting load data with well-studied prediction laws, such as for linear or exponential data. Consequently, tremendous errors are unavoidable in using those methods when the laws cannot be satisfied by the data.

Since the 1980s, artificial intelligence methods have been extensively researched [18, 8, 19, 20, 21, 22, 23]; they can solve complex load forecasting by extracting the nonlinear relationships within the provided data. Many artificial intelligence methods have since been employed in electric load forecasting [24, 25], such as neural networks [26], and support vector machines (SVMs) [27, 28]. For instance, [29] presented a hybrid SVM model based on differential-based empirical mode decomposition and auto-regression. [30] proposed a multistep approach based on phase space reconstruction and and a SVM. [31] developed a new model that combined ensemble empirical mode decomposition, an extreme learning machine (ELM), and grasshopper optimization algorithm. [32] combined a discrete wavelet transform, particle swarm optimization, and radial basis function neural network for load forecasting; [33] used ensemble empirical mode decomposition, multivariable linear regression and long short-term memory neural network algorithms for the same purpose. [34] designed a model to account for the feature extraction and improved the general regression neural network. [35] used Chongqing's electricity data to make a short-term electric load forecast through the implementation of a multi-layer bidirectional recurrent neural network. More applications of deep learning in load forecasting can be found in some recent references [36, 37, 38, 39]. In addition to these artificial intelligence methods, ELMs are also widely used in load forecasting; they are emerging generalized single-hidden-layer feed-forward neural-network learning algorithms and can randomly generate hidden variable parameters to calculate output weights [40]. Thus, compared with shallow learning systems, ELM is more efficient with less computation cost and has great generalization.

In the investigated neural networks, mean square error ($L2$ loss) is often recommended as the loss function, as it can obtain good performance in typical data modeling. However, $L2$ loss will bring some unreliability to predictions from neural networks for modeling data when there are many outliers. Therefore, it can be said that the neural network training lacks robustness. In the loss function design, rather than $L2$, the least absolute loss ($L1$ loss) can be used, but it has a huge gradient that brings poor convergence. Therefore, with a parameter τ, [41] proposed a Huber loss function, which combines the $L1$ loss and $L2$ loss for the model training.

The trained parameters in the neural network's structure are a major determinant of the forecasting performance. Due to the properties of the loss function, the neural network training often meets some problems of optimization. To address this issue, various meta-heuristic optimization methods have been proposed for the model training [42, 43, 18], such as swarm intelligence [44], a cuckoo search algorithm [45], and a particle swarm optimization algorithm [46]. Moreover, in recent years, many new meta-heuristic algorithms have been proposed, such as the grey wolf optimizer (GWO), dragonfly [47], ant lion optimizer [48], and moth-flame optimization algorithms [49]. In addition to those, the whale optimization algorithm (WOA) is one of the new meta-heuristic algorithms that has emerged recently; it is based on the hunting behavior of humpback whales. In [50], the WOA was tested by 29 mathematical optimization problems and six structural optimization problems, and the experimental results demonstrated that the WOA has undeniable advantages to other optimization algorithms [51]. However, most of the investigated meta-heuristic optimization algorithms have poor convergence; thus, some new strategies have been developed to enhance the local search post hoc. One of the most effective methods is cellular automata (CA), which provides a neighbor structure for the optimization algorithm. Information exchange is promoted between the neighboring structures and achieves good local search results, while individual processes in the different communities search in parallel for solutions. In combination with GWO, the strategy has achieved good performance for some complex optimization problems [52].

In summary, current electric forecasting requirements are becoming more demanding from the following perspectives: (1) a more highly accurate prediction is required for modern industrial applications; (2) a robust loss function needs to be incorporated to develop machine learning frameworks that can handle electric load data that contain outliers; and (3) more advanced optimization methods are required to improve the model training.

Regarding the issues above, this paper proposes a new hybrid load forecasting model, a Huber-ELM optimized by an improved WOA algorithm (designated as CA-WOA-Huber-ELM). In order to improve the robustness of the ELM, a robust ELM is proposed in which Huber loss is selected as the objective function for

the ELM training, and a cross-validation method is employed to search for the tuning parameter in the Huber loss. During the robust ELM, we present an improved WOA to search the input weights and thresholds of the hidden layer in an ELM by introducing a cellular automaton to the traditional WOA. Experimental results validated that our new hybrid load-forecasting model can effectively improve the prediction performance. The contributions of this paper are listed as follows:

- A new hybrid model is proposed to improve load-forecasting accuracy, which combines a robust ELM with an improved WOA. Specifically, an improved WOA is employed to search the weights and thresholds in our proposed robust ELM, it provides good training results for the load forecasting.
- An improved robust ELM is developed to handle data with outliers. Due to its great properties, Huber loss is incorporated into the ELM as the objective function for its training.
- An improved WOA algorithm is designed, which integrates a cellular automaton with the traditional algorithm. In the new optimizer, a cellular automaton is developed in the WOA to improve the exploitation in local space and enhance the convergence of the optimizing process.

The rest of this paper is organized as follows. In Sect. 2, we review the basic ELM and propose our robust ELM. Next, the improved WOA is presented in Sect. 3. Sect. 4 then illustrates our proposed hybrid load-forecasting model and introduces the procedure of the model training with cross-validation for the tuning parameter τ. After that, seven benchmark functions are employed to validate the effectiveness of our improved WOA in Sect. 5. In Sect. 6, the testing of the proposed hybrid load-forecasting model, CA-WOA-Huber-ELM, by using two datasets from Nanjing and New South Wales is described, along with four new algorithms added for comparative examples. Sect. 7 concludes the paper.

2 Robust Extreme Learning Machine

2.1 Extreme Learning Machine

Unlike a traditional neural network, an ELM is a type of single-hidden-layer feed-forward neural network (SLFN) that randomly selects its input weights and thresholds [40]. Typically, an ELM consists of n input nodes, l hidden nodes, m output nodes, and activation functions [53, 20]. The hidden layer output matrix H is as follows:

$$H = \begin{bmatrix} g(\omega_1 \cdot u_1 + b_1)\, g(\omega_2 \cdot u_1 + b_2) \cdots g(\omega_l \cdot u_1 + b_l) \\ g(\omega_1 \cdot u_2 + b_1)\, g(\omega_2 \cdot u_2 + b_2) \cdots g(\omega_l \cdot u_2 + b_l) \\ \vdots \qquad\qquad \vdots \qquad\qquad \ddots \qquad \vdots \\ g(\omega_1 \cdot u_n + b_1) g(\omega_2 \cdot u_n + b_2) \cdots g(\omega_l \cdot u_n + b_l) \end{bmatrix}_{n \times l.} \tag{1}$$

The output T of the ELM is then as follows:

$$\begin{aligned} T = \left[t_1, t_2, \cdots, t_n\right]_{m \times n} &= \begin{bmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{mj} \end{bmatrix}_{m \times 1} \\ &= \begin{bmatrix} \sum_{i=1}^{t} \beta_{i1} \cdot g(\omega_i \cdot u_j + b_i) \\ \sum_{i=1}^{t} \beta_{i2} \cdot g(\omega_i \cdot u_j + b_i) \\ \vdots \\ \sum_{i=1}^{t} \beta_{im} \cdot g(\omega_i \cdot u_j + b_i) \end{bmatrix}_{m \times 1} \quad (j = 1, 2, \cdots, n). \end{aligned} \tag{2}$$

According to two theorems in [54], when the activation function is differentiable, it is not necessary to adjust all the ELM parameters. The input weights ω and the thresholds in the hidden layer b can be randomly chosen in the ELM structure. Finally, the solution can be obtained as follows:

$$\hat{\beta} = H^{-1} T', \tag{3}$$

where H^{-1} is the Moore-Penrose generalized inverse of the hidden layer output matrix H.

2.2 Regression Loss Function

In this subsection, to improve the ELM, three regression loss functions are introduced, namely, $L2$ loss, $L1$ loss, and $Huber$ loss.

2.2.1 L2 Loss

$L2$ loss is the average of the squared errors between the predicted data and the original data.

For the model training using $L2$ loss, when the error is large, the gradient decreases rapidly; when the error is small, the gradient decreases slowly, which is beneficial to the convergence of the model training.

Here, the $L2$ loss is defined as follows:

$$L2 = \frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2, \tag{4}$$

where M represents the number of output samples in the training set, y_i represents the expected output of the training set, and $\hat{y}_i$ represents the predicted output of the training set. However, for data with outliers, the $L2$ loss cannot guarantee the reliability of model performance.

2.2.2 L1 Loss

Here, $L1$ loss is robust to outliers, which represents the average error margin of the predicted value, without considering the direction of the error.

Compared with $L2$ loss, $L1$ loss overcomes the problem of being greatly affected by outliers, but the convergence speed is slower than with $L2$ loss.

Mean absolute error loss is defined as follows:

$$L1 = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i|. \tag{5}$$

In addition, at a certain point it is not differentiable.

2.2.3 Huber Loss

Huber loss is a parameterized loss function for regression problems [55]. Its advantage is that it can enhance the robustness of $L2$ to outliers. Huber loss reduces the degree of punishment for outliers, so Huber loss is a commonly used robust regression loss function. It is defined as follows:

$$\rho_\tau(\epsilon) = \begin{cases} \frac{1}{2}\epsilon^2 & |\epsilon| \le \tau \\ |\epsilon|\tau - \frac{\tau^2}{2} & |\epsilon| > \tau \end{cases}, \tag{6}$$

where τ is a tuning parameter ($\tau > 0$), and ϵ is the error between the expected output and the predicted output.

Compared with $L2$ and $L1$, Huber loss maintains its differentiability and is less sensitive to outliers in data. It degrades to $L2$ as τ goes to zero, and it degrades to $L2$ as τ goes to infinity. In fact, Huber loss combines the advantages of $L2$ and $L1$. It will fall near the minimum value due to the reduction of the gradient, which makes the result more accurate. It is more robust to outliers than $L2$. Therefore, the choice of τ is very important, and cross-validation is usually employed to obtain the parameter.

2.3 Robust Extreme Learning Machine

The existence of outliers influences the prediction effect of the ELM. Because Huber loss is more robust to outliers than the squared error loss function, it is differentiable at zero. Therefore, on the basis of the traditional ELM, we introduce the Huber loss function and use the minimum average Huber loss as the objective function to train the parameters of the ELM. The model has better robustness in processing data with outliers.

According to robustness analysis theory, our proposed robust ELM takes the minimum Huber loss function value as the goal for training, which substantially improves the robustness of the ELM in improving the prediction accuracy. The goal programming model of our robust extreme learning machine is as follows:

$$\min_{\omega,b} \quad MHL = \frac{1}{M}\sum_{i=1}^{M}\rho_\tau(\hat{y}_i - y_i), \tag{7}$$

$$\text{s.t.} \quad \rho_\tau(\epsilon) = \begin{cases} \frac{1}{2}\epsilon^2 & |\epsilon| \le \tau, \\ |\epsilon|\tau - \frac{\tau^2}{2} & |\epsilon| > \tau; \end{cases} \tag{8}$$

$$\omega_i = [\omega_{i1}, \omega_{i2}, \cdots, \omega_{in}]^T; \tag{9}$$

$$b = [b_1, b_2, \cdots, b_l]^T. \tag{10}$$

3 Improved Whale Optimization Algorithm

In this section, an improved whale optimization algorithm is designed for the proposed robust ELM training.

3.1 Whale Optimization Algorithm

Whale optimization is an newly algorithm according to the action of whales in hunting their prey [50]. Each whale's position represents a achievable feasible result. In course of the hunting process of a group of whales, every whale has a choice of two behaviors. One of the behavious is to encircle the prey with all the whales moving in towards another whales; the other behavior is the bubble net in which the whales swim in a circular or spiral motion and eject bubbles to drive the prey towards the center. In every generation of swimming, whales select these two behaviors for hunting randomly, and the probability of choosing is 0.5.

3.1.1 Surround Prey

Each whale's position is a selective solution. The whales can use echolocation to identify the location of the prey and surround it.

(1) Whale swimming towards the optimal position

The formula for the updating of a whale position is as follows:

$$x_i^{t+1} = x_{best}^t - A * |C * x_{best}^t - x_i^t|, \tag{11}$$

where t is the iterations, x_{best}^t is the position vector of the optimal whale globally, and x_i^t is the position vector current. The concrete formulas of coefficient vectors $a$and $C$are as follows:

$$A = 2a * r_2 - a, \tag{12}$$

$$C = 2 * r_1, \tag{13}$$

where $r1$ and $r2$ are two numbers which is selected in the range of [−1,1] randomly, and a is the convergence factor, which linearly decreases from two to zero with the number of iterations as follows:

$$a = 2 - 2(\frac{t}{t_{max}}). \tag{14}$$

(2)Whales swimming towards random locations

The formula for updating the position of a whale is as follows:

$$X_i^{t+1} = X_{rand}^t - A * |C * X_{rand}^t - X_i^t|, \tag{15}$$

where x_{rand}^t is the position of the whale randomly selected from the current population.

The shrinking envelope is realized as the convergence factor a decreases, and the fluctuation range of the coefficient vector A also decreases with the convergence factor a. That is, when the convergence factor a decreases from two to zero in the iterative process, the fluctuation of the coefficient vector A also decreases; its range is $[-a, a]$. When the coefficient vector A is a random value in $[-1, 1]$, the position of a whale at time $t + 1$ can be anywhere between the position at time t and the global optimal position at time t.

3.1.2 Bubble Net Chase

In the process of surrounding prey, the search mode of the whale optimization algorithm is to search in the vicinity of the optimal individual or search in the vicinity of a random individual. When hunting, the whales expel air bubbles in a pattern to form a bubble net to constrain their prey. In order to use the bubble nets this way, the whales constantly update their positions. When using a bubble net, the formula for updating a whale's position is as follows:

$$X_i^{t+1} = |X_{best}^t - X_i^t| * e^{pq} * \cos(2\pi q) + X_{best}^t, \tag{16}$$

where p is a constant (the default is one), and q is a random number uniformly distributed in $[-1, 1]$.

According to the above analysis, it can be known that the main parameters of the whale optimization algorithm are the coefficient vectors A and C, and the parameter A is essential in coordinating the global exploration and local development capabilities of the whale optimization algorithm. When $|A| > 1$, the whale population expands the scope of the search to find better candidate solutions; when $|A| < 1$, the whale population narrows the search scope and performs a finer search in a local area. The value of parameter A depends to a large extent on the change of the convergence factor a: a larger convergence factor has a better global search ability and prevents the algorithm from falling into a local optimum; a smaller convergence factor has a stronger local search capability to speed up the convergence speed of the algorithm.

3.2 Cellular Automata

A cellular automaton is a grid dynamic system with a discrete time, space, and state [50]. Unlike general dynamic models, cellular automata are not determined by strictly defined physical equations or functions but are composed of a series of model construction rules. Any model that meets these rules can be counted as a cellular automaton model. Therefore, cellular automata is a general term for a class of models, or a method framework.

The basic unit of cellular automata is individual cells scattered in a regular grid, and each of them take a finite discrete state, follow the same rules of action, and perform synchronous updates according to certain evolutionary rules that apply to all the cells at a certain moment. The state of the whole is the configuration of the cellular automaton. The basic elements are composed of cells, cell states, cell spaces, neighbours, and rules, it can be represented by a four-element array, as follows:

$$A = (Z_n, S, N, f), \tag{17}$$

where A is a cellular automaton; Z_n is the cell space, and the subscript n is the dimension of the cell space; S is the finite state set of the cell; $S = \{s_1, s_2, \ldots, s_i, \ldots, s_k\}$ and s_i represents the ith state of the cellular automaton; N is the neighbor of the central cell, which is a space vector containing n different cells, expressed as $N = \{c_1, c_2, \cdots, c_i, \cdots, c_k\}, c_i$ is the position of neighbouring cell relative to the central cell, $c_i \in$ integer set, n is the number of neighboring cells; f is the evolution rule of $S_i^t \rightarrow S_i^{t+1}$, which represents the rule followed

by the evolution of the cell state from time t to time $t+1$.

3.3 Improved Whale Optimization Algorithm

In order to further improve the local search effect of the whale optimization algorithm, the cellular automaton strategy is incorporated in the WOA, and the novel improved whale optimization algorithm (referred to as CA-WOA) is proposed in our work. All the information in the CA-WOA are inherited within the unit. A whale (i.e., a solution) is defined as a cell, and all the cells are referred to as a whale swarm in the CA-WOA. During the initialization phase, whale populations are randomly created. Subsequently, each whale is randomly assigned to a grid structure. Note that the number of each whale in the index lattice structure is required in the search process. We can define the neighborhood according to the dimension structure. The interaction between whales is limited to their neighbors; each unit must make internal use of their neighbors. At the same time, overlapping neighbors provide a migration mechanism from one community to another. It can spread the information of any cell throughout the whole whale population to help in the exploration of the search space.

The steps in the improved algorithm are detailed as follows:

1) We first bring the position of all whales into the objective function and select the best position of all whales;
2) We iterate over all the whale positions. We bring the positions of all whales into Equations 11 to 16 in turn to overlap the positions of a new set of whales.
3) We calculate the differences and d_{ij} values of all the dimensions of each whale after the iteration and the whale at the best position of the previous iteration by converting the rules.
4) We compare the d_{ij} values of all the whales in turn. We get the k whales with the nearest differences;
5) The positions of the k whales and the best whales in the previous iteration are brought into the objective function to select the best whales in this iteration.
6) If the experiment reaches the iterative end condition, we perform Step 7; otherwise, return to Step 2.
7) Output the best whale position.

In the process of whales searching for prey, each whale searches within a certain range and communicates with other whales within the search range to realize information sharing in order to find the optimal position faster and improve the algorithm convergence speed. Therefore, designing an effective search radius is the key to determining the whale search range. We select the next round of optimal solutions from the neighbors of a certain round of optimal solutions, and the distance formula for the neighbors is as follows:

$$d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik}^{t} - x_{jk}^{t})^2}, \tag{18}$$

where x_{ik}^{t} represents the k latitude of the ith whale, x_{jk}^{t} represents the k latitude of the jth whale, and d_{ij} represents the distance between the ith whale and the jth whale. There are n latitudes.

We select the k closest whales, calculate their fitness, and select X_{best}^{t*}. We then bring X_{best}^{t*} into the update formula to update the position as follows:

$$x_i^{t+1} = x_{best}^{t*} - A * |C * x_{best}^{t*} - x_i^{t}|, \tag{19}$$

$$x_i^{t+1} = |x_{best}^{t} - x_i^{t}| * e^{pq} * \cos 2\pi q + x_{best}^{t*}. \tag{20}$$

Here, x_i^{t+1} represents the position of the ith whale in the $t+1$ iteration, x_{best}^{t*} represents the position vector of the best whale in the neighborhood, p is a constant (the default is one), q is a random number uniformly distributed in $[-1, 1]$, and A and C are coefficient vectors. The structure chart of the improved WOA is shown in Fig. 1. The pseudocode is shown in Algorithm 1.

4 Proposed Hybrid Forecasting Model

In this section, we propose a hybrid load forecasting model. This model is a combination of the robust ELM and the improved WOA. The flow chart is shown in Fig. 2.

In order to optimize the effect of the improved algorithm, we need to obtain the optimal solution of the tuning parameter τ for the Huber loss by cross-validation in advance. Finally, the procedure of our hybrid load-forecasting model can be given as follows:

1) Receive the raw data. The data should be divided into training and test sets.
2) The training set is then further divided into five subsets. Each non-repeating part of the training set (i.e., the subsets) is taken as a test set; the other four parts of the training sets (i.e., the remaining subsets) are used to make a training model. The MSE_i is calculated for this test set; this is repeated five times using a different subset as the test set each time.
3) First, assign the experience value and then apply the five-fold cross validation to average the MSE_i for the five values to obtain the final average MSE,

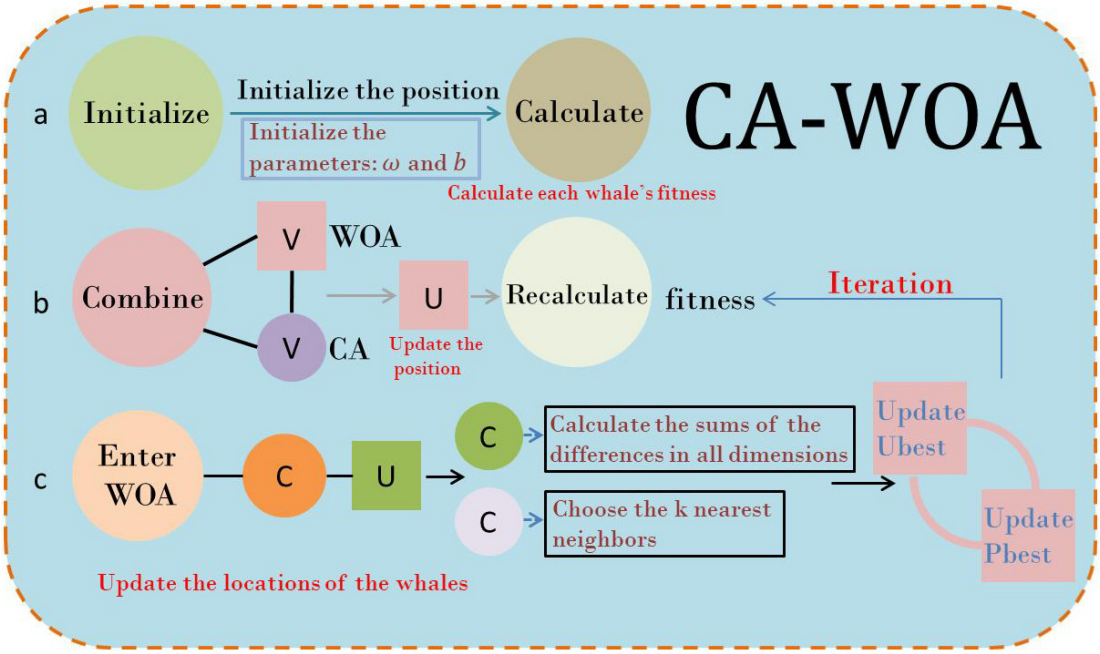

Fig. 1 Structure chart of the Improved Whale Optimization Algorithm

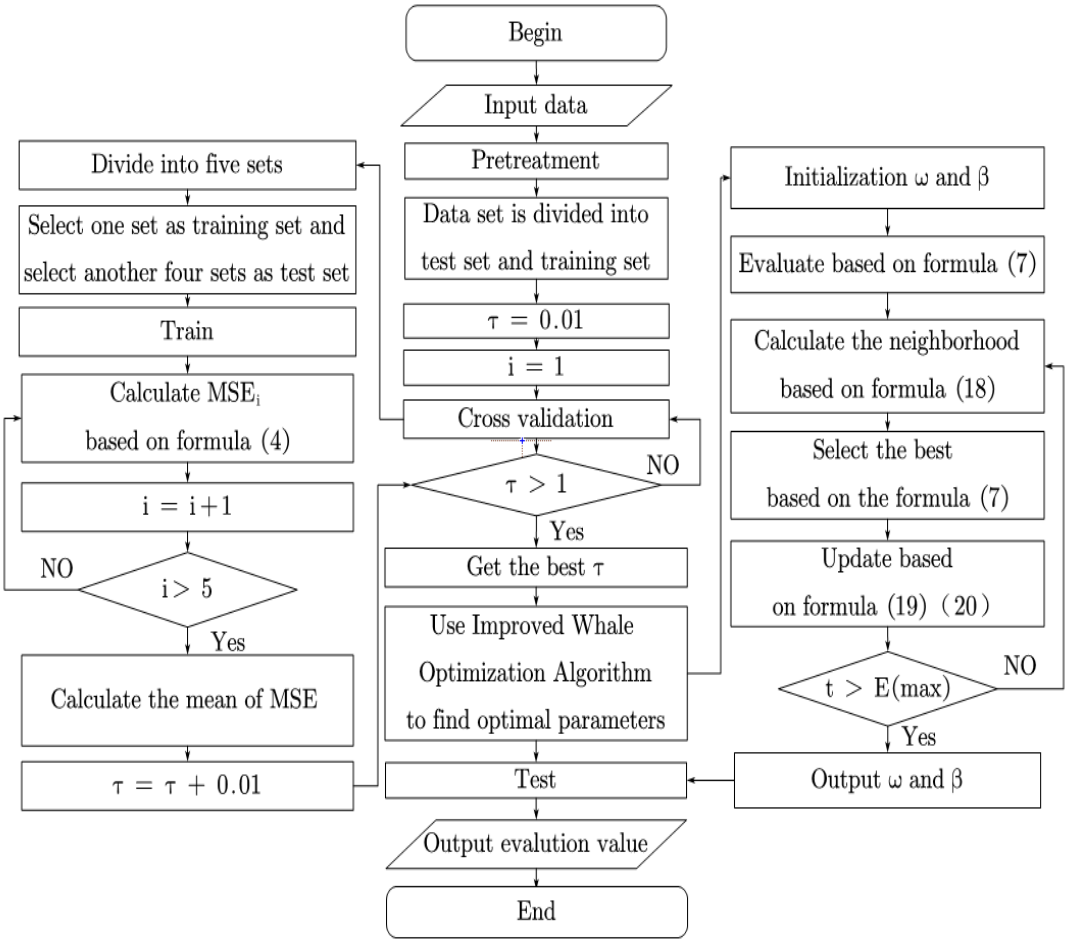

Fig. 2 The flow chart of the hybrid load forecasting model

Algorithm 1 Improved Whale Optimization Algorithm

Require: Parameters (a)

function MergerSortEvaluate
whales population $X_i \leftarrow$ initialization ($whales\_size$)
evaluate
while ($evaluation\_number <$ $max\_evaluation\_number$) **do**
  for $i \leftarrow 1$ to $whales\_size$ **do**
    neighbors $\leftarrow$ calculate Neighborhood ($whales(i)$)
    select the best k whales(neighbors);
    new position of the whales update (the current whales)
    update (a, A and C)
    evaluate (new position of the whales)
    if new position of the whales is better than that of the current whale **then**
      replacement
    end if
    $evaluation\_number^{++}$
  end for
end while
$X_i \leftarrow$ select the best whales
return X_i

called CV as CV $= (\sum_{i=1}^{k} MSE_i)/k$. The five MSEs are compared, and the minimum value of the MSEs is selected and substituted into the Huber loss regression function.

4) In the training set, with the average Huber loss (MHL) minimum as the goal, the CA-WOA is used to solve the optimal parameter w_i, b_j of the ELM, and substitute it into the formula 3 to obtain β_{jk};
5) Bring the input weight, node threshold of the hidden layer, and output weight of the training set into the ELM model and then substitute the input of the test set into the model to obtain the predicted output of the test set.

5 Verification of CA-WOA

In this section, to demonstrate the effectiveness of our optimizer, we conducted research on the two algorithms, traditional WOA and CA-WOA. We selected benchmark functions for the test, and the expression of the functions are provide in Table 1.

In our experiment, the maximum number of iterations and number of search agents of both optimization algorithms were set to 500 and 15, respectively. In addition, we defined three indexes to show the performance of our proposed optimization algorithm: BEST (the optimal fitness value), AVG (the average fitness value), and STD (the variance of the fitness value). We perform all the experiments to obtain the average validity and recorded the calculation results in Table 2.

In Table 2, the CA-WOA had a better optimization effect than the traditional WOA. The BEST, AVG, and STD from our proposed CA-WOA in general were less than the traditional WOA. For example, in f_1, compared with the traditional WOA's AVG, the CA-WOA's AVG saw a decrease of 9.34%; compared with the traditional WOA's STD, the CA-WOA's STD saw a decrease of 6.06%. Furthermore, Fig. 3 shows the average convergence curve of 30 repeated experiments for the WOA and CA-WOA. This shows that our proposed CA-WOA had a good convergence rate and the best optimal solution. For example, in the convergence curve of function f_2, the CA-WOA converged after the eighth iteration, while the WOA converged after 20 iterations. In the function f_1, the convergence accuracy of the CA-WOA was 10^{-4}, and the convergence accuracy of the WOA was only about 10^{-2}.

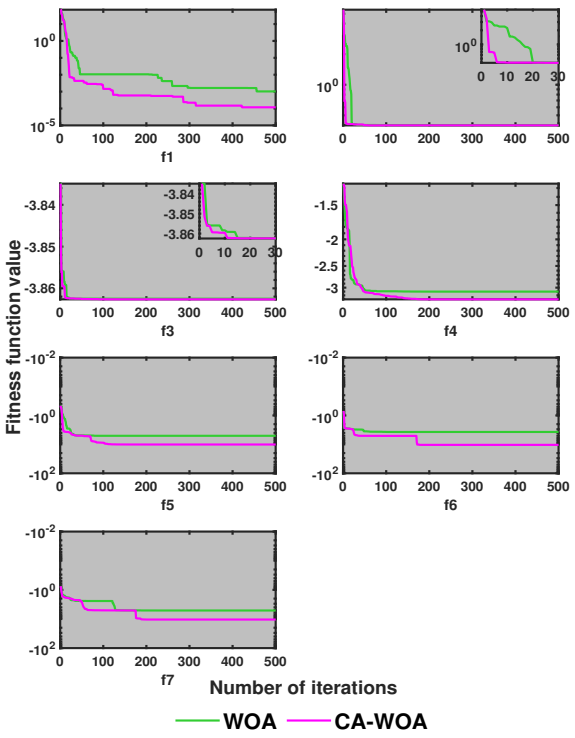

Fig. 3 Convergence curves of two algorithms applied to the benchmark functions

Table 1 Benchmark test functions*

Function	Dimension	Range
$f_1(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-1.28, 1.28]
$f_2(x) = (x^2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6) + 10(1 - \frac{1}{8\pi})\cos x_i + 10$	2	[-5, 5]
$f_3(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1, 3]
$f_4(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]
$f_5(x) = -\sum_{i=1}^{5} [(X - a^i)(X - a^i)^2 + c_i]^{-1}$	4	[0, 10]
$f_6(x) = -\sum_{i=1}^{7} [(X - a^i)(X - a^i)^2 + c_i]^{-1}$	4	[0, 10]
$f_7(x) = -\sum_{i=1}^{10} [(X - a^i)(X - a^i)^2 + c_i]^{-1}$	4	[0, 10]

* $a = [3\ 10\ 30; 0.1\ 10\ 35; 3\ 10\ 30; 0.1\ 10\ 35]$
$p = [0.3689\ 0.117\ 0.2673; 0.4699\ 0.4387\ 0.747; 0.1091\ 0.8732\ 0.5547; 0.03815\ 0.5743\ 0.8828]$,
$c = [1\ 1.2\ 3\ 3.2]$.

Table 2 The experiment results of benchmark test functions

	WOA			CA-WOA		
function	BEST	AVG	STD	BEST	AVG	STD
f_1	0	0.75	5.78	0	0.68	5.43
f_2	0.4	0.46	0.39	0.4	0.42	0.27
f_3	-3.86	-3.86	0	-3.86	-3.86	0
f_4	-3.1	-3.04	0.24	-3.31	-3.20	0.32
f_5	-5.05	-4.87	0.72	-10.11	-9.17	2.06
f_6	-3.72	-3.64	0.26	-10.40	-8.46	2.73
f_7	-5.11	-4.39	1.24	-10.46	-8.22	3.13

In short, the CA-WOA converged faster than than the traditional WOA, and it demonstrated that it can effectively solve actual optimization problems.

6 Case Studies

In the section, we attempt to answer the following questions in order to analyse the experimental results comprehensively:

(1) How do different regression loss functions affect the load forecasting? Does our Huber loss function provide higher prediction accuracy?
(2) How do different optimization algorithms affect the load forecasting? Does the proposed CA-WOA algorithm perform better?
(3) How does our CA-WOA-HUBER-ELM perform in comparison with other popular prediction models?

A set of benchmark models for the exploration of our questions are presented in Table 3. We also added four new models published in recent literature, namely DWT-PSO-RBFNN [32], EEMD-GOA-ELM [31], PSR-SVM [30] and EEMD-MLR-LSTM [33], to further demonstrate the performance of our proposed model CA-WOA-HUBER-ELM. All the programs were run in the MATLAB2019a environment.

Note that the L1- and L2-ELM are presented to show the benefit from the Huber loss in our robust ELM. As for the second question, the improved whale optimization algorithm was calculated and compared with the current mainstream GA, WOA, and other optimization algorithms. Finally, to verify the performance of the proposed CA-WOA-HUBER-ELM model, three popular models, back propagation neural network (BPNN), SVM and long short-term memory (LSTM), are investigated.

Table 3 The all investigated models

	Model	Definition
Basic benchmark Models	BPNN [56]	Back Propagation Neural Network
	SVM [57]	Support Vector Machine
	LSTM [1]	Long Short-Term Memory
	GA-ELM	ELM optimized by GA
	GA-L1-ELM	L1-ELM optimized by GA
	GA-L2-ELM	L2-ELM optimized by GA
	GA-HUBER-ELM	Huber-ELM optimized by GA
	WOA-ELM	ELM optimized by WOA
	WOA-L1-ELM	L1-ELM optimized by WOA
	WOA-L2-ELM	L2-ELMoptimized by WOA
	WOA-HUBER-ELM	Huber-ELM optimized by WOA
	CA-WOA-ELM	ELM optimized by CA-WOA
	CA-WOA-L1-ELM	L1-ELM optimized by CA-WOA
	CA-WOA-L2-ELM	L2-ELM optimized by CA-WOA
New benchmark Models	DWT-PSO-RBFNN [32]	DWT-RBFNN optimized by PSO
	EEMD-GOA-ELM [31]	EEMD-ELM optimized by GOA
	PSR-SVM [30]	SVM optimized by PSR
	EEMD-MLR-LSTM [33]	EMD-LSTM optimized by MLR
Proposed Models	CA-WOA-Huber-ELM	Huber-ELM optimized by CA-WOA

We conduct electric-forecasting analysis on electric data of Nanjing and New South Wales in Table 4. We recorded the electric load data of Nanjing New South Wales every half hour. The Nanjing data set had 1,920 data points(from 00:00 on February 18, 2003 to 23:30 on March 29, 2003). The training set consisted of 1152 data points, and the test set consisted of 768 data points. The dataset for New South Wales included 2256 sample points(from 00:00 on January 1, 2018 to 23:30 on February 16, 2018); the data set was divided into two subsets: the training set consisted of 1134 data points. The test set included 912 data points. The electricity data is shown in Figs. 4 and 5 (the red line divides the training and test sets).

The target of the research was to use historical power load data (the latest 48 load observations) to predict the 1-step-ahead electrical load.

We analysed the experimental results for the two datasets from three aspects, the performance and prediction accuracy of the different optimization algorithms, the different regression loss functions, and the prediction accuracy of the various currently popular prediction models, as described in the following subsections. The prediction results obtained were given by the three mainstream evaluation functions of mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE). Here, it should be mentioned that we empirically normalized the original data and set the search space for the weights and biases of the ELM models optimized by the WOA, GOA and CA from -1 to 1 in our experiment. All the experimental settings can be obtained in our source code at: https://github.com/piobotHome/ELM

6.1 Experiment Analysis for Nanjing

The experimental results for Nanjing are reported in Table 5. The actual power and prediction result graph based on CA-WOA-HUBER-ELM in Nanjing is shown in Fig. 6. The experimental analysis is detailed as follows.

6.1.1 Comparisons of Robustness among the Huber-ELM, L1-ELM, L2-ELM and Basic ELM

To identify the differences among the $L1$-ELM, $L2$-ELM, Huber-ELM and basic ELM, we plotted the electric load prediction diagram in Fig. 7.

As can be seen from Fig. 7, compared with the other three figures, the electric load prediction curve of the Huber losses in the improved algorithm, in other words, (a) Huber-ELM, (b) basic ELM, (c) $L1$-ELM and (d) $L2$-ELM, were closer to the actual electric data curve. The error distribution figures show that Huber-ELM not only had the smallest error curve but also had the best error distribution, which indicates that this model

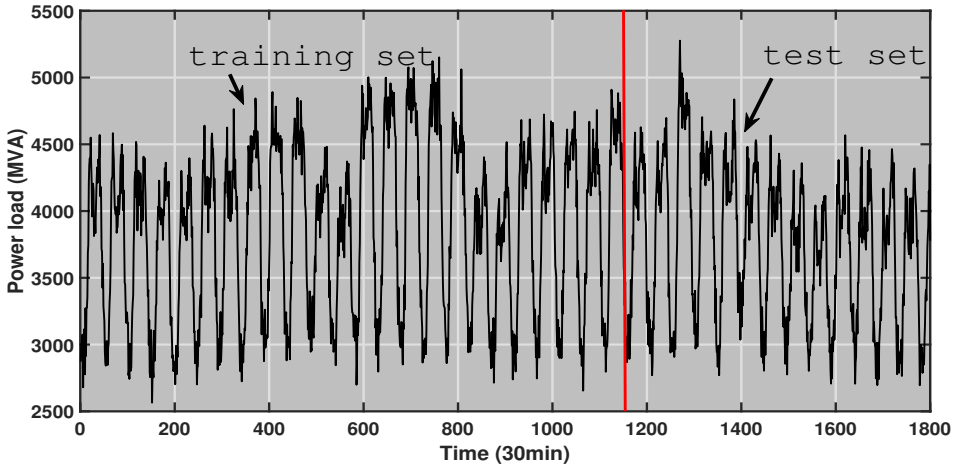

Fig. 4 The electricity data of Nanjing (from 00:00 on February 18, 2003 to 23:30 on Marth 29, 2003)

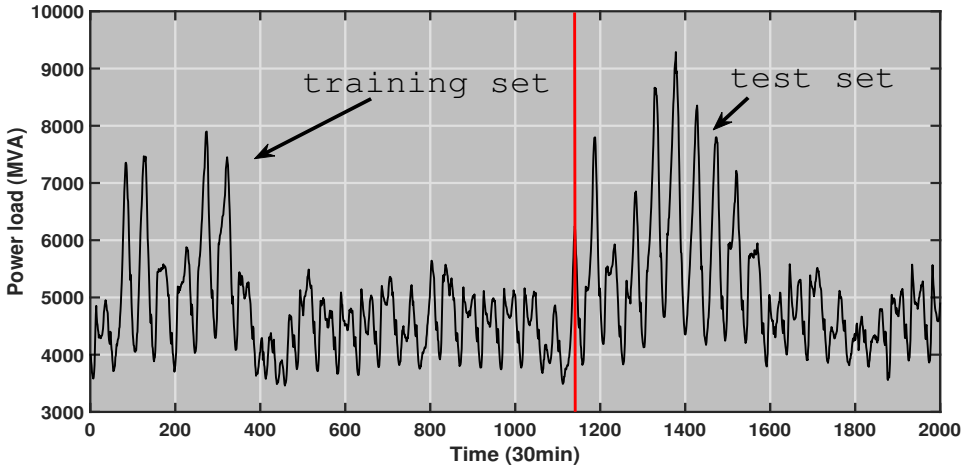

Fig. 5 The electricity data of New South Wales (from 00:00 on January 1, 2018 to 23:30 on February 16, 2018)

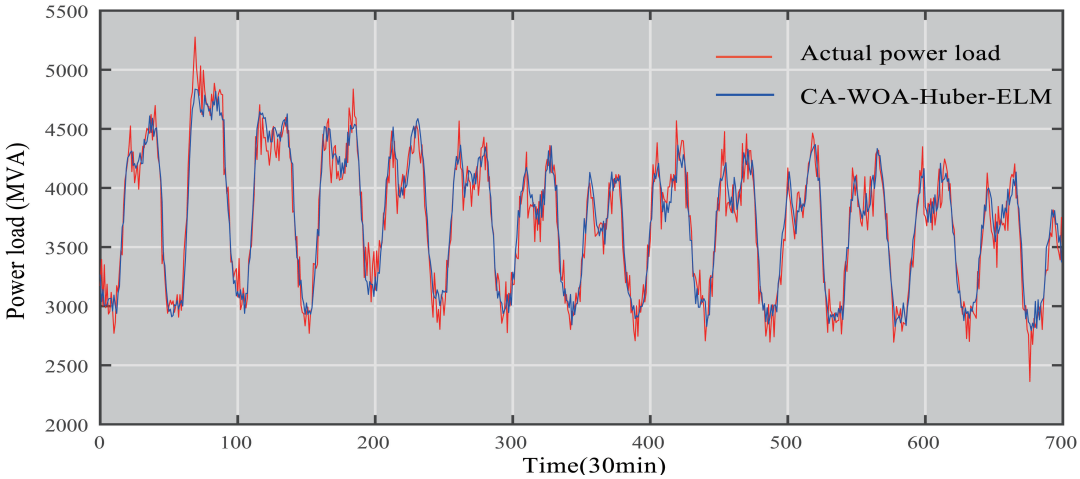

Fig. 6 The actual power and prediction result graph based on CA-WOA-Huber-ELM in Nanjing

Table 4 The data description for in Nanjing and New South Wales

	Size	Max	Min	Mean	Std
Nanjing					
All	1920	5276.50	2362.15	3802.73	607.71
Training	1200	5151.65	2564.23	3858.86	632.78
Test	720	5276.50	2362.15	3709.19	551.32
New South Wales					
All	2256	9289.50	3458.91	4894.25	942.57
Training	1344	8665.64	3458.91	4840.19	921.12
Test	912	9289.50	3458.91	4973.92	968.34

Table 5 Error indexes of the experimental data in Nanjing

	Model	Train			Test		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
Basic benchmark models	BPNN	181.44	159.63	0.03	203.63	162.34	0.04
	SVM	221.63	162.33	0.04	243.72	182.59	0.05
	LSTM	160.23	129.36	0.03	163.81	132.80	0.04
	GA-ELM	204.65	161.31	0.04	216.27	174.09	0.05
	GA-L1-ELM	170.05	133.47	0.04	170.22	136.24	0.04
	GA-L2-ELM	162.40	127.29	0.03	167.65	132.83	0.04
	GA-HUBER-ELM	162.33	127.84	0.03	170.35	134.86	0.04
	WOA-ELM	190.76	150.30	0.04	201.26	156.81	0.04
	WOA-L1-ELM	164.83	129.89	0.04	171.43	136.47	0.04
	WOA-L2-ELM	161.25	127.22	0.03	171.24	137.00	0.04
	WOA-HUBER-ELM	169.00	133.22	0.04	174.62	139.67	0.04
	CA-WOA-ELM	188.87	149.66	0.04	187.40	149.18	0.04
	CA-WOA-L1-ELM	158.30	122.98	0.03	164.97	130.94	0.04
	CA-WOA-L2-ELM	156.32	123.98	0.03	166.22	132.81	0.04
New benchmark models	DWT-PSO-RBFNN	132.13	105.50	0.04	165.34	132.05	0.04
	EEMD-GOA-ELM	199.91	158.12	0.04	202.26	159.12	0.04
	PSR-SVM	139.05	120.53	0.04	205.78	163.06	0.04
	EEMD-MLR-LSTM	225.54	179.65	0.04	271.97	172.32	0.05
Proposed Model	CA-WOA-HUBER-ELM	**158.11**	**123.08**	**0.03**	**163.59**	**129.54**	**0.04**

performed better than the others in the forecasting. From Table 5, compared with the basic ELM's RMSE, the Huber-ELM's RMSE dropped by 12.71%; compared with the basic ELM's MAE, the Huber-ELM's MAE dropped by 13.16%; compared with the basic ELM's MAPE, the Huber-ELM's MAPE dropped by 0%. Compared with the traditional ELM, the $L1$-ELM, $L2$-ELM, and Huber-ELM had smaller MAE, RMSE, and MAPE values. The results show that the CA-WOA-HUBER-ELM algorithm achieved a more compact network structure and better generalization performance, and could refer to a wider range of algorithms. The predicted electric load data were more stable and had better robustness.

6.1.2 Comparisons of the Local Search Capabilities among the Huber-ELM using the GA, WOA, and CA-WOA

To illustrate the local search capabilities among the Huber-ELM using the GA, WOA and CA-WOA, we made the electric load error distribution diagram corresponding to the three situations that included the GA, WOA and CA-WOA, as shown in Fig. 8. The error distribution figuresg show that the CA-WOA-Huber-ELM not only had the smallest error curve but also had the best error distribution, which indicates that this model performed better than the others in the forecasting. From Fig. 8, we can see that (a) CA-WOA's improved algorithm had a more accurate prediction re-

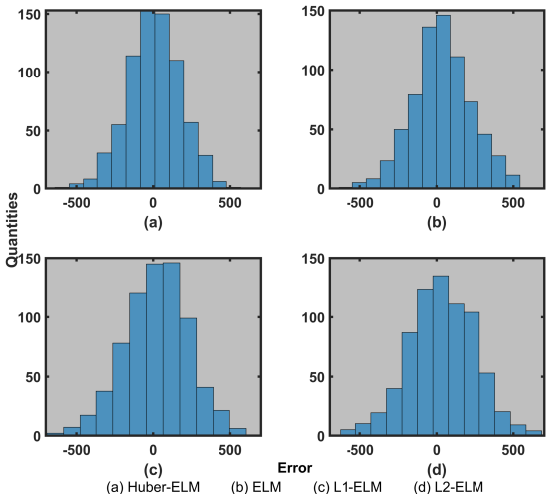

Fig. 7 Comparison of the error distribution by using the improved ELM using different loss functions in Nanjing

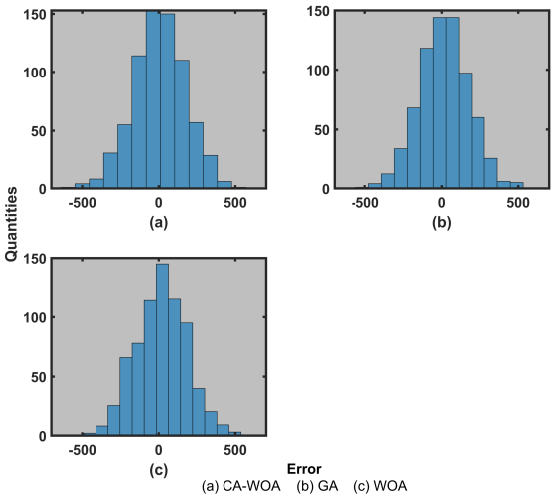

Fig. 8 Comparison of the error distribution by using the improved ELM using different optimization algorithms in Nanjing

sult when compared with the other two figures, (b) GA and (c) WOA. From Table 5, compared with GA's RMSE and the traditional WOA's RMSE, the RMSE of the CA-WOA decreased by 3.97% and 6.31%, respectively; compared with the GA's MAE and the traditional WOA's MAE, the MAE of the CA-WOA decreased by 3.94% and 7.25%, respectively; compared with the GA's MAPE and the traditional WOA's MAPE, the MAPE of CA-WOA decreased by 0% and 0%, respectively. It can be seen that CA-WOA's RMSE, MAE, and MAPE were smaller than the GA's and WOA's RMSE, MAE and MAPE, respectively. The CA-WOA-HUBER-ELM had good convergence precision, and the convergence effect was very good in the local area. The convergence rate was relatively fast. In actual electric load forecasting, electric load can be better predicted.

Additionally, the convergence curves of all the algorithms are plotted in Fig. 9. Through the analysis of three convergence curves, we can see that when we use the Huber-ELM, the CA-WOA's convergence rate is close to that of the GA, which is faster than that of the other algorithms. In the end, the CA-WOA had a better convergence effect than the GA. When improving the ELM with $L1$ loss, at the beginning, the CA-WOA converged faster than the WOA and slower than the GA. However, later, the CA-WOA converged faster than the GA and had the best convergence effect. When improving the ELM with $L2$ loss, the CA-WOA always had the fastest convergence speed and the best effect. Thus, we can see that the CA-WOA-HUBER-ELM algorithm had high speed and convergence accuracy. Its convergence ability was stronger, and the effect was more remarkable.

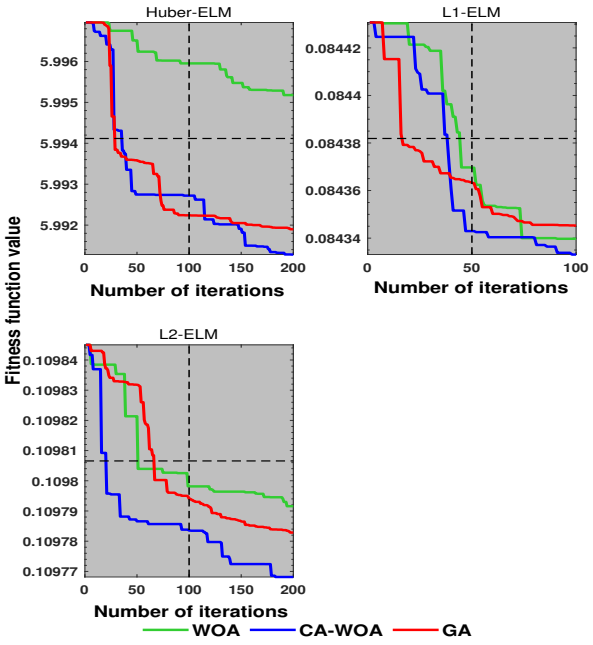

Fig. 9 Convergence curve of different optimization algorithms in predicting Nanjing electric data

6.1.3 Comparisons of the Predictive Capabilities of the BPNN, SVM, LSTM and CA-WOA-HUBER-ELM

To show the predictive capabilities of the BPNN, SVM, LSTM, and CA-WOA-HUBER-ELM, the corresponding electric load prediction diagram is shown in Fig. 10. From Fig. 10, we can see that the errors between the prediction curves of the (a) CA-WOA-HUBER-ELM, (b) BPNN, (c) SVM, and (d) LSTM and the actual electric load curve were relatively large.

In comparison, the forecast results were closer to the actual electric load. From Table 5, compared with the BPNN's, SVM's, and LSTM's RMSE, the RMSE of the CA-WOA-HUBER-ELM decreased by 19.66% , 32.88% and 0.13%, respectively; compared with the BPNN's, SVM's, and LSTM's MAE, the MAE of the CA-WOA-HUBER-ELM decreased by 20.20% , 29.05%, and 2.45%, respectively; compared with the BPNN's, SVM's, and LSTM's MAPE, the MAPE of the CA-WOA-HUBER-ELM decreased by 0%, 20% and 0%, respectively. It can be seen that the CA-WOA-HUBER-ELM's RMSE, MAE, and MAPE were smaller than the BPNN's, SVM's, and LSTM 's RMSE, MAE, and MAPE. respectively. In short, the CA-WOA-HUBER-ELM had higher accuracy.

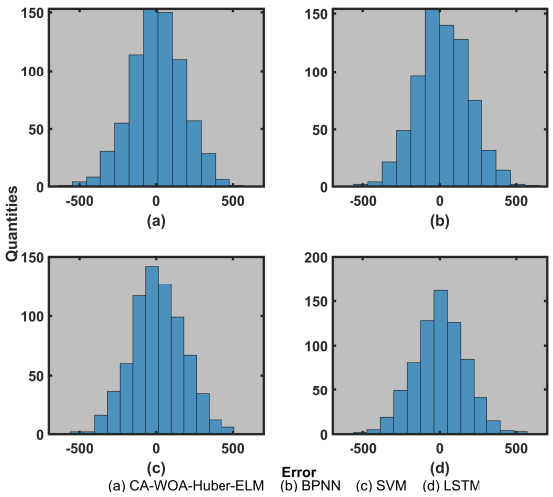

Fig. 10 Comparison of the error distribution by using several popular prediction models in Nanjing

6.1.4 Comparisons of the Predictive Capabilities of the DWT-PSO-RBFNN, EEMD-GOA-ELM, PSR-SVM, EEMD-MLR-LSTM and CA-WOA-HUBER-ELM

In this section, we further compare four different prediction models with our proposed CA-WOA-HUBER-ELM model. In comparison, the CA-WOA-HUBER-ELM forecast results were closer to the actual electric load. Compared with the DWT-PSO-RBFNN's, EEMD-GOA-ELM's, PSR-SVM's, and EEMD-MLR-LSTM's RMSE, the RMSE of the CA-WOA-HUBER-ELM decreased by 1.06% , 19.11%, 20.5%, and 39.85%, respectively; compared with the DWT-PSO-RBFNN's, EEMD-GOA-ELM's, PSR-SVM's, and EEMD-MLR-LSTM's MAE, the MAE of CA-WOA-HUBER-ELM decreased by 1.9%, 18.59%, 20.56%, and 24.83%, respectively; compared with the DWT-PSO-RBFNN's, EEMD-GOA-ELM's, PSR-SVM's, and EEMD-MLR-LSTM's MAPE, the MAPE of the CA-WOA-Huber-ELM decreased by 0%, 0%, 0%, and 20%, respectively. It can be seen that the CA-WOA-HUBER-ELM's RMSE, MAE, and MAPE were smaller than the DWT-PSO-RBFNN's, EEMD-GOA-ELM's, PSR-SVM's, and EEMD-MLR-LSTM's RMSE, MAE, and MAPE, respectively. In brief, the CA-WOA-HUBER-ELM was more accurate.

6.1.5 Wilcoxon Signed-rank Test with RMSE Criterion in the New South Wales Test Set

In this subsection, we present the results of the experiments on the proposed model and other models to test the differences among them. We selected the RMSE values of the Nanjing test set as a sample for conducting a Wilcoxon signed-rank test.

As shown in Table 6, the result $h = 1$ indicates a rejection of the null hypothesis, and $h = 0$ indicates a failure to reject the null hypothesis at a 5% significance level. According to the results, the proposed model was significantly different from the other models. Moreover, according to Table 5, the value of the RMSE of the proposed model was somewhat smaller than the others.

In brief, the CA-WOA-HUBER-ELM algorithm proposed by us was used to predict electric load data with good robustness, fast convergence speed, and high accuracy with the Nanjing load-forecasting data.

6.2 Experiment Analysis for New South Wales

This subsection describes an additional evlauation; the performance of the proposed CA-WOA-HUBER-ELM algorithm was also evaluated using the New South Wales electric load data, as recorded in Table 7. The actual power and the prediction result graph based on the CA-WOA-HUBER-ELM for the New South Wales data is shown in Fig. 11.

The experimental details are as follows.

6.2.1 Comparison of Robustness among the Huber-ELM, L1-ELM, L2-ELM and Basic ELM

To show the different performances among the $L1$-ELM, $L2$-ELM, Huber-ELM, and basic ELM, the corresponding electric load prediction diagram is shown in Fig. 12. The error distribution figures show that the Huber-ELM not only had the smallest error curve but also had the best error distribution, which indicates that this model performed better than the others in the forecasting.

From Fig. 12, we can see that the electric load prediction curve of the Huber loss improved algorithm,

Table 6 The Wilcoxon signed-rank test results with RMSE criterion in Nanjing

Proposed Model	CA-WOA-Huber-ELM	h	p-value
Basic benchmark models	BPNN	1	0.00
	SVM	1	0.00
	LSTM	1	0.00
	GA-ELM	1	0.00
	GA-L1-ELM	1	0.00
	GA-L2-ELM	1	0.00
	GA-HUBER-ELM	1	0.00
	WOA-ELM	1	0.00
	WOA-L1-ELM	1	0.00
	WOA-L2-ELM	1	0.00
	WOA-HUBER-ELM	1	0.03
	CA-WOA-ELM	1	0.00
	CA-WOA-L1-ELM	1	0.00
	CA-WOA-L2-ELM	1	0.00
New benchmark models	DWT-PSO-RBFNN	1	0.00
	EEMD-GOA-ELM	1	0.00
	PSR-SVM	1	0.00
	EEMD-MLR-LSTM	1	0.00

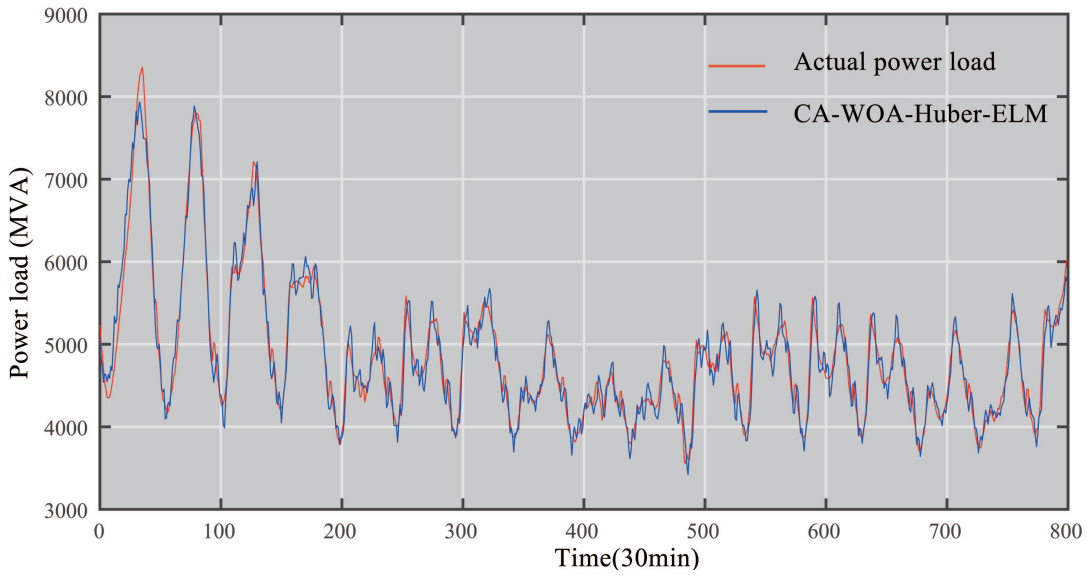

Fig. 11 The actual power and the prediction result graph based on CA-WOA-Huber-ELM in New South Wales

(a) Huber-ELM, was closer to the actual electric data curve than the other three graphs, (b) basic ELM, (c) $L1$-ELM, and (d) $L2$-ELM. From Table 7, compared to the basic ELM's RMSE, the Huber-ELM's RMSE decreased by 43.20%; compared to the basic ELM's MAE, the Huber-ELM's MAE decreased by 43.03%; compared to the basic ELM's MAPE, the Huber-ELM's MAPE decreased by 33.34%. It can be seen that the Huber-ELM had a smaller MAE, RMSE, and MAPE values than the traditional ELM, $L1$-ELM, and $L2$-ELM. The results show that the CA-WOA-HUBER-ELM algorithm achieved a more compact network structure and better generalization performance and could use a wider range of algorithms. The predicted electric load data was more stable and had better robustness.

Table 7 Error indexes of the experimental data in New South Wales

	Model	Train			Test		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
Basic benchmark models	BPNN	226.86	186.69	0.03	243.77	190.52	0.04
	SVM	399.84	205.63	0.04	417.82	210.50	0.04
	LSTM	130.66	97.65	0.02	132.94	98.03	0.02
	GA-ELM	204.03	161.49	0.03	234.00	181.16	0.04
	GA-L1-ELM	118.91	93.63	0.02	182.63	131.48	0.03
	GA-L2-ELM	131.63	106.05	0.02	183.24	138.81	0.03
	GA-HUBER-ELM	130.08	101.43	0.02	172.98	127.24	0.03
	WOA-ELM	235.23	184.60	0.04	319.39	241.39	0.05
	WOA-L1-ELM	132.16	104.74	0.02	156.09	121.59	0.02
	WOA-L2-ELM	113.42	89.99	0.02	164.46	118.97	0.02
	WOA-HUBER-ELM	120.32	94.30	0.02	168.65	118.06	0.02
	CA-WOA-ELM	214.46	152.01	0.03	197.26	155.77	0.03
	CA-WOA-L1-ELM	122.82	97.52	0.02	122.50	96.11	0.02
	CA-WOA-L2-ELM	188.02	138.44	0.03	134.00	106.32	0.02
New benchmark models	DWT-PSO-RBFNN	118.15	93.18	0.04	122.65	95.80	0.04
	EEMD-GOA-ELM	181.92	229.94	0.04	209.69	268.59	0.04
	PSR-SVM	191.33	160.45	0.04	253.56	170.21	0.04
	EEMD-MLR-LSTM	285.55	199.66	0.05	305.55	203.47	0.06
Proposed model	CA-WOA-HUBER-ELM	**108.05**	**83.26**	**0.02**	**112.03**	**88.73**	**0.02**

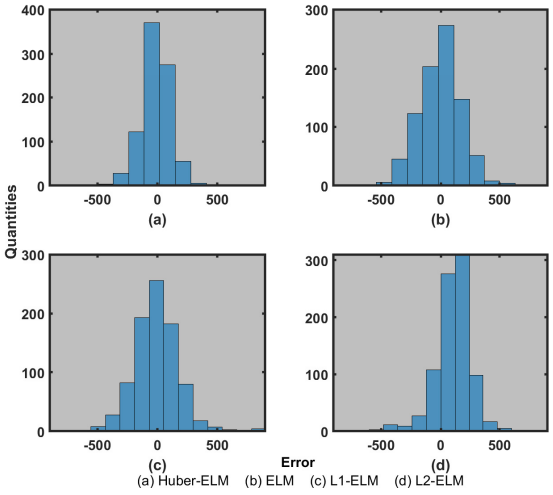

Fig. 12 Comparison of the error distribution by using the improved ELM using different loss functions in New South Wales

6.2.2 Comparison of the Local Search Capabilities among the Huber-ELM Using the GA, WOA and CA-WOA

To display the performance of the Huber-ELM training using the GA, WOA, and CA-WOA, and the corresponding predictions are plotted in Table 13.

From Fig. 13, we can see that compared to the other two, (a) the CA-WOA and (b) the GA, the prediction of the improved algorithm with (c) the WOA was the least close to the actual data. The error distribution figures show that the CA-WOA-Huber-ELM not only had the smallest error curve but also had the best error distribution, which indicates that this model performed better than the others in the forecasting. From Table 12, compared with the GA's and traditional WOA's RMSE, the CA-WOA's RMSE decreased by 35.23% and 33.5%, respectively; compared with the GA's and traditional WOA's MAE, the CA-WOA's MAE decreased by 30.26% and 24.84%, respectively; compared with the GA's and traditional WOA's MAPE, the CA-WOA's MAPE decreased by 33.34% and 0%, respectively. We found that the CA-WOA in predicting on the test set had lower values for the RMSE, MAE, MAPE than the GA and WOA. Moreover, the CA-WOA-HUBER-ELM had good convergence, and the convergence performance was very good in local areas. The convergence speed was relatively fast. In actual electric load forecasting, the electric load can be better predicted. In addition, the training curve for all the investigated optimization algorithms are recorded in Fig. 14. Through the analysis, we can see that when we used Huber loss to improve the ELM, the CA-WOA was close to the GA convergence rate and faster than the other

algorithms. In the end, the CA-WOA convergence effect was better than the GA. When using the $L1$ loss to improve the ELM, initially, the CA-WOA convergence rate was as fast as WOA but slower than GA. However, the CA-WOA convergence rate exceeded that of the GA. The convergence effect is the best. When using $L2$ loss to improve the ELM, the CA-WOA convergence rate was the fastest; it worked the best. Therefore, we can see that the CA-WOA-Huber-ELM algorithm had the great convergence and better results.

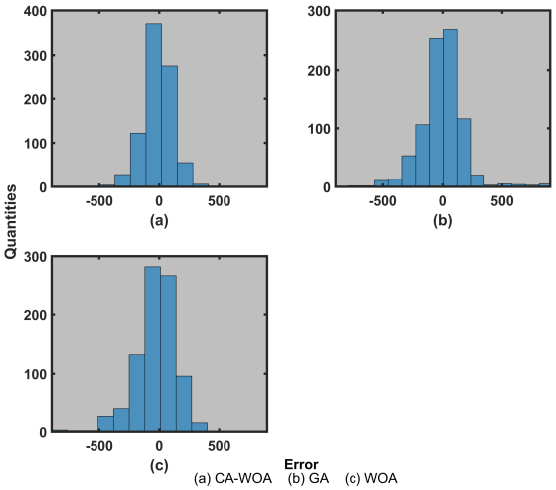

Fig. 13 Comparison of the error distribution by using the improved ELM using different optimization algorithms in New South Wales

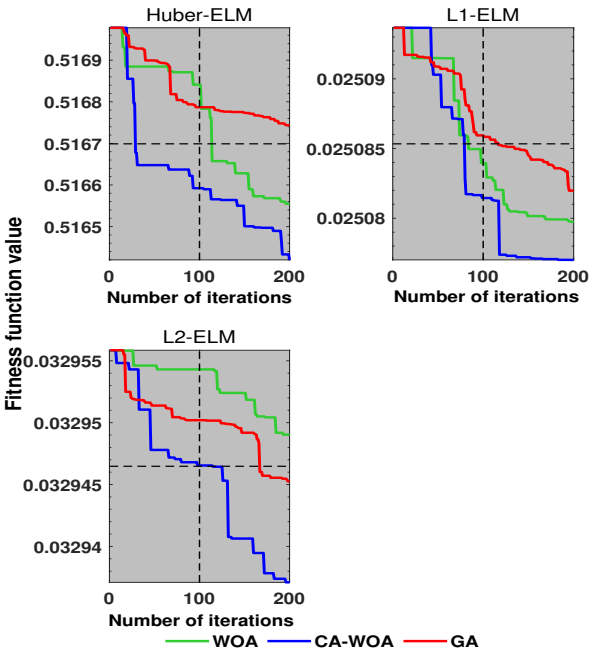

Fig. 14 Convergence curve of different optimization algorithms in predicting New South Wales electric data

6.2.3 Comparison of the Predictive Capabilities among the BPNN, SVM, LSTM and CA-WOA-HUBER-ELM

To explore the predictive capabilities of the BPNN [56], SVM, LSTM [53], and CA-WOA-HUBER-ELM, we drew the electric load prediction diagram in Fig. 15. We can see from Fig. 15 that the errors between the prediction curves of (a) BPNN, (b) SVM, and (c) LSTM and the actual electric load curve were relatively large. By comparison, the results of the (d) CA-WOA-HUBER-ELM prediction were closer to the actual electric load. From Fig. 12, compared with the BPNN's, SVM's, and LSTM's RMSE, the RMSE of the CA-WOA-HUBER-ELM decreased by 54.04%, 72.19%, and 15.73%, respectively; compared with the BPNN's, SVM's, and LSTM's MAE, the MAE of the CA-WOA-HUBER-ELM decreased by 53.43% , 57.85% and 9.49%, respectly; compared with the BPNN's, SVM's, and LSTM's MAPE, the MAPE of the CA-WOA-HUBER-ELM decreased by 50% , 50% and 0%, respectively. It can be seen that the CA-WOA-HUBER-ELM's RMSE, MAE, and MAPE were smaller than the BPNN's, SVM's, and LSTM 's RMSE, MAE and MAPE, respectively. In short, the CA-WOA-HUBER-ELM had higher convergence accuracy.

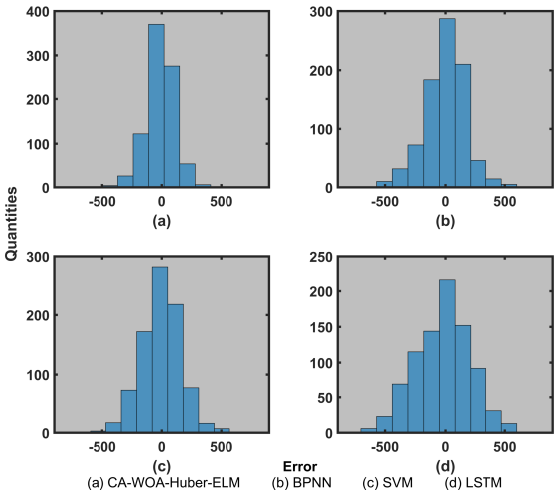

Fig. 15 Comparison of the error distribution by using several popular prediction models in New South Wales

6.2.4 Comparisons of the Predictive Capabilities of the DWT-PSO-RBFNN, EEMD-GOA-ELM, PSR-SVM, EEMD-MLR-LSTM and CA-WOA-HUBER-ELM

In this sub-section, we present the further comparison of four different prediction models with our proposed CA-WOA-HUBER-ELM model.

By comparison, the results of CA-WOA-HUBER-ELM prediction are closer to the actual electric load. compared with DWT-PSO-RBFNN's RMSE, EEMD-GOA-ELM's RMSE, PSR-SVM's RMSE and EEMD-MLR-LSTM's RMSE, the RMSE of CA-WOA-HUBER-ELM decreases by 8.65% ,by 46.57% ,55.81% and 63.33%; compared with DWT-PSO-RBFNN's MAE, EEMD-GOA-ELM's MAE, PSR-SVM's MAE and EEMD-MLR-LSTM's MAE, the MAE of CA-WOA-HUBER-ELM decreases by 7.37% ,by 66.96% ,47.87% and 56.39%; compared with DWT-PSO-RBFNN's MAPE, EEMD-GOA-ELM's MAPE, PSR-SVM's MAPE and EEMD-MLR-LSTM's MAPE, the MAPE of CA-WOA-HUBER-ELM decreases by 50% , 50%, 50% and 66.67%. It can be seen that the CA-WOA-HUBER-ELM's RMSE, MAE, and MAPE were smaller than the BPNN's, SVM's, and LSTM 's RMSE, MAE, and MAPE, respectively. In short, the CA-WOA-HUBER-ELM's convergence accuracy was higher.

6.2.5 Wilcoxon Signed-rank Test with RMSE Criterion in the New South Wales Test Set

In this sub-section, we present the results of the experiments on the proposed model and other models to test the differences among them. We selected the RMSE values from the New South Wales test set as a sample for conducting a Wilcoxon signed-rank test.

As shown in Tab. 8, the result $h = 1$ indicates a rejection of the null hypothesis, and $h = 0$ indicates a failure to reject the null hypothesis at a 5% significance level. According to the results, the proposed model was significantly different from the other models. Moreover, according to Table 7, the value of the RMSE of the proposed model was somewhat smaller than the others.

All in all, using the New South Wales data, our proposed CA-WOA-HUBER-ELM algorithm was used to predict the electric load data with good robustness, fast convergence, and high accuracy.

7 Conclusion

Accurate and effective electric load forecasting is an essential part of ensuring the safe operation of power grids. However, the complexity of the power grid poses many future challenges to electric load forecasting, and the current popular forecasting methods cannot address all these challenges. To address this, this paper proposed a new hybrid load-forecasting model, CA-WOA-HUBER-ELM, which is the combination of a robust extreme learning machine and an improved whale optimization algorithm. Furthermore, the Huber loss, which is insensitive to outliers, is treated as the objective function for our robust ELM training. Also, an improved whale optimization algorithm (CA-WOA) is designed by introducing a cellular automaton for the local search to improve the convergence of the optimizer. Our proposed improved CA-WOA achieved great performance in seven benchmark test functions, which showed great improvement for exploitation. Finally, according to two real electric load-forecasting datasets for Nanjing and New South Wales and comparative experiments with four newly added algorithms, our CA-WOA-HUBER-ELM model demonstrated great advantages in dealing with outliers and improving forecasting accuracy.

Declaration of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Yang Yang: Project administration, Supervision, Investigation **Zhenghang Tao:** Software, Visualization, Writing - original draft. **Chen Qian:** Formal analysis, Writing - original draft **Yuchao Gao:** Visualization, Writing - original draft. **Hu Zhou:** Writing – review & editing. **Zhe Ding:** Writing- review & editing. **Jinran Wu:** Supervision, Project administration,Investigation, Writing – review & editing.

Acknowledgements

This work was supported by Natural Science Foundation of China (61873130), by Natural Science Foundation of Jiangsu Province (BK20191377), by 1311 Talent Project of Nanjing University of Posts and Telecommunications, and by Postgraduate Research & Practice Innovation Program of Jiangsu Province (SJCX190264), and by National Key R & D Program of China (2018YFA 0702200). Also, this work was supported by the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), grant number CE140100049.

Table 8 The Wilcoxon signed-rank test results with RMSE criterion in New South Wales

Proposed model	CA-WOA-HUBER-ELM	h	p-value
Basic benchmark models	BPNN	1	0.00
	SVM	1	0.00
	LSTM	1	0.01
	GA-ELM	1	0.00
	GA-L1-ELM	1	0.00
	GA-L2-ELM	1	0.00
	GA-HUBER-ELM	1	0.01
	WOA-ELM	1	0.00
	WOA-L1-ELM	1	0.00
	WOA-L2-ELM	1	0.01
	WOA-HUBER-ELM	1	0.03
	CA-WOA-ELM	1	0.00
	CA-WOA-L1-ELM	1	0.00
	CA-WOA-L2-ELM	1	0.00
New benchmark models	DWT-PSO-RBFNN	1	0.00
	EEMD-GOA-ELM	1	0.00
	PSR-SVM	1	0.00
	EEMD-MLR-LSTM	1	0.00

References

1. Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2017.
2. Weilin Guo, Liang Che, Mohammad Shahidehpour, and Xin Wan. Machine-learning based methods in short-term load forecasting. *The Electricity Journal*, 34(1):106884, 2021.
3. Hamidreza Zareipour, Claudio A Canizares, and Kankar Bhattacharya. Economic impact of electricity market price forecasting errors: a demand-side analysis. *IEEE Transactions on Power Systems*, 25(1):254–262, 2009.
4. Jingrui Xie, Ying Chen, Tao Hong, and Thomas D Laing. Relative humidity for load forecasting models. *IEEE Transactions on Smart Grid*, 9(1):191–198, 2016.
5. Tinghui Ouyang, Yusen He, Huajin Li, Zhiyu Sun, and Stephen Baek. Modeling and forecasting short-term power load with copula model and deep belief network. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(2):127–136, 2019.
6. Kunjin Chen, Kunlong Chen, Qin Wang, Ziyu He, Jun Hu, and Jinliang He. Short-term load forecasting with deep residual networks. *IEEE Transactions on Smart Grid*, 10(4):3943–3952, 2018.
7. Zhesen Cui, Jinran Wu, Zhe Ding, Qibin Duan, Wei Lian, Yang Yang, and Taoyun Cao. A hybrid rolling grey framework for short time series modelling. *Neural Computing and Applications*, pages 1–15, 2021.
8. Azim Heydari, Meysam Majidi Nezhad, Elmira Pirshayan, Davide Astiaso Garcia, Farshid Keynia, and Livio De Santoli. Short-term electricity price and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm. *Applied Energy*, 277:115503, 2020.
9. Wenquan Xu, Hui Peng, Xiaoyong Zeng, Feng Zhou, Xiaoying Tian, and Xiaoyan Peng. A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. *Applied Intelligence*, 49(8):3002–3015, 2019.
10. KB Lindberg, P Seljom, H Madsen, D Fischer, and M Korpås. Long-term electricity load forecasting: Current and future trends. *Utilities Policy*, 58:102–119, 2019.
11. Ming Dong and Lukas Grumbach. A hybrid distribution feeder long-term load forecasting method based on sequence prediction. *IEEE Transactions on Smart Grid*, 11(1):470–482, 2019.
12. Abdullah Al Mamun, Md Sohel, Naeem Mohammad, Md Samiul Haque Sunny, Debopriya Roy Dipta, and Eklas Hossain. A comprehensive review of the load forecasting techniques using single and

hybrid predictive models. *IEEE Access*, 8:134911–134939, 2020.
13. Mo Zhou and Min Jin. Holographic ensemble forecasting method for short-term power load. *IEEE Transactions on Smart Grid*, 10(1):425–434, 2017.
14. Gopal Chitalia, Manisa Pipattanasomporn, Vishal Garg, and Saifur Rahman. Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks. *Applied Energy*, 278:115410, 2020.
15. Shu Zhang, Yi Wang, Yutian Zhang, Dan Wang, and Ning Zhang. Load probability density forecasting by transforming and combining quantile forecasts. *Applied Energy*, 277:115600, 2020.
16. Furong Ye, Liming Zhang, Defu Zhang, Hamido Fujita, and Zhiguo Gong. A novel forecasting method based on multi-order fuzzy time series and technical analysis. *Information Sciences*, 367:41–57, 2016.
17. Juan Camilo López, Marcos J Rider, and Qiuwei Wu. Parsimonious short-term load forecasting for optimal operation planning of electrical distribution systems. *IEEE Transactions on Power Systems*, 34(2):1427–1437, 2018.
18. Ghulam Hafeez, Khurram Saleem Alimgeer, and Imran Khan. Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid. *Applied Energy*, 269:114915, 2020.
19. Zhuochun Wu, Xiaochen Zhao, Yuqing Ma, and Xinyan Zhao. A hybrid model based on modified multi-objective cuckoo search algorithm for short-term load forecasting. *Applied Energy*, 237:896–909, 2019.
20. Mao Ye and Hai Wang. Robust adaptive integral terminal sliding mode control for steer-by-wire systems based on extreme learning machine. *Computers & Electrical Engineering*, 86:106756, 2020.
21. M Talaat, MA Farahat, Noura Mansour, and AY Hatata. Load forecasting based on grasshopper optimization and a multilayer feed-forward neural network using regressive approach. *Energy*, 196:117087, 2020.
22. Mohammadali Alipour, Jamshid Aghaei, Mohammadali Norouzi, Taher Niknam, Sattar Hashemi, and Matti Lehtonen. A novel electrical net-load forecasting model based on deep neural networks and wavelet transform integration. *Energy*, 205:118106, 2020.
23. Gabriel Trierweiler Ribeiro, Viviana Cocco Mariani, and Leandro dos Santos Coelho. Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting. *Engineering Applications of Artificial Intelligence*, 82:272–281, 2019.
24. Ziyu Sheng, Huiwei Wang, Guo Chen, Bo Zhou, and Jian Sun. Convolutional residual network to short-term load forecasting. *Applied Intelligence*, pages 1–15, 2020.
25. Ehab E Elattar, Nehmdoh A Sabiha, Mohammad Alsharef, Mohamed K Metwaly, Amr M Abd-Elhady, and Ibrahim BM Taha. Short term electric load forecasting using hybrid algorithm for smart cities. *Applied Intelligence*, 50:3379–3399, 2020.
26. Jatin Bedi and Durga Toshniwal. Energy load time-series forecast using decomposition and autoencoder integrated memory network. *Applied Soft Computing*, 93:106390, 2020.
27. Sebastian Maldonado, Agustin Gonzalez, and Sven Crone. Automatic time series analysis for electric load forecasting via support vector regression. *Applied Soft Computing*, 83:105616, 2019.
28. Jinran Wu, You-Gan Wang, Yu-Chu Tian, Kevin Burrage, and Taoyun Cao. Support vector regression with asymmetric loss for optimal electric load forecasting. *Energy*, 223:119969, 2021.
29. Guo-Feng Fan, Li-Ling Peng, Wei-Chiang Hong, and Fan Sun. Electric load forecasting by the svr model with differential empirical mode decomposition and auto regression. *Neurocomputing*, 173:958–970, 2016.
30. Gen Li, Yunhua Li, and Farzad Roozitalab. Midterm load forecasting: A multistep approach based on phase space reconstruction and support vector machine. *IEEE Systems Journal*, 14(4):4967–4977, 2020.
31. Jinran Wu, Zhesen Cui, Yanyan Chen, Demeng Kong, and You-Gan Wang. A new hybrid model to predict the electrical load in five states of australia. *Energy*, 166:598–609, 2019.
32. Eric Ofori-Ntow Jnr, Yao Yevenyo Ziggah, and Susana Relvas. Hybrid ensemble intelligent model based on wavelet transform, swarm intelligence and artificial neural network for electricity demand forecasting. *Sustainable Cities and Society*, 66:102679, 2021.
33. Jian Li, Daiyu Deng, Junbo Zhao, Dongsheng Cai, Weihao Hu, Man Zhang, and Qi Huang. A novel hybrid short-term load forecasting method of smart grid using mlr and lstm neural network. *IEEE Transactions on Industrial Informatics*, 2020.
34. Yi Liang, Dongxiao Niu, and Wei-Chiang Hong. Short term load forecasting based on feature extraction and improved general regression neural network model. *Energy*, 166:653–663, 2019.
35. Xianlun Tang, Yuyan Dai, Ting Wang, and Yingjie Chen. Short-term power load forecasting based on multi-layer bidirectional recurrent neural net-

work. *IET Generation, Transmission & Distribution*, 13(17):3847–3854, 2019.
36. Gopal Chitalia, Manisa Pipattanasomporn, Vishal Garg, and Saifur Rahman. Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks. *Applied Energy*, 278:115410, 2020.
37. Yunxuan Dong, Xuejiao Ma, and Tonglin Fu. Electrical load forecasting: A deep learning approach based on k-nearest neighbors. *Applied Soft Computing*, 99:106900, 2021.
38. Mohammad Navid Fekri, Harsh Patel, Katarina Grolinger, and Vinay Sharma. Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. *Applied Energy*, 282:116177, 2021.
39. Linfei Yin and Jiaxing Xie. Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems. *Applied Energy*, 283:116328, 2021.
40. Jie Yang, Jiuwen Cao, Tianlei Wang, Anke Xue, and Badong Chen. Regularized correntropy criterion based semi-supervised elm. *Neural Networks*, 122:117–129, 2020.
41. Bingzhi Su, Rongjun Mu, Teng Long, Yuntian Li, and Naigang Cui. Variational bayesian adaptive high-degree cubature huber-based filter for vision-aided inertial navigation on asteroid missions. *IET Radar, Sonar & Navigation*, 14(9):1391–1401, 2020.
42. Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114:163–191, 2017.
43. Rui Wang, Jiyang Wang, and Yunzhen Xu. A novel combined model based on hybrid optimization algorithm for electrical load forecasting. *Applied Soft Computing*, 82:105548, 2019.
44. He Bo, Ying Nie, and Jianzhou Wang. Electric load forecasting use a novelty hybrid model on the basic of data preprocessing technique and multi-objective optimization algorithm. *IEEE Access*, 8:13858–13874, 2020.
45. Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.
46. Jiaohui Xu, Wen Tan, and Tingshun Li. Predicting fan blade icing by using particle swarm optimization and support vector machine algorithm. *Computers & Electrical Engineering*, 87:106751, 2020.
47. Seyedali Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4):1053–1073, 2016.
48. Seyedali Mirjalili. The ant lion optimizer. *Advances in Engineering Software*, 83:80–98, 2015.
49. Seyedali Mirjalili. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89:228–249, 2015.
50. Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67, 2016.
51. Yuchao Gao, Cheng Qian, Zhenghang Tao, Hu Zhou, Jinran Wu, and Yang Yang. Improved whale optimization algorithm via cellular automata. In *2020 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 34–39. IEEE, 2020.
52. Chao Lu, Liang Gao, and Jin Yi. Grey wolf optimizer with cellular topological structure. *Expert Systems with Applications*, 107:89–114, 2018.
53. Jitong Ma, Hao Liu, Chen Peng, and Tianshuang Qiu. Unauthorized broadcasting identification: A deep lstm recurrent learning approach. *IEEE Transactions on Instrumentation and Measurement*, 69(9):5981–5983, 2020.
54. Bo Zhang, Runhua Tan, and Cheng-Jian Lin. Forecasting of e-commerce transaction volume using a hybrid of extreme learning machine and improved moth-flame optimization algorithm. *Applied Intelligence*, pages 1–14, 2020.
55. Jian Ge, Han Li, Hongpeng Wang, Haobin Dong, Huan Liu, Wenjie Wang, Zhiwen Yuan, Jun Zhu, and Haiyang Zhang. Aeromagnetic compensation algorithm robust to outliers of magnetic sensor based on huber loss method. *IEEE Sensors Journal*, 19(14):5499–5505, 2019.
56. Gonggui Chen, Lijun Li, Zhizhong Zhang, and Shuaiyong Li. Short-term wind speed forecasting with principle-subordinate predictor based on conv-lstm and improved bpnn. *IEEE Access*, 8:67955–67973, 2020.
57. Jun Zhang, Yu-Fan Teng, and Wei Chen. Support vector regression with modified firefly algorithm for stock price forecasting. *Applied Intelligence*, 49(5):1658–1674, 2019.