A Novel Neural Network-based Multi-objective Evolution Lower Upper Bound Estimation Method for Electricity Load Interval Forecast

He, Yaoyao; Zhu, Jianhua; Wang, Shuo

DOI:
10.1109/TSMC.2024.3352665

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
He, Y, Zhu, J & Wang, S 2024, 'A Novel Neural Network-based Multi-objective Evolution Lower Upper Bound Estimation Method for Electricity Load Interval Forecast', *IEEE Transactions on Systems, Man and Cybernetics: Systems*. https://doi.org/10.1109/TSMC.2024.3352665

A Novel Neural Network-based Multi-objective Evolution Lower Upper Bound Estimation Method for Electricity Load Interval Forecast

Yaoyao He, *Member, IEEE*, Jianhua Zhu and Shuo Wang, *Member, IEEE*

***Abstract*—Currently, an interval prediction model, lower and upper bounds estimation (LUBE) which constructs the prediction intervals (PIs) by using the double outputs of the neural network (NN) is growing popular. However, existing LUBE researches have two problems. One is that the applied NNs are flawed: feedforward neural network (FNN) cannot map the dynamic relationship of data and recurrent neural network (RNN) is computationally expensive. The other is that most LUBE models are built under single-objective frame in which the uncertainty cannot be fully quantified. In this paper, a novel wavelet neural network with direct input-output links (DLWNN) is proposed to obtain PIs in a multi-objective LUBE frame. Different from WNN, the proposed DLWNN adds the direct links from the input layer to output layer which can make full use of the information of time series data. Besides, a niched differential evolution non-dominated fast sort genetic algorithm (NDENSGA) is proposed to optimize the prediction model, so as to achieve a balance between estimation accuracy and the average width of the PIs. NDENSGA modifies the traditional population renewal mechanism to increase population diversity and adopts a new elite selection strategy for obtaining more extensive and uniform solutions. The effectiveness of DLWNN and NDENSGA is evaluated through a series of experiments with real electricity load data sets. The results show that the proposed model has better performance than others in terms of convergence and diversity of obtained non-dominated solutions.**

***Index Terms*—Short-term load interval prediction, uncertainty forecasting, wavelet neural networks, multi-objective optimization.**

I. Introduction

SHort-term load forecasting of power systems can not only help operators to reduce costs, but also assist the power market to save momentous materials for environmental protection. Therefore, it has been increasingly essential for a reliable power system operation in the long time. However, load forecasting has presented a huge challenge in which the demand for forecasting reliability has greatly increased along with the continuous development of renewable energy and the growing electrical load demand. To address this huge challenge, many researchers have proposed different methods to implement load forecasting, mainly including point forecasting and probabilistic forecasting.

Load forecasting approaches with point prediction had been dominating the research for a long time, which provides an expected value of electricity load. In this category, statistical models (such as autoregressive moving average (ARMA)) [1] had been widely used in early days. Then, machine learning approaches (such as multi-layer perceptron (MLP), recurrent neural networks (RNN) and long short-term memory neural networks (LSTM)) [2], [3] gradually became mainstream benefiting from stronger nonlinear capabilities. In particular, ensemble learning [4] received great attention which combines data decomposition techniques, statistical methods and machine learning methods. Although these statistical and machine learning methods can provide future power load as a guidance, forecasting accuracy is still a major issue due to the uncertainty and dynamics of power systems. Hence, quantifying such uncertainty in forecasting has become extremely vital.

To take consideration of power load uncertainty, the probabilistic forecasting approaches that construct probability density functions, prediction intervals (PIs), and quantiles [5], [6], are proposed in recent years. From the perspective of power system operators, interval forecasting can simply and effectively assist them in decision making. Thus, constructing PIs plays a significant role in uncertainty forecasting.

Traditional parametric interval prediction models like fuzzy inference, Gaussian process and Bayesian method [7], [8] have to come up with a prior assumption about the shape of the error distribution in which the parameters of distribution are estimated based on historical data. The accuracy is greatly affected by the effect of relevant numerical prediction. Then, Khosravi et al. [9] proposed the lower and upper bound estimation (LUBE) method through nonlinear computation of the neural network with double outputs. Without minimizing the sum of squared errors or weight decay cost function, it outputs the PI with a high coverage probability and a small width. The accuracy of this method is greatly affected by the choice of NN. However, in the current research on LUBE, the applied NN models are flawed. Quan et al. [10] used feedforward neural networks (FNN) in which each neuron can only connect to the neurons in the previous layer. In spite of FNN's great nonlinear processing capability, it does not take into account the inherent relationships between data, leading to the issue that the accuracy of time series forecasting does not achieve the expectation. Therefore, algorithms using recurrent neural network (RNN) [11], [12] were proposed. Its

Manuscript received June, 2022; revised XXXX, 2022. The authors would like to thank the National Natural Science Foundation of China(Nos.72171068, 71771073.). This work was partially supported by the Anhui Provincial Natural Science Foundation for Distinguished Young Scholars (2108085J36).

Y. He and J. Zhu are with the School of Management, Hefei University of Technology, Hefei 230009, China. e-mail: (hy-342501y@163.com; 15501444502@163.com).

S. Wang is with CERCIA, the School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK. e-mail: (s.wang.2@bham.ac.uk)

interrelated structure leads to better prediction than FNN-based algorithms in time series forecasting, but they are extremely computationally expensive. To overcome these problems, Pao et al. [13] proposed a random vector functional link network (RVFL). In this network, the input weights and hidden nodes are all randomly selected. Its structure is similar to the conventional FNN, the only difference is that the RVFL allows the direct input-output links. The direct input-output links of RVFL can map the dynamic input-output relationship preferably in time series prediction. Ren et al. [14] verified that direct input-output links can effectively increase the accuracy of power load forecasting. But the randomization of input weight and hidden node only considers empirical risk, which may lead to the problem of over-fitting.

As a type of FNN, wavelet neural network (WNN) [15] combines the merits of artificial neural network (ANN) and wavelet analysis. It possesses the time-frequency local analysis ability and fast convergence speed. In previous studies, WNN has been proven to be successfully applied to point forecasting [15] and interval prediction [16]. Different from previous studies, this paper proposes a wavelet neural network with direct input-output links (DLWNN), as the base model for multi-objective LUBE. The proposed DLWNN changes the original network structure of WNN, increasing the direct links from the input layer to the output layer. In the face of complex time series data, the input-output links of DLWNN can directly transmit the characteristics of the time series data to the output layer through weighting on the premise of the advantage of wavelet analysis, so as to make full use of the information of time series data and improve the forecasting accuracy. Furthermore, existing researches on WNN [15], [16] focus on its application to single-objective problems, while the DLWNN is built in a multi-objective LUBE frame to obtain high quality PIs.

To the best of our knowledge, deep neural networks (DNNs) have achieved unprecedent success in time series forecasting due to the strong representation ability. For example, Li et al. [17] proposed a temporal convolutional network (TCN) based hybrid PV forecasting framework for enhancing hours-ahead utility-scale PV forecasting. Lu et al. [18] proposed a non-crossing sparse-group Lasso-quantile regression deep neural network to address electricity load forecasting. But as Ref [19] discussed, using multi-objective optimization algorithms to optimize DNNs is extremely complex and time consuming. Therefore, it is an important challenge to balance the training effect and speed of the model. In this paper, WNN is extanded as a multi-hidden layer deep learning framework and the dropout theroy is used to improve its generalization. This deep wavelet neural network (DWNN) is used to compare with the proposed DLWNN on LUBE. In addition, the ablation of DWNN is studied from the aspects of hidden layer number and dropout probability.

The PIs constructed by LUBE have two objectives, the coverage probability and the width. The former represents the probability that the true values are covered by the PIs, while the latter indicates the average width between the upper and lower bounds of PIs. When more actual values are covered by the PIs, the coverage probability is higher, but obviously the widths of the PIs increase. So, the two indices are in conflict with each other. These objective functions are generally nonanalytic and non-differential. But the traditional weight correction method like gradient descent (GD) is only implemented under the condition that the loss function is differentiable. Thus, the single and multi-objective optimization algorithms are gradually used by researchers to optimize the NN in LUBE. In recent years, this multi-objective optimization problem of LUBE is generally handled in two ways, one is to transform the two objectives to a single-objective problem through penalty terms or constraints, and the other is to use a multi-objective optimization algorithm to obtain a set of non-dominated solutions. The method of converting two objectives into a single objective has gained the attention of many researchers because it can obtain reasonable results at a fast speed and is simple to implement. Quan et al. [10] proposed a method to minimize the width of PIs with coverage probability restraints. Furthermore, Quan et al. [20] introduced a novel cost function that assigns weights to two objectives. Both of the above methods use an aggregation method to link the two evaluation indices together. The quality of the built interval depends heavily on the aggregation approach, but it is hard to determine which way of aggregation is more suitable.

In the multi-objective framework, two objectives are synchronously optimized for a set of Pareto-optimal solutions and the trade-off solution is selected among them. Different from existing multi-objective learning algorithms [21], [22], LUBE-based multi-objective learning method simultaneously optimizes the accuracy (coverage probability or estimation error) and dimension (width) of PIs instead of reducing the training error of the cost function. Ak et al. [23] employed the LUBE method to build PIs based on multilayer perceptron neural network (MLP) in a multi-objective framework. Li et al. [24] proposed a knee-based lower upper bound estimation method which uses non-dominated fast sort genetic algorithm II (NSGA II) to train a NN and knee selection criterion to select the best trade-off solution. Various heuristic algorithms can be capitalized on multi-objective problems, such as evolutionary algorithms (EA) [25], [26], particle swarm optimization (PSO) [27], differential evolution (DE) [28], [29]. EA represents a special class of evolutionary algorithms that uses techniques inspired by evolutionary biology. Although it can rapidly provide solutions to both discrete and continuous problems, its local search ability is still weak. In face of the flaws of these algorithms, this paper, based on NSGA II [25], proposed a novel niched differential evolution non-dominated fast sort genetic algorithm (NDENSGA). It added the adaptive differential evolution variation operation before the population cross operation to expand the search scope. To increase the closeness between iterations of the population, the merging mechanism of the population are adjusted, that is, the individuals generated by the adaptive difference evolution variation and cross operation of the current iteration were merged with the individuals generated by polynomial variation operation of the previous iteration. This also proposed a modified elite selection strategy for increasing the universality of the solution set. All the proposed methods have been assessed by the case study with a real-world electricity load.

To sum up, the major contributions of this paper are as follows:

1) To overcome the problems that are the computational complexity of RNN and the absence of dynamic nature of FNN in LUBE, a novel DLWNN with PIs outputs is proposed for LUBE, which is faster and more accurate in constructing PIs of power load data.

2) To increase local search capability and population diversity, a novel NDENSGA is presented to optimize the new LUBE model.

3) DLWNN and NDENSGA are firstly employed in LUBE to construct PIs of power load data.

4) The obtained experiment results show that the solutions obtained by DLWNN and NDENSGA perform best compared with conventional benchmark NNs and optimization algorithms, and the quality of constructed PIs are significantly improved compared with other interval prediction models.

5) The disturbance and multi-step experiment results indicate the strong robustness of the proposed model.

6) The online NDENSGA-optimized DLWNN is successfully designed and its feasibility is verified in real data.

The rest of this paper is arranged as follows. Section II introduces the basic knowledge of PIs. In Section III, the proposed DLWNN-based LUBE model, NDENSGA, the implementation of multi-step forecasting and online prediction are elaborated. The electricity load forecasting case studies are presented in Section IV. Finally, Section V draws the conclusion of this paper.

II. Preliminaries

The evaluation metrics of PIs fall into the following two categories. One is the accuracy that represents whether the uncertainty of the data is effectively quantified. The other is the width of intervals. For power system operators, narrower PIs means more valuable information. Main assessment metrics of PIs are introduced in this section.

A. Prediction Interval Coverage Probability

Prediction interval coverage probability (PICP) represents the prediction quality, indicating whether the future power load is within the lower and upper bounds. Therefore, a higher PICP indicates better forecasting intervals. It can be mathematically defined as follows [30]:

$$\mathrm{PICP} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i \tag{1}$$

where N is the number of test data samples, and α_i is a binary value described as follows:

$$\alpha_i = \begin{cases} 1,\ y_i \in [L_i, U_i] \\ 0,\ y_i \notin [L_i, U_i] \end{cases} \tag{2}$$

where y_i is the target of test samples, L_i and U_i are lower and upper bound of the PIs. From most studies, PICP is usually set to be higher than a pre-defined confidence level, and the incongruent PIs should be cast away.

B. Prediction Interval Normalized Average Width

Although PICP is a crucial indicator for forecasting intervals, it is easy to obtain an extremely high PICP with an amply wide width. Too wide intervals contain little useful forecasting information, becoming meaningless in the worst case. So, prediction interval normalized average width [31] (PINAW) is defined to measure the width of PIs and calculated as follows:

$$\mathrm{PINAW} = \frac{1}{N \cdot W}\sum_{i=1}^{N}(U_i - L_i) \tag{3}$$

where W means that the difference between the target maximum and minimum.

C. Prediction Intervals Estimation Error

Binary variables of PICP that express whether the test samples are in the lower and upper bounds. In addition, many forecasting intervals researches based on the PICP index mainly focused on the coverage probability and ignored the risk outside the interval consequently. Thus, Zhou et al. [32] introduced a novel evaluation metric, called prediction intervals estimation error (PIEE), to quantify the forecasting error of PIs. PIEE is defined as follows:

$$\mathrm{PIEE} = \frac{1}{N \cdot W}\sum_{i=1}^{N} Ei \tag{4}$$

where E_i is described as follows:

$$E_{\mathrm{i}} = \begin{cases} (y_i - U_i), & \text{if } y_i \geq U_i \\ 0 & \text{if } L_i < y_i < U_i \\ (L_i - y_i), & \text{if } L_i \geq y_i \end{cases} \tag{5}$$

Compared with PICP, PIEE is better at estimating errors of PIs, but loses the aspect of the width of PIs. In the actual operation of the electricity market, the situation of a greatly low PIEE with extremely large PINAW is faced frequently. To achieve a small PI width and a low forecasting error, PIEE (accruacy) and PINAW (width) are simultaneously optimized in a multi-objective framework for high-quality PIs in this paper.

III. DLWNN Interval Prediction Model Based on NDENSGA

Getting high quantity PIs is the main goal in the field of uncertainty forecasting. LUBE is widely used for constructing PIs for its simplicity and computational efficiency. It often works with neural networks and optimization algorithms. In this section, the novel DLWNN prediction model based on NDENSGA is presented.

A. Wavelet Neural Network with Direct Input-Output Links

Researchers usually recommend single hidden layer feedforward network (SLFN) when solving regression and classification problems because of its efficiency and prospective performance. The nonlinear mapping between inputs and outputs depends largely on the activation function of hidden neurons. Traditionally, the parameters of SLFN are adjusted by the

gradient of the loss function. However, in LUBE, nonanalytic objective functions make this approach infeasible. The parameters of NN are optimized by the meta-heuristic algorithm, the structure and activation function of NN becomes more important. Unfortunately, this kind of architecture of SLFN cannot take full advantage of the historical data. Therefore, the RVFL [13] model, in which the input-hidden layer weights and the number of hidden nodes are randommized, is noticed. Compared with the traditional SLFN, its structure increases the links from the input layer to the output layer. The input-output links of NN can improve the generalization ability by nonlinear transformation of the original features of the data and make better use of the dynamic information existing in the data. The randomization method is very effective in solving the problem that the traditional NN cannot converge globally when propagating back. But it is inapplicable in LUBE.

Based on the above situations, this paper proposes a novel neural network model called wavelet neural network with direct input-output links (DLWNN) which employs the direct input-output links in WNN for LUBE. Compared with the Fourier transformation, wavelet analysis can better analyze the local features of signals through the transformation of wavelet basis functions. DLWNN takes the wavelet basis function as the transfer function of the hidden layer nodes can well capture the characteristics of time series. In addition, the input-output links are able to solve the structure flaw of WNN. The structure of DLWNN is a fully connected single hidden layer wavelet neural network with direct input-output links as shown in Fig. 1.

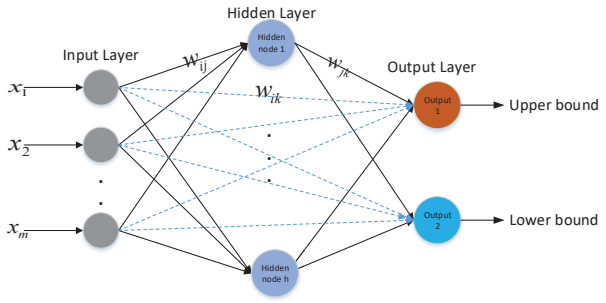

Fig. 1. The structure of DLWNN

In the input layer, each neuron x_i, $i \in \{1, 2, ..., m\}$ represents a feature of the input vector and passes not only to the hidden layer neurons but also to the output layer neurons. Each hidden layer neuron h_j, $j \in \{1, 2, ..., l\}$ calculates nonlinear weighted values of the input layer:

$$h_j = f\left(\frac{\sum_{i=1}^{m} w_{ij}x_i - b_j}{a_j}\right) \quad j = 1, 2, ..., l \tag{6}$$

where m is the number of input layer neurons, l is the number of hidden layer neurons, w_{ij} expresses the weight from the input layer to the hidden layer. Based on DLWNN, b_j is the shift factor and a_j is the scalability factor of wavelet basis function. $f(\cdot)$ is the wavelet basis function defined as follows:

$$f = \cos(1.75x)e^{-x^2/2} \tag{7}$$

The two output neurons O_k , $k \in \{1, 2\}$, are the weighted sum of the outputs of the hidden layer and the outputs of the input layer which can be computed as follows:

$$O_k = \sum_{j=1}^{l} w_{jk}h_j + \sum_{i=1}^{m} w_{ik}x_i \tag{8}$$

where w_{jk} is the weight from the hidden layer to the output layer and w_{ik} means the weight from the input layer to the output layer.

B. Niched differential evolution non-dominated fast sort genetic algorithm

In order to optimize PIEE and PINAW concurrently, this paper searches for the optimal parameter combination ω^* of DLWNN in a multi-objective framework. The objective function is as follows:

$$\begin{aligned} \text{Objectives}: &\quad \text{Finding optimal } \omega^* \text{ to} \\ &\quad \text{Min PIEE}(\omega^*) \\ &\quad \text{Min PINAW}(\omega^*) \\ \text{Constraints}: &\quad \text{PINAW}(\omega^*) > 0 \end{aligned}$$

NSGA II [25] has been used in the LUBE model because of its fast convergence and low computation complexity. But it controls the process of parental crossover and variation according to fitness values, which leads to the decline of local search ability and insufficient diversity of the population. Meanwhile, the mutation vector of differential evolution algorithm [28] generated by parent difference vector can improve the effect of approaching the optimal solution set. Therefore, based on NSGA II, this paper proposes the NDENSGA in which the adaptive differential evolution mutation operation is added before the crossover operation. To expand the search scope and increase the connection between population iterations, NDENSGA integrates the individuals generated by adaptive difference evolution variation and crossover of the current iteration with the individuals generated by polynomial variation of the previous iteration. The adaptive differential evolution mutation operation can be defined as follows:

$$V_i(It) = q{\cdot}bestX(It-1)+q{\cdot}X_i(It)+F{\cdot}(X_{r1}(It)-X_{r2}(It)) \tag{9}$$

where It is the current iterative number,$X_i(It)$, $X_{r1}(It)$, $X_{r2}(It)$ respectively represent the three individuals of the $It-$th generation population and $i \neq r1 \neq r2$, $V_i(It)$ indicates the individual for the It-th generation self-adaption differential evolution variation population. It is different from the traditional differential evolution algorithm because our algorithm uses $bestX$ which is the highest non-dominant rank and largest crowding degree individual in the previous generation. The function of $bestX$ is to optimize the population in the direction of the target. Both q and F are decreasing adaptive parameters which can be calculated by the following equations:

$$q = \frac{MaxIt - It}{MaxIt} \tag{10}$$

where $MaxIt$ represents the maximum number of iterations.

$$F = \frac{(F_{\max} - F_{\min}) \cdot (MaxIt - It)}{MaxIt} + F_{\min} \tag{11}$$

where the values of $F_{\max}$ and $F_{\min}$ are the maximum and minimum of inertia weights.

In addition, NDENSGA introduces an elite selection strategy [33] which can be summarized as the following steps:

1) Assuming that the sorted population is P_u. The subpopulation of P_u is P_{fst}, in which the value of non-dominant rank is 1. The population quantity of P_{fst} is $npop$, and the predetermined population number is pop. If $npop \leq pop$, the new population should take the first pop individuals of P_u. Oppositely, if $npop > pop$, the Euclidean distances between individuals are calculated firstly in P_{fst} which can be defined as follows:

$$E(i,j) = E(j,i) = \sqrt{\sum_{t=1}^{n} (y_{it} - y_{jt})^2} \tag{12}$$

where i and j are individuals in P_{fst} , n means the number of targets, y_{it} represents the tth target values of the individual i and y_{jt} is in a similar way.

2) According to (12), a Euclidean distance matrix E and $E(i,i) = 0,\ i = 1,2,...,npop$ can be obtained. For each individual i ($i = 1,2,...,npop$), the minimum value of $E(i,j)$ ($j = 1,2,...,npop,\ i \neq j$) is $\{m_{1_i}\}(i = 1,2,...,npop)$. Then the minimum value $m_{1_{k1}}$ in $\{m_{1_i}\}(i = 1,2,...,npop)$ should be found and the individual $k1$ is removed.

3) For each individual i ($i = 1,2,...,npop$), finding the minimum value $\{m_{2_i}\}(i = 1,2,...,npop)$ in $E(i,j)$ ($j = 1,2,...,npop,\ i \neq j$) except $\{m_{1_i}\}(i = 1,2,...,npop)$. Then, the minimum value $m_{2_{k2}}$ in $\{m_{2_i}\}(i = 1,2,...,npop)$ should be located and the individual $k2$ is removed.

4) Continue to remove individuals following 2) and 3) until $npop = pop$.

C. Model Implementation

Based on DLWNN, the LUBE model is run to provide future load PIs through the majorization of the NDENSGA. The multi-objective framework is guided by PIEE and PINAW. The model implementation flowchart is shown in Fig. 2 The important steps are discussed below.

1) Dataset partitioning and preprocessing. To train the forecasting model, the original dataset is divided into training data and test data. The training data are further partitioned into training and validation sets. The validation data is used to find the best hyper-parameter setting of the model. The test data are used to estimate the generalization ability of the model. All the data are normalized to [0,1].

2) Model parameters initialization. Setting It as 1 and $P(It)$ represents the $It-$th generation population. In $P(1)$, the parameters of each individual (w_{ij}, w_{jk}, w_{ik}, b_j and a_j) are initialized as random numbers in [-1,1]. $MaxIt$, pop of NDENSGA should also be predetermined.

3) Fitness value calculation. Each individual i of $P(It)$ can be deemed as a potential solution for DLWNN. Inputting the training set into the DLWNN to obtain the PI through the nonlinear weighted calculation. Then, the PIEE and PINAW of each individual can be calculated.

4) Updating population. After the initial population $P(1)$ is randomly generated, the individuals of the population are evolved towards the target functions. A variant population $P_1(It)$ attached to DE is generated according to (9)-(11).

Then, select the parent population $P_{par}(It)$ from $P(It)$ in which two individuals are randomly selected and the one with better non-dominant grade and crowing degree is kept. The progeny population $P_2(It)$ is generated based on the analog binary crossover which can be calculated by the following functions:

$$C_1(It) = \begin{cases} 0.5 \times [(1+\gamma(It))X_1(It) \\ +(1-\gamma(It))X_2(It)] & r_1(It) < 0.5 \\ X_1(It) & else \end{cases} \tag{13}$$

$$C_2(It) = \begin{cases} 0.5 \times [(1-\gamma(It))X_2(It) \\ +(1+\gamma(It))X_2(It)] & r_2(It) < 0.5 \\ X_2(It) & else \end{cases} \tag{14}$$

$$\gamma(It) = \begin{cases} (2u(It))^{1/(\varphi+1)} & u(It) < 0.5 \\ 1/(2(1-u(It)))^{1/(\varphi+1)} & else \end{cases} \tag{15}$$

where $X_1(It)$ and $X_2(It)$ are the individuals of the $It-$th generation parent population $P_{par}(It)$, $C_1(It)$ and $C_2(It)$ are the offspring individuals. In addition, $r_1(It)$ and $u(It)$ are random numbers while φ is a parameter that needs to be set in advance.

The next step is to integrate $P(It)$ into $P_1(It)$, $P_2(It)$ and $P_3(It)$ (when $It = 1$, $P_3(It) = P(1)$) and sort this new population based on the non-dominance ranks and the crowding degrees. Here, $P_3(It)$ represents $It-$th generation mutation population. The number of the new population is usually larger than the preset population amount. Thereupon, the elite selection should be implemented according to (14). After this, a new population $P(It+1)$ is created.

In the last step, a polynomial variation population $P_3(It+1)$ based on the population $P_h(It)$ which ranks highest in the non-dominant level of $P(It+1)$ is evolved to improve the search capability and increase the connection between population iterations. $P_3(It+1)$ is defined as follows:

$$C_3(It+1) = \begin{cases} X_3(It) + \Delta(It) & r_3(It) < 0.1 \\ X_3(It) & else \end{cases} \tag{16}$$

$$\Delta(It) = \begin{cases} (2\mu(It))^{1/(\varphi+1)} - 1 & \mu(It) < 0.5 \\ (1-2(1-\mu(It)))^{1/(\varphi+1)} & else \end{cases} \tag{17}$$

where $X_3(It)$ is the $It-$th generation parent individual in $P_h(It)$, $C_3(It+1)$ is the offspring individual for $P_3(It+1)$, $r_3(It)$ and $\mu(It)$ are the random numbers.

5) population iteration. $It = It + 1$.

6) Termination criterion. The training is terminated when the current generation It is arriving at the maximum iteration $MaxIt$ in this work.

7) The constructed PIs for test data. When the training has been completed, the optimal parameters of the DLWNN model are obtained. Possessing this optimal forecasting model, the model can easily construct PIs on test data. The relevant metrics are calculated to evaluate the PIs quality.

Algorithm1: NDENSGA

/*Population initialization, It represents the current generation and $P(1)$ represents the initial population*/
set $It = 1$, Initialize population $P(1)$
/* Calculate the fitness value of the initial population*/
for $i = 1 : populationsize(P(1))$ **do**
  Calculate the PIEE and the PINAW values for individual i by (3), (6)
end for
/* Get the dominant relationship between individuals in the initial population*/
for $i = 1 : populationsize(P(1))$ **do**
  for $j = i + 1 : populationsize(P(1))$ **do**
    Compare the dominant relationship between individual i and individual j
  end for
end for
While $It \leq MaxIt$ **do**
  /*Update population, fitness values are calculated for each offspring population obtained*/
  Get the differential evolution variation population $P_1(It)$ according to (11), (12), (13)
  Get the cross-progeny population $P_2(It)$ according to (15), (16), (17)
  /*Population union, $P_3(It)$ represents Itth mutation population and it is $P(It)$ when $It = 1$ */
  Integrate $P(It)$ with $P_1(It)$, $P_2(It)$ and $P_3(It)$ to obtain $P_u(It)$
  /*Dominance comparison*/
  for $i = 1 : populationsize(Pu(It))$ **do**
    for $j = i + 1 : populationsize(Pu(It))$ **do**
      Compare the dominant relationship between individual i and individual j
    end for
  end for
  /*Elite selection, $P_{fst}(It)$ indicates the highest non-dominant rank subpopulation of $P_u(It)$, $npop$ is the quantity of $P_{fst}(It)$, pop is the preset population number*/
  while $npop > pop$ **do**
    Remove individuals of $P_{fst}(It)$ according to (14), obtain an elite population $P(It + 1)$
  end while
  Generate mutation population $P_3(It + 1)$ for next generation accordingto (18), (19)
  $It = It + 1$
end while
output $P(It)$

D. Prediction implementation:single step and multistep forecasting

In short-term power load time series forecasting issues, due to the day similarity of time series changing, the reference value of load output in the forecasting day can be calculated by using the load data in the same historical periods and the adjoining dates according to the time correlation characteristic [34]. In other words, the key of similar day forecasting is to use a series of data at the same time points to predict the value at the same time point in the future day. Moreover, the direct multi-step prediction method [35] is implemented in which the function, rather than the input set, changes during the repetition process. The proposed network executes iteratively for each time step. When historical load data $X = [x_1, x_2, ..., x_j]$ are selected as inputs, for a specific value of $k, k \in \{1, 2..., K\}$, the k-th step estimation of the direct strategy is the k-th step function map of inputs, i.e.,

$$[y^u_{j+k}, y^l_{j+k}] = F(x_1, x_2, ...x_j), k \in \{1, 2, ...K\} \qquad (18)$$

where $X = [x_1, x_2, ..., x_j]$ is the input sequence, y^u_{j+k}, y^l_{j+k} are the k-th step estimations of upper bound and lower bound.

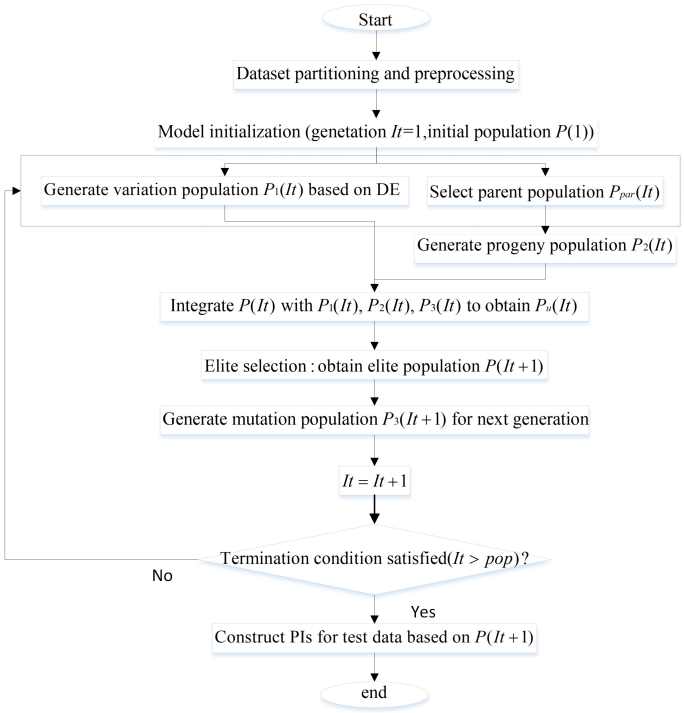

Fig. 2. Evolving DLWNNs in a multi-objective framework.

$F()$ is the training process of DLWNN, j indicates the length of input sequence, and K is the length of steps. When k is equal to 1, it is a single step forecasting model. If the value of k is greater than 1, it is a multi-step forecasting model. This paper takes similar day forecasting method, the input sequence $[x_1, x_2, ..., x_j]$ represents the same time point from day 1 to day j. Each step indicates one day, $[y^u_{j+k}, y^l_{j+k}]$ means estimations of PI of the same time point in day $j + k$.

E. Prediction implementation: online learning

In previous sections, the implementation of the model in the offline state is introduced. However, in the actual scenario, many power systems require real-time forecasting [36], [37]. For example, the Electric Reliability Council of Texas (ERCOT) [38] offers a 1-hour ahead real-time locational marginal price forecast, updated every 5 minutes. Furthermore, the characteristics of the data may change. The offline learning-based forecasting models which are trained only from the training set may not be apply to the dataset outside training set. For solving these problems, online learning [39] is a learning-based method of training and updating, in which the model receives data points sequentially and updates its parameters incrementally based on the new data. Therefore, in this paper, the proposed model is applied to online forecasting. The proposed DLWNN is constantly trained to update the prediction results in real time as the data progresses.

The specific approach is to add sliding window and tuning module. Sliding window technique is known as one approach for solving temporal changes. Each sliding window reads the new data and discards the previous data. Given a time series dataset $\{x_j, y_j\}, j = 1, 2, ..., J$, x_j is a vector composed of

past loads and y_j is the target value to be predicted. As with batch processing, the n-th window is the data batch B_n that includes the following data samples:

$$B_n = \{x_j, y_j\}, j = 1+S(n-1), 2+S(n-1), ..., L+S(n-1) \tag{19}$$

where S is the sliding steps, L is the window length, and S must be less than or equal to L. Fig. 3 shows an example (S=2, L=4).

The tuning module adapts the model hyperparameters to new data in the online learning process. However, adjusting the hyperparameters for each batch of data leads to long computation time. A threshold for the solutions obtained from each batch of data needs to be set. It is worth noting that only non-structural parameters can be adjusted through the online learning because such changes do not modify the structure of the neural network. In this paper, the hyperparameter of the proposed method is the number of iterations. The HV of the Pareto solution obtained from each batch of data is first calculated, and if it is less than 0.9, then the optimum hyperparameter is picked from the given range using grid search (GS) [40].

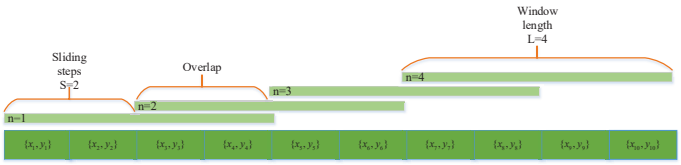

Fig. 3. An example of sliding window of online learning

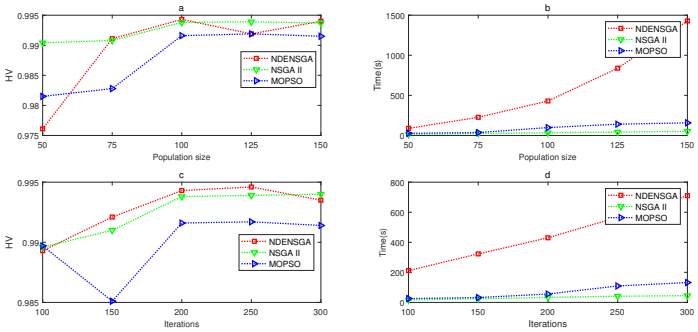

Fig. 4. The effect of population size and number of iterations on the convergence degree and training time of the algorithm.

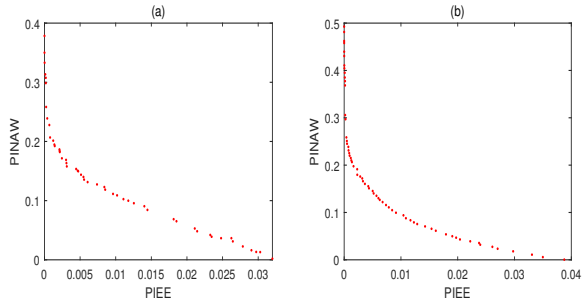

Fig. 5. The Pareto-optimal front received by the proposed prediction model on two datasets. a) dataset 1. b) dataset 2.

IV. CASE STUDIES

To validate the effectiveness and efficiency of the proposed LUBE model based on DLWNN, two real-world datasets of electricity load are conducted in this section. Datasets are introduced firstly, parameter settings of the model, and determine the optimal structure of the proposed neural network. Then, the experiment results of the proposed model are shown. Lastly, the robustness analysis, multi-step forecasting and online learning results of the proposed model are presented.

A. Data Sets

The case study 1 data come from the Australian Energy Market Operator (AEMO) [41] which contains half-hourly electricity load data (unit is MW) recorded in New South Wales (NSW) and Queensland (QLD), Australia (48 points in one day). Dataset 1 use the historical electricity load data of AEMO in NSW which are from 1 January 2014 to 30 March 2014. The whole dataset is further divided into three subsets for training, validation, and test. The training set accounts for about 70% of the whole dataset, validation set and test set severally occupy 15% of the entire dataset. To verify the effectiveness of the proposed method in long data sets, The load data from 1 January, 2014 to 30 June 2015 in QLD are employed for dataset 2. Its three subsets partition is the same as dataset 1.

Furthermore, to verify the validity of the proposed model in online learning, the case study 2 uses actual electricity load data (unit is MW) from German which are public in European network of transmission system operator for electricity (ENTSOE) [42]. This dataset includes 14484 points from January 1, 2023 to May 31, 2023, with 15-min resolution. Because online learning methods adjust hyperparameters online for each batch of data, there is no need to divide a validation set for this dataset to pick hyperparameters. The original data are divided into a training set and a test set in a 3:1 ratio.

B. Parameter Setting

Before estimating the weights and biases, the structure of NN should be determined first. As the output includes the lower and upper bounds, the number of output nodes is set as 2. The number of input nodes represent the lag of the historical load data and has a significant impact on the prediction accuracy of the model. As discussed in section III-D, similar day method is applied in case 1 to implement day-ahead forecasting, which utilizes the same point in historical data to forecast the current load point. Case 2 is applied to validate the effectiveness of the proposed model for real-time online prediction at a shorter period of 15-min in advance in which the similar day approach is inappropriate to adopt. By the trial and error in a large number of literature research [43], [44], The number of input nodes for case 1 and case 2 are set to 7 and 8 respectively.

In this paper, three other SLFNs including WNN, RVFL, extreme learning machine (ELM) are used to compare with DLWNN. The cross-validation method is applied for determining the number of hidden nodes. For the simplification

of the model, the number of hidden nodes varies from 1-10. To evaluate the performance of different Pareto-optimal solutions caused by diverse hidden nodes, a widely used metric hypervolume (HV) used. This index can measure the size of the part of the target space that is dominated by the Pareto optimal solution. Through the cross-validation, the optimal numbers of hidden nodes of DLWNN, WNN, RVFL and ELM on two datasets are shown in Table I. It is worth noting that DLWNN and RVFL have an extremely low number of hidden nodes. This is due to the fact that both DLWNN and RVFL are SLFNs with direct input-output links, based on which the final output of the NN consists of a superposition of the mapping from the hidden layer to the output layer and the mapping from the input layer to the output layer. Compared to the traditional FNN, this approach obviously leads to a significant reduction in the number of hidden nodes. These links emulate the finite impulse response filter (FIR) [14] and substitute for a portion of the hidden nodes, so the model does not suffer from overfitting or underfitting, as demonstrated by experimental results on subsequent test sets. This is also consistent with the conclusion described in Ref [14] that direct input-output links promote the NN more dependent on the mapping between inputs and outputs, reduce training time and increase prediction accuracy.

Regarding NDENSGA, the inertia decreasing weights $F_{\max}$, $F_{\min}$ in adaptive methods [45], φ in analog binary crossover and polynomial variation operation and other parameters are tabulated in Table I. Two conventional algorithms (NSGA II and MOPSO) and are used as benchmarks. It is well known that the population size pop and the number of iterations $MaxIt$ have a great impact on the degree and efficiency of convergence. In this paper, we vary pop in [50:25:150] to explore the impact of population size on the effectiveness and efficiency of three algorithms. Similarly, $MaxIt$ is varied at [100:50:300]. To simplify the experiments, all algorithms are used to optimize the DLWNN on dataset 1 of case 1 with different population size and iterations. Fig. 4 illustrates the HV values and times of the algorithms for different population sizes and iterations on validation set. It can be clearly seen that when the population size rises to 100, the increase in HV is minimal and even decreases on NDENSGA. But the training time of three algorithms increases steadily. The same situation occurs when the number of iterations is 200. Therefore, setting the population size and iterations to 100 and 200 for algorithms is a reasonable choice that takes into account both the optimization effect and the iteration speed. In addition, as in most existing studies [26], [28], the algorithms are set to the same population size and iterations allowing for a fairer comparison.

All the experiments of this paper have been run 30 times by Matlab 2018b software on a PC with Intel Core i9-11900K 11^{th} generation CPU @ 3.50 GHz, 32 GB of DDR4 RAM @3200 MHz, and Windows 10 operating system. In order to effectively compare the performance of the proposed algorithm, a widely used HV metric was calculated for each experiment with the reference point (1,1).

TABLE I
PARAMETERS FOR COMPARATIVE STUDY

Models	Parameters	description	values
DLWNNM	(m_1, m_2)	the input nodes in two case studies	(7,8)
WNN	$(h_{d1}, h_{w1}, h_{r1}, h_{e1})$	the optimal hidden nodes of four SLFNs in dataset 1	(1,4,1,8)
RVFL	$(h_{d2}, h_{w2}, h_{r2}, h_{e2})$	the optimal hidden nodes of four SLFNs in dataset 2	(1,9,2,10)
ELM	O	the output nodes	2
	$F_{\max}$	the maximum of ineritia weight	0.9
NDENSGA	$F_{\min}$	the minimum of ineritia weight	0.4
NSGA II	φ	distribution factor	1
MOPSO	$MaxIt$	the number of iterations	200
	pop	the preset population number	100

TABLE II
PIS CONSTRUCTION RESULTS WITH DIFFERENT PINCS IN THE TEST SET OF TWO DATASETS

	PINC	PIEE	PICP	PINAW
Dataset 1	0.0005	3.8801E-04	0.9380	0.2390
	0.001	7.8745E-04	0.9112	0.2067
	0.0025	2.4471E-03	0.8392	0.1715
	0.005	4.7355E-03	0.7789	0.1496
	0.01	9.6231E-03	0.6298	0.1114
Dataset 2	0.0005	4.7625E-04	0.9765	0.2510
	0.001	9.6072E-04	0.9575	0.2200
	0.0025	2.3074E-03	0.8613	0.1511
	0.005	4.3699E-03	0.8613	0.1511
	0.01	9.1254E-03	0.7296	0.0995

C. Experimental Results

In the training process, through the iteration of NDENSGA, a set of non-dominated solutions can be obtained. With these solutions, a range of different PIs for the test set are constructed, providing more information for decision-making in power system operation.

For a visual comprehending of the distribution of solutions, Fig. 5 shows the Pareto-optimal fronts of the particular run, where the obtained HV result is the closest to the mean value on the test set of two datasets. What can be found is that PIEE and PINAW conflict with each other. When reducing the width of PIs, the forecasting estimation error is increased due to more observed values being out of PIs, vice versa. As shown in Fig. 5, unlike the single-objective framework, the two intrinsically conflict indices are well balanced by the multi-objective framework, which can provide more solutions to meet the demands of power system operators.

It is significant that the actual power operation process is generally prescribed to achieve the value which represents the level of reliability. Therefore, the solution that is as low in width as possible while the value of PIEE wirelessly approaching this value can be selected. This value is usually called prediction interval nominal confidence (PINC) and the PIEE and PINAW should meet the following two requirements:

1) $\mathrm{PIEE} \leq \mathrm{PINC}$

2) PINAW should be as small as possible

Therefore, according to different PINC values, the corresponding solutions can be selected from the particular non-dominated solutions of 30 runs where the obtained HV result is the closest to the mean value on the test set. In Table II, the corresponding solutions and calculated PICP in addition to PIEE and PINAW with 5 different PINC values are shown. The results obtained from these five PINC values can basically meet the three requirements of the actual power system operation: 1) the high coverage with high width which means that the real power load almost in the constructed interval but the width of PIs is too large; 2) the low coverage with low width

which represents that although the width of the constructed PIs is very low, there are many real power loads outside the constructed PIs; 3) the balanced coverage and width which represents that the constructed interval can cover most of the real power load and the width is not particularly large.

To achieve a balance of coverage and width, Figs. 6 and 7 plot the PIs construction results with the PINC of 0.001 of two datasets. Due to the excessive number of samples in dataset 2, 1000 samples are selected to be displayed in Fig. 7 in order to better analyze the relationship between the PIs and the true values. Whether in short or long datasets, the upper and lower bounds of the constructed PIs can almost cover all the target values which indicates that the proposed prediction model is reliable. In addition, implying that the proposed forecasting model can capture the dynamic feature of electric power data well, both the upper and lower bounds have a similar trend with the real load data. To sum up, the proposed method can successfully construct high quality PIs and provide more decision combinations for power systems.

TABLE III
$p-$values OF THE WILCOXON RANK SUM TESTS OF THE FIVE ALGORITHMS FOR THE TEST SET OF TWO DATASETS AT THE SIGNIFICANCE LEVEL $\alpha = 0.05$. RESULTS WITH SIGNIFICANT DIFFERENCES ARE HIGHLIGHTED IN BOLD

		NSGA II	MOPSO	CL-NSGA II	PI-NSGA II
Dataset 1	NDENSGA	**2.93E-06**	**4.23E-07**	**4.47E-02**	**2.99E-11**
	NSGA II	-	**2.46E-02**	**2.87E-03**	**2.98E-11**
	MOPSO	-	-	**2.74E-04**	**3.01E-11**
	CL-NSGA II	-	-	-	**2.89E-11**
		NSGA II	MOPSO	CL-NSGA II	PI-NSGA II
Dataset 2	NDENSGA	**4.61E-09**	**2.86E-08**	**9.06E-11**	**2.92E-11**
	NSGA II	-	1.10E-01	7.28E-01	**3.28E-11**
	MOPSO	-	-	1.65e-01	**2.99E-11**
	CL-NSGA II	-	-	-	**2.14E-11**

TABLE IV
COMPUTATIONAL RESULTS (HV) OF THE FIVE ALGORITHMS FOR THE TEST SET OF TWO DATASETS. THE BETTER RESULTS ARE HIGHLIGHTED IN BOLD.

		Metric (HV)				
		Best	Worst	Average	Median	Std
Dataset 1	NDENSGA	**0.9966**	**0.9936**	**0.9951**	**0.9952**	**9.14E-04**
	NSGA II	0.9964	0.9905	0.9932	0.9936	1.60E-03
	MOPSO	0.9955	0.9784	0.9909	0.9919	4.30E-03
	CL-NSGA II	0.9961	0.9846	0.9941	0.9949	2.32E-03
	PI-NSGA II	0.8542	0.8037	0.8300	0.8296	1.12E-02
Dataset 2	NDENSGA	**0.9975**	**0.9952**	**0.9969**	**0.9970**	**5.17E-04**
	NSGA II	0.9968	0.9659	0.9921	0.9935	6.40E-03
	MOPSO	0.9968	0.9785	0.9940	0.9956	4.50E-03
	CL-NSGA II	0.9960	0.9855	0.9942	0.9952	3.18E-03
	PI-NSGA II	0.8566	0.8298	0.8439	0.8438	4.68E-o3

TABLE V
$p-$values OF THE WILCOXON RANK SUM TESTS OF THE FOUR NEURAL NETWORK MODELS BASED ON NDENSGA FOR THE TEST SET OF TWO DATASETS AT THE SIGNIFICANCE LEVEL $\alpha = 0.05$. RESULTS WITH SIGNIFICANT DIFFERENCES ARE HIGHLIGHTED IN BOLD

		WNN	RVFL	ELM
Dataset 1	DLWNN	**4.80E-04**	**1.56E-05**	**9.42E-05**
	WNN	-	8.60E-01	**4.92E-02**
	RVFL	-	-	**4.35E-02**
		WNN	RVFL	ELM
Dataset 2	DLWNN	**1.64E-09**	**1.18E-09**	**2.92E-11**
	WNN	-	5.18E-02	**4.20E-02**
	RVFL	-	-	**9.01E-08**

TABLE VI
COMPUTATIONAL RESULTS OF THE FOUR NEURAL NETWORK MODELS BASED ON NDENSGA ON TWO DATASETS. THE BETTER RESULTS ARE HIGHLIGHTED IN BOLD.

		Metric (HV)				
		Best	Worst	Average	Median	Std
Dataset 1	DLWNN	**0.9966**	**0.9936**	**0.9951**	**0.9952**	**9.14E-04**
	WNN	0.9960	0.9656	0.9921	0.9937	5.80E-03
	RVFL	0.9958	0.9879	0.9933	0.9938	1.80E-03
	ELM	0.9960	0.9692	0.9902	0.9929	6.40E-03
Dataset 2	DLWNN	**0.9975**	**0.9952**	**0.9969**	**0.9970**	**5.17E-04**
	WNN	0.9968	0.9625	0.9884	0.9919	1.02E-03
	RVFL	0.9967	0.9694	0.9936	0.9947	4.90E-03
	ELM	0.9942	0.9821	0.9892	0.9891	3.00E-03

TABLE VII
DWNN WITH DIFFERENT HIDDEN LAYERS AND DROPOUT PROBABILITY

	Hidden layers	Dropout probability	Optimal hidden nodes	
			Dataset 1	Dataset 2
DWNN-1	2	0.2	(7,9)	(10,10)
DWNN-2	3	0.2	(8,9,8)	(10,7,7)
DWNN-3	4	0.2	(6,6,10,9)	(9,7,10,10)
DWNN-4	5	0.2	(9,10,10,8,10)	(7,9,9,10,10)
DWNN-5	3	0.3	(8,9,10)	(10,10,9)
DWNN-6	3	0.4	(7,7,9)	(10,7,8)
DWNN-7	3	0.5	(8,6,8)	(10,10,9)

D. Effectiveness of the optimization algorithm

In order to prove the effectiveness of the proposed algorithm, two conventional algorithms including NSGA II and MOPSO are used as benchmark algorithms. Furthermore, two state-of-the-art algorithms, competitive learning-NSGA II (CL-NSGA II) [32] and prediction interval-NSGA II (PI-NSGA II) [24] are applied for comparison. CL-NSGA II promotes the evolution of individuals towards the elite individual for increasing the degree of convergence by introducing a competitive learning strategy as follows:

$$P_{\mathrm{a}} \leftarrow P_a + \eta * Rand(0,1) * (P_s - P_a) \tag{20}$$

where η is the control parameter, P_a is randomly selected from the present population and solution P_s is an elite individual.

PI-NSGA II alters the traditional dominance mechanism. For solution p in PI-NSGA II, if its first objective f_1 (which is PIEE) satisfies $f_1 \leqslant 0.01$, then let its non-dominance rank increase, and vice versa. It selects preferentially the individuals in the population with better interval coverage probabilities. Both two algorithms have been proved to obtain high-quality PIs in LUBE. All parameters are set as suggested in the literature. The population size and number of iterations are set to be the same as the other algorithms for a fair comparison. All these algorithms are used to optimize the DLWNN-based LUBE model and the value of HV metric is calculated respectively on two datasets. In order to ensure that the performance of the proposed NDENSGA is statistically different from other algorithms, the Wilcoxon rank sum test [46] at the significance level $\alpha = 0.05$ is used. As shown in Table III, the $p-$values between the proposed NDENSGA and other algorithms are lower than 0.05 which also indicates that the performance of NDENSGA is statistically significantly different from other two algorithms. To further evaluate the validity of the proposed algorithm, Table IV lists the best, worst, average, median and standard deviation (std) values of HV for three algorithms. As it can be seen, the proposed NDENSGA has the best computational results in every respect

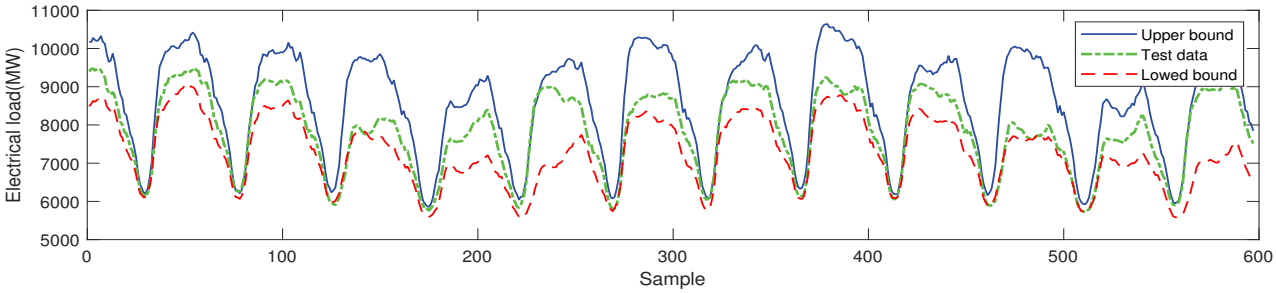

Fig. 6. PIs construction results of DLWNN-based LUBE model in the test set of dataset 1 with PINC of 0.001

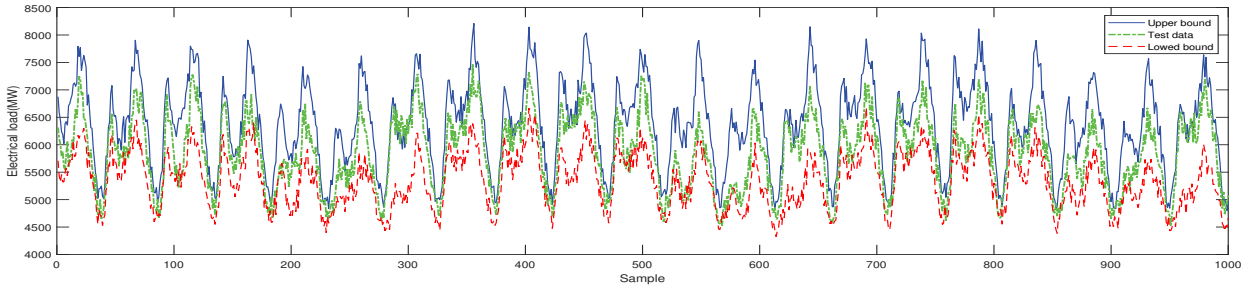

Fig. 7. PIs construction results of DLWNN-based LUBE model in the test set of dataset 2 with PINC of 0.001

on two datasets which means that NDENSGA can obtain more diverse and convergent solutions. In addition, with the lowest standard deviation(std), the NDENSGA method has better stability. The HV of CL-NSGA II is significantly higher than conventional NSGA II, which also demonstrates the role of competitive learning strategies in expanding population diversity. However, PI-NSGA II has the lowest HV in both datasets because it can only find solutions where PIEE performs well, which greatly reduces the breadth and diversity of the algorithm. To visually compare these algorithms, in Fig. 8, the Pareto fronts of five algorithms in the test set of two datasets are shown. Deservedly, each front represents a particular run where the result of HV is the closest to the average. In both short and long datasets, the Pareto-optimal front obtained by the NDENSGA performs better than the other algorithms in convergence and uniformity of solution set. Therefore, possessing high diversity and convergency, the NDENSGA can be used for the electrical load multi-objective interval prediction model as a greatly valid method.

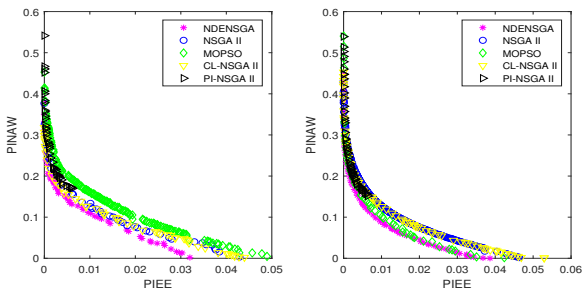

Fig. 8. Comparison of NDENSGA with other four benchmark algorithms on DLWNN-based interval prediction model in the test set of two datasets. a) dataset1. b) dataset 2.

TABLE VIII
THE RUNNING TIME(S) OF DIFFERENT OPTIMIZATION ALGORITHMS AND NEURAL NETWORKS ON TWO DATASETS

Dataset 1(s)	Algorithms	NDENSGA	NSGA II	MOPSO	CL-NSGA II	PI-NSGA II
		380.35	29.12	60.75	61.37	31.15
	NNs	DLWNN	WNN	RVFL	ELM	-
		380.35	440.77	369.59	410.45	-
Dataset 2(s)	Algorithms	NDENSGA	NSGA II	MOPSO	CL-NSGA II	PI-NSGA II
		722.35	178.14	249.48	366.62	203.01
	NNs	DLWNN	WNN	RVFL	ELM	-
		722.35	881.47	773.98	868.83	-

TABLE IX
COMPARISN RESULTS OF DIFFERENT INTERVAL PREDICTION MODELS ON TWO DATASETS

	Dataset1		Dataset2	
	PICP	PINAW	PICP	PINAW
DLWNN-NDENSGA	0.9598	0.2583	0.9575	0.2200
ER-GRU	0.9378	0.2894	0.9562	0.2780
DWT-QRF	0.9124	0.3074	0.9426	0.3247
BOOTSTRAP	0.8179	0.3841	0.8588	0.3799

E. Validity of the proposed neural network

To assess the effectiveness and efficiency of the proposed DLWNN, it is compared with three other benchmark neural networks in the same test set of two datasets. The traditional

TABLE X
PERFORMANCE METRICS AFTER ADDING PERTURBED DATA

	HV	PINC	PIEE	PICP	PINAW
Dataset 1	0.9866	0.0005	3.63E-04	0.9749	0.3217
		0.001	6.35E-04	0.9631	0.3058
		0.0025	2.21E-03	0.8961	0.2507
		0.005	4.27E-03	0.8358	0.2076
		0.01	8.92E-03	0.7186	0.1648
Dataset 2	0.9888	0.005	4.51E-04	0.9825	0.3156
		0.001	8.45E-04	0.9696	0.2827
		0.0025	2.33E-03	0.8997	0.2175
		0.005	4.63E-03	0.8253	0.1784
		0.01	9.98E-03	0.6822	0.1285

TABLE XI
THE 7-STEP-AHAED PIS CONSTRUCTION RESULTS WITH DIFFERENT PINCS IN THE TEST SET OF DATASET 1

HV		PINC														
		0.0005			0.001			0.0025			0.005			0.01		
		PIEE	PICP	PINAW	PIEE	PICP	PINAW	PIEE	PICP	PINAW	PIEE	PICP	PINAW	PIEE	PICP	PINAW
Step 1	0.9966	3.88E-04	0.9380	0.2390	7.87E-04	0.9112	0.2067	2.45E-03	0.8392	0.1715	4.74E-03	0.7789	0.1496	9.62E-03	0.6298	0.1114
Step 2	0.9952	4.91E-04	0.9593	0.3091	8.33E-04	0.9356	0.2485	2.21E-03	0.8610	0.1887	4.94E-03	0.7729	0.1664	9.91E-03	0.6254	0.1215
Step 3	0.9924	3.37E-04	0.9708	0.3422	9.10E-04	0.9228	0.2565	2.50E-03	0.8559	0.2004	4.93E-03	0.7410	0.1680	8.21E-03	0.6552	0.1280
Step 4	0.9920	3.18E-04	0.9722	0.3284	9.62E-04	0.9253	0.2335	2.35E-03	0.8542	0.1785	4.00E-03	0.8299	0.1614	9.65E-03	0.6684	0.1115
Step 5	0.9832	4.98E-04	0.9577	0.2805	4.98E-04	0.9577	0.2805	1.94E-03	0.8856	0.2229	4.95E-03	0.8099	0.1833	1.00E-03	0.6444	0.1278
Step 6	0.9936	4.32E-04	0.9572	0.3269	9.21E-04	0.9376	0.2942	2.41E-03	0.8859	0.2321	4.54E-03	0.8093	0.1563	9.90E-03	0.6560	0.1200
Step 7	0.9779	3.37E-04	0.9621	0.3440	6.36E-04	0.9549	0.3033	2.21E-03	0.8484	0.1347	4.51E-03	0.7220	0.1031	9.10E-03	0.5325	0.0711

TABLE XII
THE DETERMINISTIC PARAMETERS AND DYNAMIC PARAMETER OF FOUR ONLINE METHODS

	Deterministic parameters	value	Dynamic parameter	range
The proposed model	Sliding steps	100	Iterations	[50:10:100]
	Window length	200		
	Hidden nodes	1		
	Population size	30		
	Adjustment threshold	0.9920		
O-QRNN O-QRLSTM O-QRGRU	Sliding steps	100	Training epochs	[50:10:100]
	Window length	200		
	Hidden layers	3		
	Hidden nodes	10		
	Learning rate	0.2		
	Adjustment threshold	0.9920		

TABLE XIII
THE EVALUATION METRICS OF PIS CONSTRUCTED BY FOUR ONLINE METHODS ON TEST SET OF CASE STUDY 2

	PIEE	PICP	PINAW
The proposed model	4.04E-04	0.9525	0.0617
O-QRNN	4.18E-04	0.9508	0.0843
O-QRLSTM	4.98E-04	0.9453	0.1189
O-QRGRU	3.35E-04	0.9725	0.2256

single hidden layer FNN models including WNN and ELM are applied for comparison. With the same input-output direct links structure as DLWNN, RVFL is also implemented to forecast future load intervals. To make a fair comparison, the interval prediction models of these neural networks have respectively selected the optimal number of hidden nodes in the validation set. Moreover, all these NNs-based LUBE models are optimized by the NDENSGA. Table V shows the $p-$values of Wilcoxon rank sum tests between DLWNN and other neural network models. The $p-$values between the proposed DLWNN and other models are all less than 0.05 on two datasets. So, the proposed DLWNN is significantly different from other models. The overall statistical results (HV) of four models are displayed in Table VI. Furthermore, the Pareto-optimal fronts of four NNs-based LUBE models are given in Fig. 9. Similarly, each front is a particular run in which the value of HV is the closest to the mean value of 30 runs. From the above results, DLWNN can obtain very precise solutions with various and evenly distributed populations in two datasets. Moreover, having the same input-output direct connections as DLWNN, the average performance of RVFL is better than the other two basic FNNs. As noted before, these links can capture the characteristics of time series data to improve the forecasting accuracy. To sum up, regarding the PIs constructed by the DLWNN-based LUBE model have the best quality, DLWNN shows the superiority in LUBE.

F. Comparison results with deep wavelet neural network

Deep learning frameworks have been widely used for load forecasting in recent years due to their excellent generalization ability, but the complex multi-layer structure also results in time-consuming. Therefore, in this paper, the traditional WNN is extended to a DNN. Deep wavelet neural network (DWNN) is a network with a multi-hidden layer structure, where the activation function of each layer is the wavelet basis function as presented in (7). In addition, a regularization method, dropout, which is commonly used in deep learning frameworks, is used for the DWNN. Sufficient ablation experiments are conducted to verify the effects of the number of hidden layers and dropout probability on DWNN. As before, the optimal number of hidden nodes for each hidden layer is picked by cross-validation on validation set. All DWNN used in ablation studies are shown in Table VII.

All DWNNs in Table VII are optimized by NDENSGA to obtain PIs. Fig. 10 shows the HV and training time of seven different DWNNs and DLWNN on test sets of two datasets. It can be clearly seen that the number of hidden layers has a great influence on DWNN. The optimal number of hidden layers of dataset 1 and dataset 2 is 4 and 3. When the number of hidden layers is fixed and the dropout probability is increased or decreased, the HV and training time of DWNNs change by an extremely small amount. However, the HV values of the nondominated solution sets obtained by either DNN are lower than those of the proposed DLWNN, proving the effectiveness of the proposed DLWNN. In addition, due to the high complexity of deep learning, the training time of DWNN is much higher than DLWNN. In summary, adding direct input-output links to the structure of WNN can map dynamic input-output relations with shorter training time and obtain the best results. Therefore, it is reasonable to use DLWNN as the forecast model in order to get PIs with higher quality.

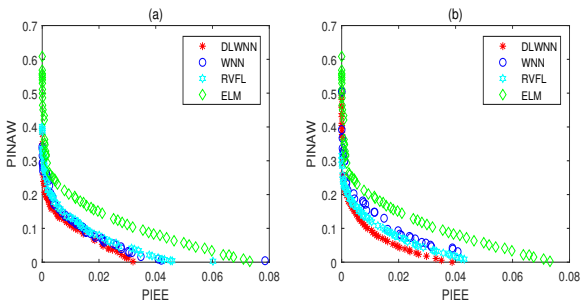

Fig. 9. Comparison of DLWNN with other three benchmark neural networks under the optimization of NDENSGA in the test set of two datasets. a) dataset 1. b) dataset 2.

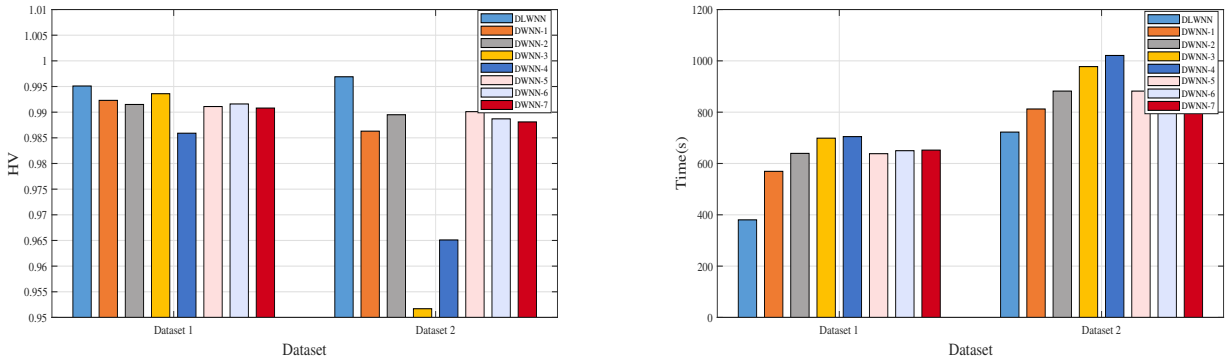

Fig. 10. The HV and training time of DLWNN and DWNNs on two datasets.

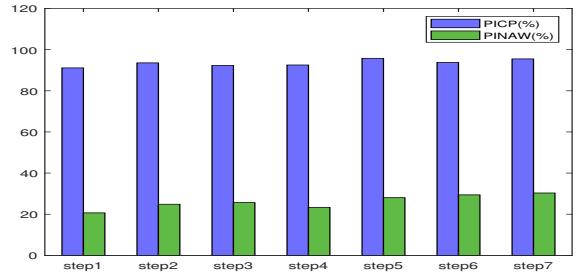

Fig. 11. The PIs results of multi-step prediction in test set of dataset 1 with PINC 0.001

G. computational performance analysis

The training efficiency is an important evaluation index. The computational complexity of training of the optimization algorithm can be calculated as:

$$e\_OA = O(M * MaxIt * P^2) \tag{21}$$

where M is the number of objectives, P is the population size. NDENSGA, NSGA II, CL-NSGA II and PI-NSGA II use fast non-dominated sorting method, but NDENSGA and CL-NSGA II adds offspring populations to expand the population diversity. Obviously, its computational complexity is larger than that of NSGA II. MOPSO updates the personal optimal solution and the global optimal solution based on the dominant relationship. So, the complexity is lowest with the minimum size of P. The computational complexity of four neural networks on each iteration can be calculated as follows:

$$e\_DLWNN = O(S * (d_x * d_h + d_h * d_o + d_x * d_o)) \tag{22}$$

$$e\_WNN = O(S * d_h * (d_x + d_o)) \tag{23}$$

$$e\_RVFL = O(S * d_o * (d_h + d_x)) \tag{24}$$

$$e\_ELM = O(S * d_h * d_o) \tag{25}$$

where S is the number of samples, d_x, d_h and d_o are the dimensions of the input layer, hidden layer and output layer respectively. Compared with S, all the values of d_x, d_h and d_o are small. Therefore, the computational complexity of the four neural networks is actually $O(S)$. In Table. VIII, the specific running times are presented, from which the following results can be shown:

1) The running time of the proposed method (DLWNN+NDENSGA) in two datasets are 380.35s and 722.35s which verifies the realizability of the model on short-term and long-term data.

2) Considering the added offspring population, NDENSGA has the highest computational time cost.

3) It is worth noting that although MOPSO has low computational complexity, MOPSO takes longer to run than NSGA II. The reason may be that MOPSO is easy to fall into local optimal solutions when optimizing such problems.

4) The running time of DLWNN and RVFL is significantly lower than the other two. Although the computational complexity of the four neural networks is all about the linear order of S, DLWNN and RVFL can be optimized with very few hidden nodes. It is the reduction in parameters that leads to shorter optimization times for DLWNN and RVFL

H. Comparision results with other interval prediction models

Previously, this paper compares and verifies the optimization effect of NDENSGA and the superiority of DLWNN. However, all the above are based on the comparison of LUBE. In this section, some advanced interval prediction methods in recent years are implemented to compare with the proposed method. In [47], a new interval construction model based on error prediction was proposed in which the variational mode decomposition (VMD) was used to decompose the complex wind speed time series into simplified modes and two types of GRU models were built for interval prediction (ER-GRU). In [48], the author used the wavelet-based decompositions (DWT) to address the effect of preprocessing load time series, and then the quantile regression forests (QRF) were implemented to build probabilistic forecasts (DWT-QRF). In addition, the bootstrap [49], as a statistical method, is used for comparison.

The parameters of all the above model are set as follows according to the references suggested: the decomposition number of VMD is 4; maximum number of training epochs are 100; batch size for training model is 64 (ER-GRU). The decomposition levels of DWT are 4; the number of decision trees of the RF is 500 (DWT-ARF). The resampling size is set as 200 (Bootstrap).

Table. IX shows the PIs construction results. Since the proposed model (DLWNN-NDENSGA) is implemented in a

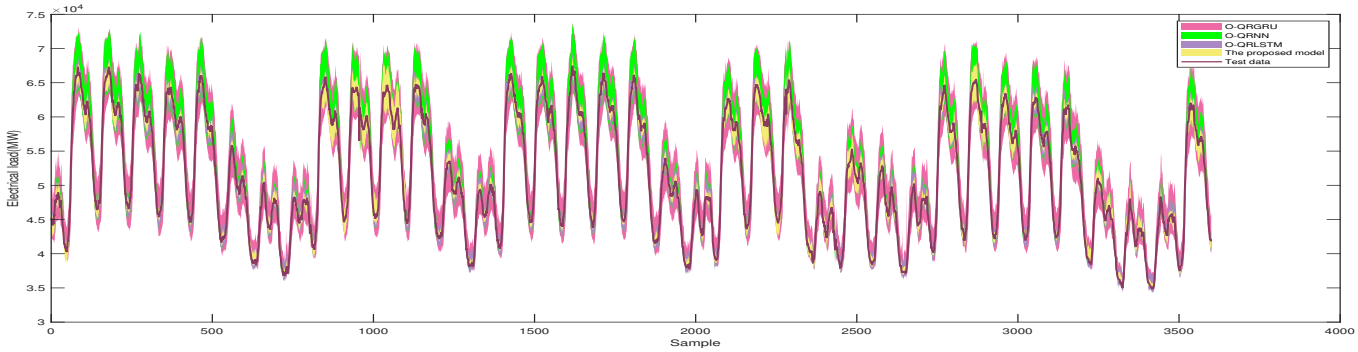

Fig. 12. The PIs construction results of four online models on test set of case study 2

multi-objective framework, the results with PICP higher than 95% and the lowest width are selected in the table. It can be seen that compared with other models, the PIs constructed by the proposed model cover most true values. Moreover, the average width of these PIs is lowest which means that the PIs constructed by the proposed model have the highest quality.

I. Robustness analysis of the proposed model

When there is noise in the training data, the prediction effect of the model will be affected. In the case of data contamination, it is necessary to verify the robustness of the model. One common way to simulate data contamination in a real-world environment is to add perturbed data [50]. The specific procedures are as follows: First, the dataset is normalized in the range (0,1). Secondly, 5% stochastic disturbances with uniform random numbers in [-0.05,0.05] are added to the normalized data.

The robustness analysis experiments are conducted on the above two datasets with each dataset running 30 times. Table. X shows the performance of the proposed model in the two disturbed data sets at different PINC values (The presented result is the closest to the average value of HV in 30 experiments). As can be seen, although the value of HV decreases, the proposed model constructs PIs with high coverage probability and moderate width under the PINC with higher requirements, and obtains extremely narrow PIs under the PINC with lower requirements. Thus, the proposed model has good robustness and low sensitivity to perturbed data.

J. Experimental Results-multistep forecasting

As discussed in Section III-D, this paper uses the proposed model to achieve 7-step-ahead forecasting in dataset 1. Similarly, for each k-step-ahead model, 30 experiments (the result in which the obtained HV is the closest to the mean value is shown) have been run for avoiding occasionality. With five values of PINC, the 7-step-ahead interval prediction results are presented in Table XI. The proposed model all achieves the preset PINCs from step 1 to step 7, which verifies that the proposed model can obtain a widely distributed solution set under multiple steps. It is worth noting that, although the HV of the non-dominated solution set generally shows a decreasing trend with the increase of steps, the 6-step-ahead result is better which might be caused by connections within the data. Furthermore, when the requirement of PINC is high (0.005, 0.001), the coverage probability of PIs obtained by the model imposes a high level. Conversely, when the PINC is set to a low level (0.0025 0.005, 0.01), the constructed PIs is extremely narrow. As shown in Fig. 11, when the PINC is set to 0.001, from step 1 to step 7, the coverage probability of the PI constructed by the proposed model is above 90% and the average width is below 30%. Therefore, the proposed model can be availably applied in multi-step forecasting.

K. Experimental Results-online learning-based forecasting

Previous experiments have verified the effectiveness and efficiency of the proposed model in day-ahead forecasting with offline state. However, in the actual scenario, power system scheduling and generation planning requires earlier access to future load real-time forecasting. New data patterns resulting from temporal changes cannot be learned from old data. Therefore, the online NDENSGA-optimized DLWNN is implemented to achieve 15 min-ahead real-time interval prediction on case study 2.

As discussed in Section III-E, each data batch generated by siding window technology is input into the proposed NDENSGA-optimized DLWNN in turn to obtain PIs, and the number of iterations of each batch are adjusted online by tuning module. To verify the validity of this method, several comparison methods are designed: online quantile regression neural network (O-QRNN) [51], online quantile regression long short-term memory (O-QRLSTM) [52], online quantile regression gate recurrent unit (O-QRGRU) [53]. These methods, like the proposed model, add the same sliding window and tuning module to achieve online prediction. All the models are implemented on case study 2 and the load inputs are load values with 8 points lagged (The resolution between each point is 15 minutes). All models enable GS to adjust hyperparameters when the HV value of each batch of data is lower than a threshold. The main parameters are shown in Table XII.

As previously discussed, the online NENSGA-optimized DLWNN produces a set of non-dominated solutions on each batch of data. The solutions with PIEE less than 0.0005 and PINAW minimum of each batch of data in the test set are

selected to form the final test set PIs. In Table XIII, the evaluation metrics of four methods are displayed. In comparison with O-QRNN and O-QRLSTM, the proposed method obtains the PI with not only the highest coverage probability but also the lowest width. Although the PI constructed by O-QRGRU can cover the largest number of target values, its width is too high compared to other methods. Such PIs are meaningless. Furthermore, Fig. 12 plots the PIs construction results of four methods on test set. It can be seen that all four methods can cover the vast majority of the true values, but the PI the proposed method is significantly narrower. In summary, the proposed method can be successfully applied to online learning to obtain high quality PIs.

V. CONCLUSION

Electric load forecasting plays an important role in power operators' decision-making. Compared with conventional point forecasting, interval prediction can solve the inherent randomness of data more effectively. Therefore, in this paper, a DLWNN-based LUBE model is proposed to construct dependable PIs. In order to obtain various solutions and balance the conflicting indices, the proposed model is implemented in a multi-objective framework. With two objectives, this paper improves NSGA II and proposes a novel NDENSGA to optimize the above model. Through the test of actual data, the proposed prediction model can construct high quality PIs by balancing between PIEE and PINAW. In comparison with other algorithms, NDENSGA produces the most converged and diverse solutions. Our proposed novel DLWNN model is proved to be the most effective and efficient approach to construct PIs by comparing it with other neural networks. Compared with some advanced interval prediction methods, the performance of the proposed model is still optimal on the two datasets. In addition, it has been verified that the model obtains high-quality multi-step-ahead PIs, and exhibits strong robustness in the presence of data interference.

In real scenario, the characteristics of power load data often change, which leads to the failure of traditional off-line forecasting methods. Real-time forecasting has also become an important auxiliary means for increasing power systems. In order to solve these problems, this paper designs the online NDENSGA-optimized DLWNN to achieve 15 min-ahead real-time interval prediction. In a recent dataset, the effectiveness of the proposed online model is verified. In the future, the work is not limited to the electrical load prediction. Such multi-objective interval prediction, and even the probability density prediction model can be used to quantify the uncertainty of future data under different scenarios.

REFERENCES

[1] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on european data," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2213–2219, 2007.

[2] K. George and P. Mutalik, "A multiple model approach to time-series prediction using an online sequential learning algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 976–990, 2019.

[3] S. M. J. Jalali, S. Ahmadian, A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "Automated deep cnn-lstm architecture design for solar irradiance forecasting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 54–65, 2022.

[4] M. De Felice and X. Yao, "Short-term load forecasting with neural network ensembles: A comparative study [application notes]," *IEEE Comput. Intell. Mag.*, vol. 6, no. 3, pp. 47–56, 2011.

[5] Y. Wang, Q. Chen, N. Zhang, and Y. Wang, "Conditional residual modeling for probabilistic load forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 7327–7330, 2018.

[6] Z. Han, W. Pedrycz, J. Zhao, and W. Wang, "Hierarchical granular computing-based model and its reinforcement structural learning for construction of long-term prediction intervals," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 666–676, 2022.

[7] P. Pinson and G. Kariniotakis, "Conditional prediction intervals of wind power generation," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1845–1856, 2010.

[8] A. Bracale, P. Caramia, G. Carpinelli, A. R. Di Fazio, and P. Varilone, "A bayesian-based approach for a short-term steady-state forecast of a smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 1760–1771, 2013.

[9] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Trans. Neural Networks*, vol. 22, no. 3, pp. 337–346, 2011.

[10] H. Quan, D. Srinivasan and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 303–315, 2014.

[11] Z. Cen and J. Wang, "Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer," *Energy*, vol. 169, pp. 160–171, 2019.

[12] F. Yang, W. Li, C. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network," *Energy*, vol. 175, pp. 66–75, 2019.

[13] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, 1992.

[14] Y. Ren, P. Suganthan, N. Srikanth, and G. Amaratunga, "Random vector functional link network for short-term electricity load demand forecasting," *Inform. Sciences*, vol. 367-368, pp. 1078–1093, 2016.

[15] K. Bhaskar and S. N. Singh, "Awnn-assisted wind power forecasting using feed-forward neural network," *IEEE Trans. Sustainable Energy*, vol. 3, no. 2, pp. 306–315, 2012.

[16] Z. Shi, H. Liang, and V. Dinavahi, "Wavelet neural network based multiobjective interval prediction for short-term wind speed," *IEEE Access*, vol. 6, pp. 63 352–63 365, 2018.

[17] Y. Li, L. Song, S. Zhang, L. Kraus, T. Adcox, R. Willardson, A. Komandur, and N. Lu, "A tcn-based hybrid forecasting framework for hours-ahead utility-scale pv forecasting," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4073–4085, 2023.

[18] S. Lu, Q. Xu, C. Jiang, Y. Liu, and A. Kusiak, "Probabilistic load forecasting with a non-crossing sparse-group lasso-quantile regression deep neural network," *Energy*, vol. 242, p. 122955, 2022.

[19] J. Zhu, Y. He, and Z. Gao, "Wind power interval and point prediction model using neural network based multi-objective optimization," *Energy*, vol. 283, p. 129079, 2023.

[20] H. Quan, D. Srinivasan, and A. Khosravi, "Incorporating wind power forecast uncertainties into stochastic unit commitment using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2123–2135, 2015.

[21] Z. Gong, H. Chen, B. Yuan, and X. Yao, "Multiobjective learning in the model space for time series classification," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 918–932, 2019.

[22] H. Chen and X. Yao, "Multiobjective neural network ensembles based on regularized negative correlation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 12, pp. 1738–1751, 2010.

[23] R. Ak, V. Vitelli, and E. Zio, "An interval-valued neural network approach for uncertainty quantification in short-term wind speed prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2787–2800, 2015.

[24] K. Li, T. Zhang, R. Wang, L. Wang, and H. Ishibuchi, "An evolutionary multiobjective knee-based lower upper bound estimation method for wind speed interval forecast," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1030–1042, 2022.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[26] Y. Tian, C. He, R. Cheng, and X. Zhang, "A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 9, pp. 5880–5894, 2021.

[27] I. U. Rahman, Z. Wang, W. Liu, B. Ye, M. Zakarya, and X. Liu, "An n-state markovian jumping particle swarm optimization algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 6626–6638, 2021.
[28] Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A deep evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1798–1810, 2015.
[29] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3954–3967, 2021.
[30] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Trans. Neural Networks*, vol. 22, no. 9, pp. 1341–1356, 2011.
[31] Z. Shi, H. Liang, and V. Dinavahi, "Direct interval forecast of uncertain wind power based on recurrent neural networks," *IEEE Trans. Sustainable Energy*, vol. 9, no. 3, pp. 1177–1187, 2018.
[32] M. Zhou, B. Wang, S. Guo, and J. Watada, "Multi-objective prediction intervals for wind power forecast based on deep neural networks," *Inform. Sciences*, vol. 550, pp. 207–220, 2021.
[33] S. M. K. Heris, "improved-nsga2," 2019, https://github.com/duquanquanquan/improved-NSGA2.
[34] F. Wang, Z. Xuan, Z. Zhen, K. Li, T. Wang, and M. Shi, "A day-ahead pv power forecasting method based on lstm-rnn model and time correlation modification under partial daily pattern prediction framework," *Energ. Convers. Manage.*, vol. 212, p. 112766, 2020.
[35] C. Y. C. S. Nima and G. C. D. Price, "Novel multi-step short-term wind power prediction framework based on chaotic time series analysis and singular spectrum analysis," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 590–601, 2018.
[36] V. X. S. Huu, N. H. M. Giang, H. Takano, N. D. Tuyen *et al.*, "Deep learning-based real-time solar irradiation monitoring and forecasting application for pv system," in *2023 7th International Conference on Green Energy and Applications (ICGEA)*. IEEE, 2023, pp. 40–45.
[37] Y. Ji, R. J. Thomas, and L. Tong, "Probabilistic forecasting of real-time lmp and network congestion," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 831–841, 2017.
[38] "Electric reliability council of texas," http://www.ercot.com/news/press_releases/show/26244.
[39] L. V. Krannichfeldt, Y. Wang, and G. Hug, "Online ensemble learning for load forecasting," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 545–548, 2021.
[40] S. Carcangiu, A. Fanni, A. Mereu, and A. Montisci, "Grid-enabled tabu search for electromagnetic optimization problems," *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3265–3268, 2010.
[41] "Australian energy market operator," http://www.aemo.com.au/.
[42] "European network of transmission system operator for electricity," https://transparency.entsoe.eu/.
[43] U. B. Tayab, A. Zia, F. Yang, J. Lu, and M. Kashif, "Short-term load forecasting for microgrid energy management system using hybrid hho-fnn model with best-basis stationary wavelet packet transform," *Energy*, vol. 203, p. 117857, 2020.
[44] M. Malekizadeh, H. Karami, M. Karimi, A. Moshari, and M. Sanjari, "Short-term load forecast using ensemble neuro-fuzzy model," *Energy*, vol. 196, p. 117127, 2020.
[45] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, 1999, pp. 1945–1950 Vol. 3.
[46] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[47] G. Tang, Y. Wu, C. Li, P. K. Wong, and X. An, "A novel wind speed interval prediction based on error prediction method," *IEEE Trans. Ind. Inf.*, vol. PP, no. 99, pp. 1–1, 2020.
[48] L. Alfieri and P. De Falco, "Wavelet-based decompositions in probabilistic load forecasting," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1367–1376, 2020.
[49] L. Pan and D. N. Politis, "Bootstrap prediction intervals for markov processes," *Comput. Stat. Data Anal.*, vol. 100, pp. 467–494, 2015.
[50] L. Luo, H. Li, J. Wang, and J. Hu., "Design of a combined wind speed forecasting system based on decomposition-ensemble and multi-objective optimization approach," *Appl. Math. Model.*, vol. 89, pp. 49–72, 2021.
[51] J. W. Taylor, "A quantile regression neural network approach to estimating the conditional density of multiperiod returns," *J. Forecasting*, vol. 19, no. 4, pp. 299–311, 2000.
[52] D. Zhang, S. Wang, Y. Liang, and Z. Du, "A novel combined model for probabilistic load forecasting based on deep learning and improved optimizer," *Energy*, vol. 264, p. 126172, 2023.
[53] C. Zhang, C. Ji, L. Hua, H. Ma, M. S. Nazir, and T. Peng, "Evolutionary quantile regression gated recurrent unit network based on variational mode decomposition, improved whale optimization algorithm for probabilistic short-term wind speed prediction," *Renew. Energ.*, vol. 197, pp. 668–682, 2022.

Yaoyao He (Member, IEEE) received the Ph.D. degree in water conservancy and hydropower engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2010. From 2016 to 2017, he was an Academic Visitor with the University of Birmingham, Birmingham, U.K. He is currently a Professor with the School of Management, Hefei University of Technology, Hefei, China.

His current research interests include AI and statistical methods for probability density forecasting and scheduling in power systems and water resources systems. Prof He is currently an Associate Editor for the IEEE Transactions on Industrial Informatics.

Jianhua Zhu received the bachelor's degree from China University of Mining and Technology, Xuzhou, China, in 2020. Currently, he is pursuing the Ph.D degree in management science and engineering at Hefei University of Technology, Hefei, China.

His research interests include smart manufacturing, time series analysis and evolutionary computation with application to neural networks.

Shuo Wang (Member, IEEE) is a lecturer at School of Computer Science, the University of Birmingham, UK. She was a research fellow at the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) in the University of Birmingham between 2011 and 2018. She received the Ph.D. degree in Computer Science from the University of Birmingham in 2011, sponsored by the Overseas Research Students Award (ORSAS) from the British Government.

Dr. Wang's research interests include data stream classification, class imbalance learning and ensemble learning approaches in machine learning, and their applications in social media analysis, software engineering and fault detection.