Robust Adaptive Rescaled Lncosh Neural Network Regression Toward Time-Series Forecasting

Yang Yang, *Senior Member, IEEE*, Hu Zhou, Jinran Wu, Zhe Ding, Yu-Chu Tian, *Senior Member, IEEE*, Dong Yue, *Fellow, IEEE*, and You-Gan Wang

***Abstract*—In time series forecasting with outliers and random noise, parameter estimation in a neural network via minimizing the l_2 loss is unreliable. Therefore, an adaptive rescaled lncosh loss function is proposed in this article to handle time series modeling with outliers and random noise. It overcomes the limitation of the single distribution of traditional loss functions and can switch among l_1, l_2, and the Huber losses. A tuning parameter in the loss function is estimated by using a "working" likelihood approach according to estimated residuals. From the proposed loss function, a robust adaptive rescaled lncosh neural network (RARLNN) regression model is developed for highly accurate predictions. In the training phase of the model, an iterative learning procedure is presented to estimate the tuning parameter and train the neural network in iterations. A new prediction interval construction method is also developed based on quantile theory. The proposed RARLNN model is applied to two groups of wind speed forecasting tasks. The results show that the proposed RARLNN model is more conducive to enhancing forecasting accuracy and stability from the perspectives of noise distribution and outliers.**

***Index Terms*—Outliers, prediction interval (PI), robust loss function, time series forecasting (TSF).**

I. Introduction

TIME-SERIES forecasting (TSF) using a neural network is popular in engineering and physical science [1], [2], [3], such as SOM neural networks [4], extreme learning machine (ELM) methods [5], [6], recurrent neural network [7], long short-term memory (LSTM) [8], encoder–decoder structure [9], and some generalized and combined LSTM models [10]. Generally, the l_2 loss is used as the objective function in most of these methods for TSF machine learning training. Although it works well in general, it is sensitive to outliers. As a result, parameter estimation in a neural network via minimizing the l_2 loss is not reliable in the presence of outliers. Outliers are those points that are different from other sample points in terms of patterns. They will cause the built models to deviate from the correct fitting. To address this problem, this article aims to develop a robust neural network regression for TSF by designing an effective loss function.

In robust loss designs, two types of loss functions are widely used in data modeling with outliers: 1) Vapnik's loss [11] and 2) Huber's loss [12]. Vapnik's loss introduces an insensitive parameter to l_1 loss, eliminating training samples with small noise that fall into the ϵ-insensitive area. It achieves good performance, particularly in support vector regression. Huber's loss combines the advantages of l_1 and l_2 losses. It can control the violation from quantization bounds for two properties [13]: 1) it is differentiable and 2) it is less sensitive to outliers than any quadratic loss and thus can improve the forecasting performance for a data set with outliers. Accordingly, if the noise distribution is known, an appropriate loss function can be easily chosen. However, it is difficult to obtain prior knowledge of the noise in a data set and thus generally impractical to fully determine which loss function should be used.

In the work of [14], a lncosh loss function is proposed, which provides a unifying framework for several mainstream loss functions, such as l_2, l_1, and the Huber losses. The main properties and behavior of lncosh loss are controlled by a tuning parameter. As recommended in [14], the lncosh loss can approximate real noise distributions by adjusting the tuning parameter. However, the selection of the tuning parameter is difficult. Therefore, a new and specialized lncosh-based loss function is proposed in this article for a time series with outliers. It can approximate an unknown noise distribution, guarantee fortitude robustness, and perform intelligently. In addition, this lncosh-based loss function is integrated into a neural network to construct a robust neural network regression structure for TSF.

The confidence interval is another interesting problem in real applications of TSF. Conventional point prediction cannot describe the prediction results regarding the confidence level. So, prediction interval (PI) construction methods are

Manuscript received 24 January 2023; revised 22 April 2023; accepted 30 April 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFA0702200; in part by the Natural Science Foundation of China under Grant 61873130; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191377; in part by the 1311 Talent Project of Nanjing University of Posts and Telecommunications; in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant SJCX190264; in part by the Australian Research Council (ARC) through the Discovery Grant Scheme under Grant DP220100580; and in part by the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) under Grant CE140100049. This article was recommended by Associate Editor S. Agaian. *(Corresponding authors: Yu-Chu Tian; Dong Yue; You-Gan Wang.)*

AQ1

Yang Yang, Hu Zhou, and Dong Yue are with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: yyang@njupt.edu.cn; medongy@vip.163.com).

Jinran Wu and You-Gan Wang are with the Institute for Learning Sciences and Teacher Education, Australian Catholic University, Brisbane, QLD 4000, Australia (e-mail: ryan.wu@acu.edu.au; you-gan.wang@acu.edu.au).

Zhe Ding and Yu-Chu Tian are with the School of Computer Science, Queensland University of Technology, Brisbane, QLD 4000, Australia (e-mail: zhe.ding@hdr.qut.edu.au; y.tian@qut.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2023.3272880.

Digital Object Identifier 10.1109/TSMC.2023.3272880

proposed to describe the reliability of prediction results [15]. An effective method is the lower upper bound estimation (LUBE) method [16], which outperforms delta, Bayesian, and bootstrap techniques. It is incorporated into a gated recurrent unit neural network, achieving good performance [17]. However, all existing interval prediction construction methods are restricted by the strict distribution assumptions of data. To address this issue, a new construction approach is required with undeniable predicting quality.

Our work in this article designs a robust neural network based on an adaptive rescaled lncosh loss function. Then, it applies this method to real wind speed forecasting applications to verify the effectiveness of the method. This neural network can be used to generate deterministic predictions and interval predictions at the same time. For the deterministic predictions, we apply the lncosh loss function and its "working" likelihood function to optimize the weights of the neural network and hyper-parameter of lncosh. To this end, we design an iterative training procedure to obtain the best results of all parameters. Then, the trained model can generate deterministic predictions. From the fitting residuals and deterministic predictions, we design a novel approach to generate interval predictions with different confidence levels.

The main contributions of this article include the following.

1) A new robust adaptive rescaled lncosh neural network (RARLNN) algorithm is proposed for forecasting time series with numerous outliers and complex random noise. It takes an adaptive rescaled lncosh loss function as its objective function and introduces an iterative learning procedure in the training process. The distribution of the adaptive rescaled lncosh loss function approximates three general distributions, i.e., Gaussian, Laplace, and Huber's distributions.
2) A novel PI construction method is presented via the residuals' quantile for the robust rescaled lncosh neural network. PIs with different confidence levels can be established by adding residual series with manually selected quantiles to the obtained predictions of RARLNN.

We apply our method proposed in this article to wind speed forecasting to verify the forecasting of complex time series of the proposed model.

Wind speed forecasting is an important but difficult task in energy management for microgrids system because of its persistent volatility, and intermittent and stochastic fluctuations of wind [18], [19]. The approaches used for wind speed forecasting are abundant, including physical (e.g., numerical weather prediction [20]) and statistical methods (e.g., ARIMA [21] and artificial intelligent neural networks [2]). More recently, some combined methods for handling more complex wind-speed systems have emerged [17], [22], [23]. Chen et al. [23] designed a two-layer procedure for obtaining accurate predictions. It uses ELM, Elamn neural network (ENN), and LSTM to generate multiple groups of forecasts and an ELM-based neural network is used to ensemble all forecasts. Zhang et al. [24] designed a variational local weighted deep subdomain adaptation network for the datasets in which the offline data and online data obey different distributions.

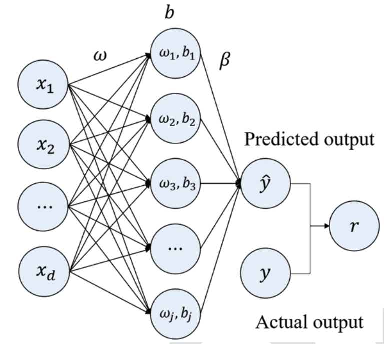


Fig. 1. Structure of a common neural network.

Zhang et al. [25] proposed a novel IMDSSN forecasting model based on a modified Transformer, which includes a multihead ProbSparse self-attention network (MPSN) and a multihead LogSparse self-attention network (MLSN). Wang et al. [26] took the maximum correntropy (MC) criterion as the cost function and designs an OS-ELM-MC model based on ELM. Bandara et al. [10] proposed LSTM-MSNet-DS and LSTM-MSNet-SE based on LSTM and STL (Seasonal and Trend decomposition using Loess). It is worth mentioning that the developed methods for wind speed forecasting considering outliers are limited. In this article, we take some recent state-of-the-art methods as benchmarks and investigate the data sets of wind speed to show the effectiveness of our proposed new neural network. Our code will be released on GitHub later.

This article is structured as follows: In Section II, we introduce an adaptive rescaled lncosh loss. Next, in Section III, we present the RARLNN algorithm, along with its training procedure and the corresponding PI. Then, in Section IV, we demonstrate the effectiveness of the proposed RARLNN and its PI by applying them to two groups of specific wind speed datasets for multiple-step forecasting. Finally, in Section V, we summarize our findings and draw conclusions.

II. Adaptive Rescaled Lncosh Loss

An artificial neural network (ANN) consists of three layers: 1) an input layer; 2) a hidden layer; and 3) an output layer. A simple three-layer ANN is shown in Fig. 1 with d nodes in the input layer, l nodes in the hidden layer, and one node in the output layer. For arbitrary samples (x_i, y_i), $x_i \in R^d$, $y_i \in R$, $i = 1, 2, \ldots, n$, the output of this ANN is defined as follows:

$$y_i = \sum_{j=1}^{l} h_j(x_i)\beta_j = h(x_i)\beta \tag{1}$$

with $h(x_i) = [h_1(x_i), h_2(x_i), \ldots, h_l(x_i)]$ and $\beta = [\beta_1, \beta_2, \ldots, \beta_l]^T$, where $h_j(x_i)$ is the output of the j-hidden node for the i-th sample and β_j is the weight between the j-hidden node and output node. Moreover, $h_j(x_i)$ can be formulated via an activation function as $g(\omega_j \cdot x_i^T + b_j)$. Here, we note H as $[h(x_1); h(x_2); \ldots; h(x_n)]$ and T as $[y_1, y_2, \ldots, y_n]$.

The traditional lncosh loss function is defined as follows [14]:

$$\ell_1 = \frac{1}{\lambda}\log(\cosh(\lambda r)) \tag{2}$$

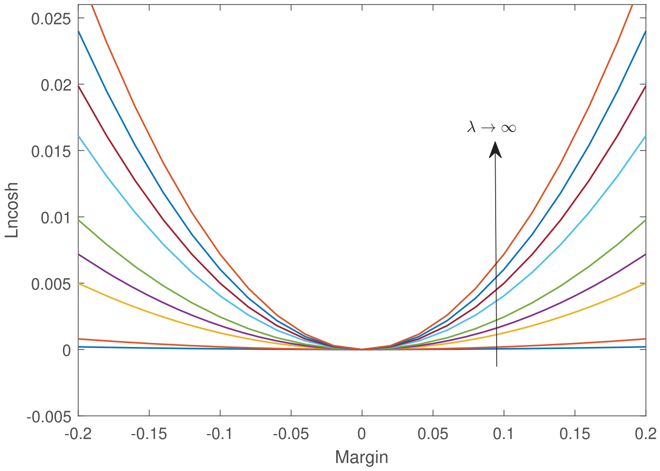


Fig. 2. Graph of the lncosh loss function for different λ.

with $\cosh(\lambda r) = (e^{\lambda r} + e^{-\lambda r})/2$, where $\lambda \in (0, +\infty)$ is a tuning parameter that can control the properties of the lncosh loss function, and the residual for response y is defined as $r = y - \hat{y}$ ($y \in R$ in the following illustration). This loss function provides a joint framework for existing loss functions (l_2 loss, l_1 loss, and Huber's loss) by adjusting parameter λ. The graph of this loss function is shown in Fig. 2 for different values of λ.

Remark 1: As shown in Fig. 2, the lncosh function transforms with the changes of λ. The properties of the lncosh loss function are entirely controlled by λ. It approaches l_1 loss function as λ tends to infinity. In addition, the lncosh loss function approaches l_2 loss function as λ tends to zero. It becomes like Huber's loss function for moderate values of λ. Thus, the lncosh loss function can be modified by adjusting the λ parameter; hence, it is possible to allow switching between different loss functions.

Moreover, in statistics, some distributional assumptions on noise are often made to obtain estimates of the parameters. Hence, we usually assume a likelihood function, and the data are generated by this likelihood function. To bypass this likelihood specification, we can simply nominate a density function, which means the data do not have to be generated by this function. More details can be found in [27], [28], and [29]. Therefore, instead of adopting a likelihood function that supposedly generates the noise, we nominate a likelihood function for the tuning parameter in the loss function. When a constant is free from the parameter of interest, it can be ignored because the constant will not play any role in optimization. Thus, ignoring the constant in the denominator of (2), we first present a rescaled lncosh loss function for the noise distribution based on the traditional lncosh loss function ($\lambda > 0$)

$$\ell_2 = \log(\cosh(\lambda r)). \tag{3}$$

Then, we construct a density function based on the rescaled lncosh loss function following the approach in [29]:

$$f(r; \lambda) = \frac{\lambda}{\pi} \cdot \frac{1}{\cosh(\lambda r)}. \tag{4}$$

Here, the constant (λ/π) comes from the fact that the total probability is 1 (after integration of f).

In statistics, the likelihood function describes how the data are generated randomly whereas the "working" likelihood function is constructed solely for parameter estimation without assuming that the data is generated from this function [29]. Treating this loss function as if it were derived from a log-likelihood function, we obtain the corresponding "working" likelihood function

$$f(r_1, r_2, \ldots, r_n; \lambda) = \prod_{i=1}^{n} f(r_i, \lambda) = \prod_{i=1}^{n} \frac{\lambda}{\pi} \cdot \frac{1}{\cosh(\lambda r_i)}. \tag{5}$$

In our case, we propose the lncosh loss function and adopt the corresponding "working" likelihood function for estimating the tuning parameter λ. Let $\zeta = 1/\lambda$, which is essentially a scale parameter, that is, the errors can be expressed as $\zeta \epsilon_i$, in which ϵ_i has a distribution free of any parameters. This motivates the use of the extended primal objective function

$$L = \sum_{i=1}^{n} \log\left(\cosh\left(\frac{r_i}{\zeta}\right)\right) + n \log(\pi \zeta) \tag{6}$$

where $r_i = y_i - \hat{y}_i$. Note that this "working" likelihood is equivalent to minimizing (3) in terms of the parameters in r_i. This is because the second term in (6) is free from residuals r, that is, optimization concerning residuals r will have the same solutions for a given λ. The advantage of (6) is that it can also be used to obtain a λ value (or $\zeta = 1/\lambda$). The rescaled lncosh loss function can effectively approximate the unknown noise distribution. It also has great robustness and works more intelligently based on the data patterns.

A great advantage of this "working" likelihood approach is that it can provide data-dependent tuning parameters, hyperparameters, and variance parameters [29], [30], [31], [32]. By setting the derivative with respect to ζ to 0, we can obtain an automatic choice of ζ as ζ^*

$$\zeta^* = n^{-1} \sum_{i=1}^{n} r_i \tanh(r_i/\zeta) \tag{7}$$

where $\tanh(r_i/\zeta) = (e^{r_i/\zeta} - e^{-r_i/\zeta})/(e^{r_i/\zeta} + e^{-r_i/\zeta})$. Here, the previous value of ζ can be used in updating ζ^*. Alternatively, the data dependent ζ^* can be obtained by minimizing (6). We define the rescaled lncosh loss of iterative learning ζ^* as adaptive rescaled lncosh loss.

III. Proposed Robust Adaptive Rescaled Lncosh Neural Network

In this section, the proposed adaptive rescaled lncosh loss function is incorporated into neural network training to develop an RARLNN. The procedure of the RARLNN training and its corresponding predictions interval construction are detailed.

A. Objective Function

According to the proposed adaptive rescaled lncosh loss function, the objective function for the proposed RARLNN model with given n samples can be formulated as follows:

$$F(r_i) = \sum_{i=1}^{n} \log(\cosh(r_i/\zeta)) \tag{8}$$

where $r_i = y_i - \hat{y}_i$ and the parameter ζ is estimated by the residuals from the last iteration. In detail, the corresponding optimized problem can be given as follows:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \log\left(\cosh\left(\frac{y_i - h(x_i)\beta}{\zeta}\right)\right). \tag{9}$$

By setting the derivative concerning β to 0, the estimates of β now can be simplified as the solution of the equation

$$\sum_{i=1}^{n} h(x_i)^T \cdot \tanh\left(\frac{y_i - h(x_i)\beta}{\zeta}\right) = \mathbf{0}. \tag{10}$$

B. Add Random Numbers Conforming to Lncosh Distribution

The noise in the time series is priori unknowable. In most practical situations, the distribution of noise in time series is always complex, and hard to describe it with a fixed distribution. Therefore, we add random numbers conforming to the lncosh distribution to enhance the fitting of the model to the real data distribution.

According to the density function of lncosh (4), we can derive the cumulative distribution function (CDF) as follows:

$$F = \int f(u;\lambda)dr = \int \frac{\lambda}{\pi}\frac{1}{\cosh \lambda u}du \tag{11}$$

$$= \frac{2}{\pi}\arctan u. \tag{12}$$

Then, we derive the corresponding inverse function

$$F = \tan\frac{u\pi}{2} \tag{13}$$

where u represents the random numbers conforming to a uniform distribution. We generate random numbers conforming to the distribution of lncosh. Adjust (13) to

$$F = \tan\frac{ku\pi}{2} \tag{14}$$

where the rescaled parameter k is to control the range of generated random numbers,

C. Initial Estimator

During the optimization process, we mainly focus on the optimization problem of the connection weight β and tuning parameter ζ. To obtain consistent and globally optimal estimators for the iterative process, we implement an l_1-norm ELM (l_1-norm-ELM) that replaces the l_2 loss with l_1 loss function (or medium regression) [33] as the initial model. This model is insensitive to outliers. Its optimized problem can be formulated as follows:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} |y_i - h(x_i)\beta|. \tag{15}$$

Remark 2: The initial estimator is crucial in robust estimation. We have adopted the l_1 estimator because of its robustness. In the presence of outliers, l_2 may be subject to large bias, hence causing inconsistency issues.

Thus, following the reference of [34], the initial solution of β can be obtained from the solution of the derivative as:

$$\sum_{i=1}^{n} h(x_i)^T \cdot \operatorname{sign}(y_i - h(x_i)\beta) = \mathbf{0}. \tag{16}$$

In specific experiments, good initial values can be set up in advance to accelerate the iteration speed and improve the trained model's ability.

D. Iterative Process

In the optimization process, each holistic iteration consists of three stages.

1) Adding random numbers conforming to the distribution of lncosh in the data set.
2) Obtaining the estimators of $\hat{\zeta}$ in the adaptive rescaled lncosh loss function by minimizing (6) with the current residuals.
3) Update the parameters β, which takes the adaptive rescaled lncosh loss with the estimated $\hat{\zeta}$ as the objective function for RARLNN training.

In the third stage of each iteration, we substitute the estimated parameter $\hat{\zeta}$ to the adaptive rescaled lncosh function and obtain the connection weight β via (10).

Then, the new residuals r can be obtained to update the parameters $\hat{\zeta}$ in the proposed loss function by minimizing (6). Following the iterative procedure, the proposed RARLNN model can be trained.

The iterative process will be terminated when any of the following conditions are met: 1) The number of iterations exceeds the maximum number of iterations and 2) the value of the adaptive rescaled lncosh loss function achieves precision. Finally, the RARLNN model can be constructed with final parameters (ζ^*, β^*), and the residuals from the trained RARLNN can be obtained as r^*.

Remark 3: In our algorithm, we introduce this idea of ELM, which simply uses pseudo-inversion for parameter updates and a random hidden parameter assignment to reduce the number of parameters. In fact, our algorithm is very different from the original ELM. Specifically, the weights between the input layer and hidden layer and thresholds in the hidden nodes are optimized using the particle swarm algorithm (PSO) instead of randomization. Further, we apply a lncosh loss function for such neural network training. Therefore, to distinguish our algorithm from ELM, the proposed algorithm is renamed as RARLNN.

The computational complexity of our algorithm is mainly caused by calculating the initial estimator, updating $\hat{\zeta}$, and updating $\hat{\beta}$. The computation of PSO in the initial estimator is mainly dominated by the number of iterations and initial particles; these parameters should be appropriately selected according to the number of parameters. Updating $\hat{\zeta}$ needs n times of addition, and its computational complexity is $O(n)$. The updating of parameters $\hat{\beta}$ needs to calculate one Moore–Penrose inverse matrix and one matrix multiplication.

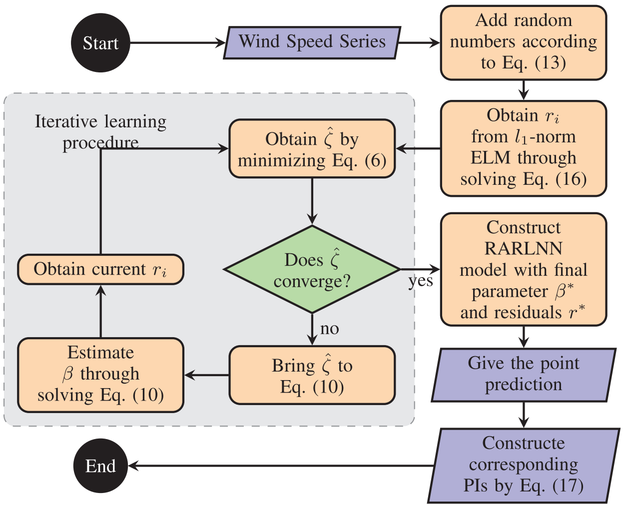


Fig. 3. Structure of our proposed model.

Remark 4: It should be noted that the ELM algorithm is LS-based and has analytical solutions. However, we are interested in robust forecasting in the presence of outliers (possibly many). The existing ELM and neural network frameworks will lead to unreliable estimation and forecasting. for their l_2 loss function. The l_2 loss will give a large loss for outliers which will cause the model to over-fit the outliers. Therefore, the generalization of trained models will be affected. The cost is that the proposed RARLNN is iterative because we do not assume a known proportion of outliers in the data and the resultant data-dependent hyper-parameter values to ensure robustness and effectiveness.

E. Prediction Interval

With respect to the confidence interval, we can further improve the proposed model when it comes to constructing the PIs. Therefore, a novel PI construction method based on the proposed RARLNN model, and the quantile of residuals is presented. In this method, the quantile of residuals is used to obtain PIs with a high confidence level. The upper bound ($\overline{B}_i$) and lower bound ($\underline{B}_i$) of the PIs for the prediction $\hat{y}_i^{test}$ can be constructed as follows:

$$\begin{cases} \overline{B}_i = \hat{y}_i^{test} + \psi_{1-\frac{\alpha}{2}}(r^*) \\ \underline{B}_i = \hat{y}_i^{test} + \psi_{\frac{\alpha}{2}}(r^*) \end{cases} \tag{17}$$

where $\alpha \in [0, 1]$ is the confidence level and ψ is a quantile function of residuals r^*.

In summary, the procedure of our proposed RARLNN can be shown in Fig. 3 and Algorithm 1.

IV. CASE STUDIES

To verify the performance of the proposed RARLNN and its PI construction method, we apply the RARLNN to two groups of real wind speed time series from North China and Vancouver. We will discuss the evaluation criteria, outlier detection, experimental settings, and experimental results.

Algorithm 1: Proposed RARLNN Model

Input: Time series (x_i, y_i), $i = 1, 2, \cdots, n$
Output: ζ^*; β^*; $\overline{B}_i$; $\underline{B}_i$; and point predictions $\hat{y}$

Add random numbers in time series according to Eq. (13).
Obtain initial estimators r from l_1-norm-ELM.
Calculate $\hat{\zeta}$ by Eq. (7).
while $\hat{\zeta}$ is not converging **do**
    Estimate β with the calculated $\hat{\zeta}$.
    Using current $\hat{\zeta}$ and trained model to iterate r.
    Update $\hat{\zeta}$ with updated r.
end while
Take optimized $\hat{\zeta}$ and β as final solutions ζ^* and β^*, and use them to construct RARLNN model.
Obatin final predictions $\hat{y}$ and residuals r^*.
Construct bounds of PIs $(\overline{B}_i, \underline{B}_i)$ according to Eq. (17).

TABLE I
STATISTICAL DESCRIPTION FOR THE DATA SETS

	Data set	Min	Max	Mean	Std	Skewness	Kurtosis
North China	Min 1	0.50	11.90	5.17	2.19	0.24	2.79
	Avg 1	0.50	11.90	5.19	2.19	0.25	2.81
	Max 1	0.60	12.00	5.29	2.20	0.27	2.84
	Min 2	0.10	12.40	4.89	2.12	0.03	2.92
	Avg 2	0.10	12.40	4.90	2.12	0.04	2.94
	Max 2	0.10	12.50	4.98	2.14	0.05	2.95
Vancouver		1.00	63.00	17.28	9.05	0.89	4.11

A. Data

Two data sets are applied to our study to establish forecasting models. The first group is six 5-minutely wind speed data sets from a wind farm in North China; these data are used to comprehensively analyze the effectiveness of the proposed model and PI construction method. The first three data sets are from 11:35, 4 June 2019, to 16:25, 9 June 2019, with a total of 1498 samples; the second three data sets are from 17:30, 5 July 2019, to 22:15, 10 July 2019, with a total of 1498 samples. Each data set of the wind speed data is divided into a training set (67%) and a test set (33%), which are then used to train the forecasting model and verify the accuracy of the trained model, respectively. For illustration, we name the six wind speed data sets: 1) Min 1; 2) Avg 1; 3) Max 1; 4) Min 2; 5) Avg 2; and 6) Max 2. Table I shows the statistical properties of the six data sets. The second data set is from Vancouver, from 17:00, 2 February 2020, to 08:00, 8 July 2020. The training set of this data set accounts for 80% and the test set accounts for 20%. Detailed information is also listed in Table I.

B. Evaluation Criterion

To evaluate the performance of the proposed model, both the mean absolute error (MAE) and root-mean-square error (RMSE) are calculated

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{18}$$

and

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{19}$$

where n is the sample size, $\hat{y}_i$ is the prediction, and y_i is the observation of wind speed.

To evaluate the capacity of the proposed PI construction method, three evaluation criteria are used to describe the properties of PIs, including the PI coverage probability (PICP), the normalized mean PI width (NMPIW), and the combinational coverage width-based criteria (CWC) [16]. They are formulated as follows:

$$\mathrm{PICP} = \frac{1}{n}\sum_{i=1}^{n} c_i, \tag{20}$$

$$\mathrm{NMPIW} = \frac{1}{nR}\sum_{i=1}^{n}\left(\overline{B}_i - \underline{B}_i\right) \tag{21}$$

and

$$\mathrm{CWC} = \mathrm{NMPIW}\left(1 + \varrho(\mathrm{PICP})e^{-\eta(\mathrm{PICP}-u)}\right) \tag{22}$$

where $c_i = 1$ if $y_i \in [\underline{B}_i, \overline{B}_i]$; otherwise $c_i = 0$, $\overline{B}_i$ and $\underline{B}_i$ are the upper bound and lower bound of the PIs, respectively. R is the extreme residual of the wind speed data set. $\varrho(\mathrm{PICP}) = 1$ and u are the confidence level. The constant η is applied to magnify the small gap between PICP and u. The last criterion, CWC, is the combination of PICP and NMPIW. In our experiments, smaller MAE, RMSE, NMPIW, CWC, and higher PICP mean the neural network is superior.

C. Outlier Detection

To validate the robustness of our proposed method, outlier detection is employed to show numerous outliers existing in our investigated wind speed data sets. Here, we simply model all data sets via the "auto.arima" function in the R package "forecast" [35]. The residuals from the autoregressive integrated moving average model are used to analyze the outliers. Specifically, we plot the corresponding Q–Q figures for six residuals of the first group of data sets in Fig. 4. Then, we plot the Q–Q figure and boxplot figure for the data set from Vancouver in Fig. 5, vividly showing outliers in our data sets.

D. Experimental Configuration

Some popular forecasting algorithms are implemented in this study for comparison, that is, ELM, OS-ELM-MC [26], LSTM, LSTM-MSNet-DS [10], LSTM-MSNet-SE [10], and Informer [36]. For the hyper-parameters in OS-ELM-MC and Informer, we select them following the original author's procedure. For the other models, detailed experimental setups are obtained through many trials. Related experimental settings are shown in the Appendix.

To further demonstrate the effectiveness of the proposed adaptive rescaled lncosh loss, the adaptive rescaled lncosh loss is replaced with Huber's loss function in RARLNN and noted as RARLNN-Huber in the experiments. Similarly, the l_1 loss and l_2 loss are applied to experiments as with the RARLNN-Huber, which are noted as RARLNN-l_1 and RARLNN-l_2, respectively.

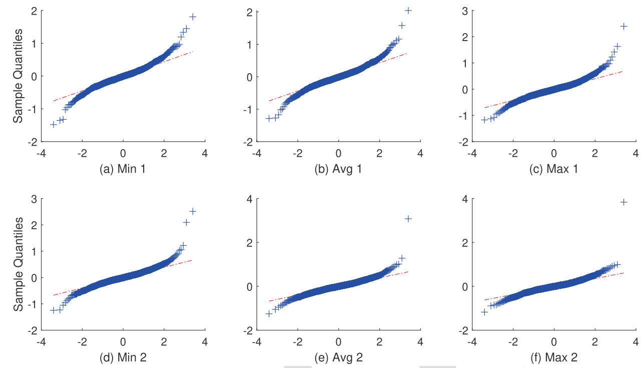


Fig. 4. QQ-plot for the six wind speed data sets. (a) Min 1. (b) Avg 1. (c) Max 1. (d) Min 2. (e) Avg 2. (f) Max 2.

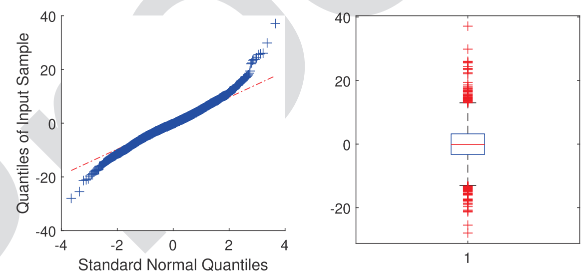


Fig. 5. QQ-plot and boxplot for the Vancouver wind speed data set.

After trials, the nodes of the hidden layer in both the neural network methods and ELM methods are set to 20, and the output neurons are set to 1. The loss function of all benchmark methods is traditional l_2 loss. For all benchmark methods, the number of input nodes is determined by partial autocorrelation function (PACF) figures for all data sets. The first a has a 0.95 confidence level and is set as the number of input nodes. In the experiments, the front a observations are used to predict the next data in 1-step predictions. In multistep predictions, the front a observations are used to predict the $(a+3)$th data and the $(a+5)$th data.

To evaluate the performance of PIs constructed by the proposed PI construction method, LUBE [16] is implemented in the experiments as a contrasting approach. To keep the validity of the PIs and the fairness of the comparison between the proposed method and LUBE, the target coverage probability u is set to 0.99 in the first group of experiments and 0.9 in the second group of experiments. The nodes of the hidden layer in LUBE are set to 20, which is the same as that in the proposed model. More details of the parameter settings can be found in the Appendix.

E. Results and Analysis for the First Group of Data Sets

In this section, we present the forecasting performance of all benchmark models for the first group of data sets, as well as the PIs constructed by the proposed PI construction method and LUBE method in Tables II and IV, respectively. Then, the empirical analysis is given.

TABLE II
ERROR COMPARISON AMONG DIFFERENT FORECASTING MODELS

Panel A														
		5min ahead		15min ahead		25min ahead			5min ahead		15min ahead		25min ahead	
Models	Data	MAE	RMSE	MAE	RMSE	MAE	RMSE	Data	MAE	RMSE	MAE	RMSE	MAE	RMSE
ELM	Min 1	0.1991	0.2596	0.4691	0.6070	0.6608	0.8662	Min 2	**0.1585**	0.2136	0.3589	0.4685	0.4985	0.6261
OS-ELM-MC		0.1972	0.2585	0.4607	0.5970	0.6300	0.8398		**0.1585**	0.2135	0.3580	0.4663	**0.4917**	0.6201
LSTM		0.1953	0.2561	0.4733	0.6141	0.6796	0.9053		0.1721	0.2286	0.3869	0.5025	0.5373	0.6847
LSTM-MSNet-DS		0.2046	0.2644	0.5194	0.6806	0.7193	0.9391		0.1802	0.2329	0.3831	0.4922	0.5192	0.6633
LSTM-MSNet-SE		0.2066	0.2691	0.4962	0.6478	0.7149	0.9310		0.1833	0.2386	0.3834	0.4934	0.5193	0.6652
Informer		0.2310	0.3008	0.5586	0.7127	0.7141	0.9441		0.1828	0.2444	0.4089	0.5384	0.5687	0.7322
Proposed.		**0.1921**	**0.2538**	**0.4557**	**0.5925**	**0.6293**	**0.8381**		0.1586	**0.2133**	**0.3579**	**0.4656**	**0.4917**	**0.6198**
ELM	Avg 1	0.1963	0.2528	0.4686	0.6044	0.6628	0.8647	Avg 2	0.1545	0.2093	0.3585	0.4666	0.4962	0.6223
OS-ELM-MC		0.1954	0.252	0.4605	0.5940	0.6335	0.8397		0.1542	0.2089	0.3570	0.4640	**0.4894**	0.6150
LSTM		0.1917	0.2498	0.4726	0.6103	0.6791	0.9049		0.1686	0.2243	0.3836	0.4997	0.5261	0.6714
LSTM-MSNet-DS		0.1990	0.2565	0.5233	0.6854	0.7039	0.9175		0.1825	0.2359	0.3826	0.4930	0.5191	0.6630
LSTM-MSNet-SE		0.2010	0.2615	0.5015	0.6546	0.7048	0.9252		0.1798	0.2338	0.3788	0.4878	0.5145	0.6584
Informer		0.2269	0.2972	0.5550	0.7095	0.7162	0.9478		0.1804	0.2420	0.4133	0.5471	0.5690	0.7339
Proposed.		**0.1915**	**0.2483**	**0.4566**	**0.5900**	**0.6330**	**0.8385**		**0.1538**	**0.2087**	**0.3552**	**0.4627**	0.4896	**0.6147**
ELM	Max 1	0.1926	0.2536	0.4641	0.6000	0.6567	0.8586	Max 2	0.1477	0.2029	0.3576	0.4601	0.4891	0.6137
OS-ELM-MC		0.1907	0.2517	0.4539	0.5886	0.6297	0.8348		**0.1473**	**0.2026**	0.3549	**0.4568**	**0.4832**	**0.6068**
LSTM		0.1922	0.2516	0.4737	0.6145	0.6748	0.9050		0.1635	0.2192	0.3819	0.4941	0.5240	0.6666
LSTM-MSNet-DS		0.1969	0.2555	0.5122	0.6708	0.6765	0.8935		0.1716	0.2263	0.3752	0.4819	0.5096	0.6505
LSTM-MSNet-SE		0.2003	0.2614	0.4863	0.6355	0.6794	0.8965		0.1797	0.2353	0.3834	0.4932	0.5101	0.6530
Informer		0.2264	0.2976	0.5356	0.6889	0.6964	0.9324		0.1784	0.2421	0.4033	0.5097	0.5651	0.7296
Proposed.		**0.1871**	**0.2481**	**0.4498**	**0.5850**	**0.6290**	**0.8334**		0.1474	**0.2026**	**0.3527**	0.4574	0.4837	0.6075

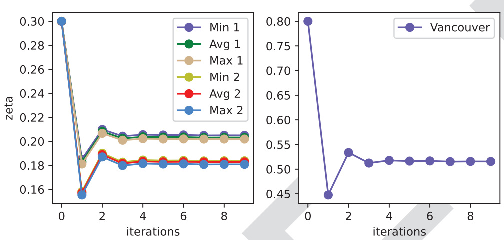


Fig. 6. Convergence of ζ versus iterations.

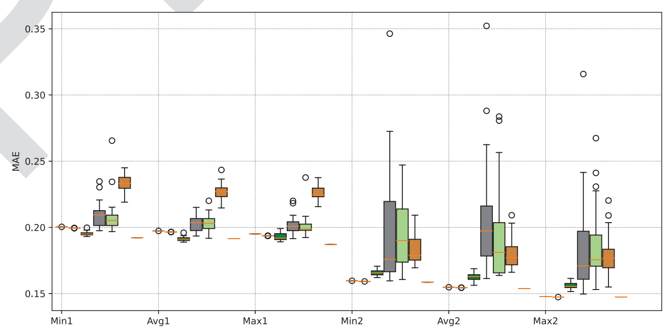


Fig. 7. Boxplots of all the models in the first group of wind speed data sets (5-min ahead). The 12 boxes for each data set correspond to the models in sequence: ELM, OS-ELM-MC, LSTM, LSTM-MSNet-DS, LSTM-MSNet-SE, Informer, and the proposed robust re-scaled Lncosh neural network.

1) Comparison Between Models Using Robust Loss Functions and Those Without: Because our wind speed data sets contain numerous outliers, we mainly focus on the robustness of the different models. To comprehensively study the performance of these models, we compare the proposed RARLNN with several popular prediction models (ELM, OS-ELM-MC, LSTM, LSTM-MSNet-DS, LSTM-MSNet-SE, and Informer.) on two groups of wind-speed data sets. Table II shows the error measures of all benchmark models.

For all six data sets, our proposed model performs considerably better than all models. In the 5-min ahead experiments, our proposed model outperforms other benchmark methods on MAE by up to 18.4%, and then by up to 16.9% on RMSE. In addition, we plot the convergence curves of ζ for the first six data sets on the left in Fig. 6. These curves demonstrate the convergence of hyper-parameter ζ because the weights $\hat{\beta}$ are optimized based on ζ. Therefore, the convergence of β can be demonstrated.

Because the proposed RARLNN performs relatively better among these models, we conclude that the models using the adaptive rescaled lncosh loss function can effectively reduce the influence of outliers in the wind speed data sets. Compared with classic models like ELM, our proposed model guarantees appreciative robustness and better forecasting accuracy. Especially compared with the l_2 loss in ELM, the better performance of the proposed model can prove the robustness of the adaptive rescaled lncosh loss.

2) Comparison Among All Benchmark Models: We first focus on 5-min ahead experiments. From the MAE values in Fig. 7 and Table II, the performance gap of all models is not very large. The OS-ELM-MC performs better than ELM in maximizing correntropy criteria. The reasons for the relatively worse performance of LSTM-related models (LSTM, LSTM-MSNet-DS, and LSTM-MSNet-SE) are that the basic LSTM model can capture long-term information in a time series. However, the best length to look back for LSTM-related models is 6, implying the earlier time has little correlation with the predicted time. Meanwhile, the l_2 loss cannot appropriately handle outliers. Thus, these LSTM-related models cannot achieve better performance than classic models. Next, we focus on the 15 and 25-min ahead experiments in Table II. An obvious conclusion is that the forecasting errors become larger with the number of predicted steps increasing, which is mainly due to the high randomness and complex noise of the wind speed series.

TABLE III
COMPARISON OF l_2, l_1, HUBER'S LOSS, AND THE ADAPTIVE RESCALED LNCOSH LOSS IN TERMS OF THE FIRST GROUP OF DATA SET

	RARLNN					Proposed
	l_2	l_1	Huber	Llncosh	lncosh ($\zeta = 1$)	
Min 1	0.2004	0.1954	0.1996	0.1946	0.1998	**0.1921**
Avg 1	0.1974	0.1917	0.1956	0.1921	0.1968	**0.1915**
Max 1	0.1951	0.1925	0.1927	0.1878	0.1927	**0.1871**
Min 2	0.1597	0.1597	0.1586	**0.1584**	0.1594	0.1586
Avg 2	0.1548	0.1545	0.1542	0.1541	0.1546	**0.1538**
Max 2	0.1478	**0.1473**	0.1477	**0.1473**	0.1475	0.1474
Mean	0.1759	0.1735	0.1747	0.1724	0.1754	**0.1718**

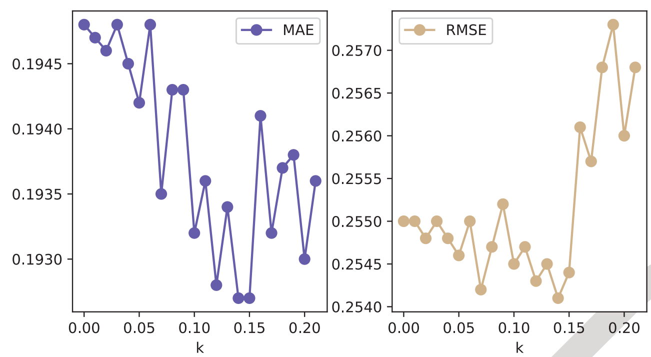


Fig. 8. Changes of MAE and RMSE criterion with the increase of k in (13) in terms of 5-min ahead predictions for Min1 data set.

3) Ablation Study: To compare the effects of our proposed adaptive rescaled lncosh loss function more fairly, we introduce l_1 loss, l_2 loss, Huber's loss, and a novel Llncosh method [37] into our proposed model, respectively. The detailed results are shown in Table III. We first compare the performance between the proposed adaptive rescaled lncosh loss and three classic loss functions (l_2 loss, l_1 loss, and Huber's loss). We can find that the performance of the adaptive rescaled lncosh loss (mean MAE: 0.1718) is stably better than the performance of the three classic loss functions (mean MAE: 0.1759, 0.1735, and 0.1747). This can demonstrate the superiority of the proposed adaptive rescaled lncosh loss. Then we make comparisons between the traditional lncosh loss, whose hyper-parameter ζ is selected as a constant 1, and the proposed adaptive rescaled lncosh loss. According to the results of the last two columns in Table III, we can find the performance of the adaptive rescaled lncosh (mean MAE: 0.1718) is stably better than the traditional lncosh (mean MAE: 0.1754). This can prove the ability of the designed "working" likelihood function of the lncosh function in fitting unknown noise distribution in wind speed series.

In addition, we like to illustrate the effects of adding random numbers conforming to the lncosh distribution and the ability to fit the lncosh distribution of the proposed adaptive rescaled lncosh loss. As shown in Fig. 8, we can find that the values of MAE and RMSE decrease and then rises with the increase of parameter k in (13), this suggests that adding moderate noise conforming to lncosh distribution can improve the forecasting performance and can also demonstrate the ability of fitting noise conforming lncosh distribution. Meanwhile, we compare another optimization method Llncosh for the hyper-parameter of lncosh loss [37]. The final performance ([37]: 0.1724, proposed: 0.1718) suggests the better performance of the proposed adaptive rescaled lncosh loss.

TABLE IV
COMPARISONS BETWEEN THE PROPOSED PI CONSTRUCTION METHOD AND LUBE METHOD AQ2

Step	Data	LUBE			The Proposed method		
		PICP(%)	NMPIW	CWC	PICP(%)	NMPIW	CWC
5 min ahead	Min 1	100	0.63	1.14	99.8	0.33	0.60
	Avg 1	100	0.65	1.18	99.6	0.33	0.63
	Max 1	100	0.56	1.01	99.8	0.35	0.65
	Min 2	100	0.60	1.08	100	0.33	0.60
	Avg 2	100	0.60	1.09	99.8	0.32	0.59
	Max 2	100	0.59	1.08	99.6	0.29	0.55
15 min ahead	Min 1	100	1.17	2.13	100	0.94	1.71
	Avg 1	99.80	1.11	2.06	100	0.95	1.72
	Max 1	100	1.16	2.10	100	0.96	1.74
	Min 2	100	0.89	1.63	100	0.78	1.42
	Avg 2	100	0.84	1.53	99.8	0.71	1.32
	Max 2	100	0.80	1.64	100	0.7	1.29
25 min ahead	Min 1	100	1.33	2.42	100	1.38	2.52
	Avg 1	100	1.37	2.49	100	1.38	2.50
	Max 1	100	1.44	2.61	100	1.39	2.53
	Min 2	100	1.15	2,09	100	1.11	2.02
	Avg 2	99.80	1.05	1.95	100	1.15	2.09
	Max 2	100	1.33	2.41	100	1.13	2.06

4) Comparisons Among the PIs Constructed by LUBE and the Proposed PI Method: The experimental results of the average criteria are tabulated in Table IV. The confidence level u is selected as 99% and η is selected as 20 in advance, aiming to obtain PIs whose PICP is larger than 0.99.

According to Table IV, the PICP of the PIs constructed by the proposed method and the LUBE can exceed 99%, and most of them can reach up to 100%. This suggests that the LUBE method and the proposed method can satisfy the requirement of the confidence level. As for the NMPIW criterion, the averages of the proposed method in 5, 15, and 25 min ahead experiments are 0.33, 0.84, and 1.26, respectively. However, the corresponding results of the LUBE method are 0.61, 1.00, and 1.28, respectively. This suggests that the bounds of the PIs constructed by the proposed method are smaller than those constructed by the LUBE method, especially in 5 and 15 min ahead experiments.

Then, we further analyze the comprehensive performance of the two methods in the aspect of the CWC criterion. As shown in Table IV, the averages of our proposed method in the 5, 15, and 25 min ahead experiments are 0.60, 1.53, and 2.29, respectively. The corresponding results of LUBE are 1.10, 1.85, and 2.33, respectively. This indicates that the comprehensive performance of our proposed method is also better than that of LUBE. Comprehensively considering the three criteria (PICP, NMPIW, and CWC), we hold the idea that LUBE may have relatively better performance with forecasting steps increasing because it can directly generate bounds without point predictions. However, our PI construction method can construct trustworthy PIs with relatively small boundaries when forecasting steps are not particularly large. This is mainly because the PI constructed by our method is highly related to the accuracy of point prediction. Figs. 9 and 10 also display that the average range of PIs constructed by LUBE has a poor performance, which supports our hypothesis.

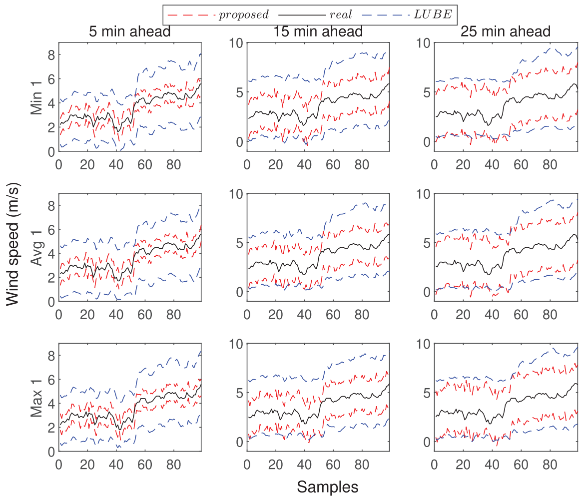


Fig. 9. Comparison of PIs between the proposed PI construction method and LUBE method for the first three wind data sets from North China. Each wind speed data set covers from 11:35, 4 June 2019, to 16:25, 9 June 2019, with a total of 1498 samples. Blue dots and red dots are the PIs constructed by LUBE and the proposed method, respectively. Black dots are the test set samples.

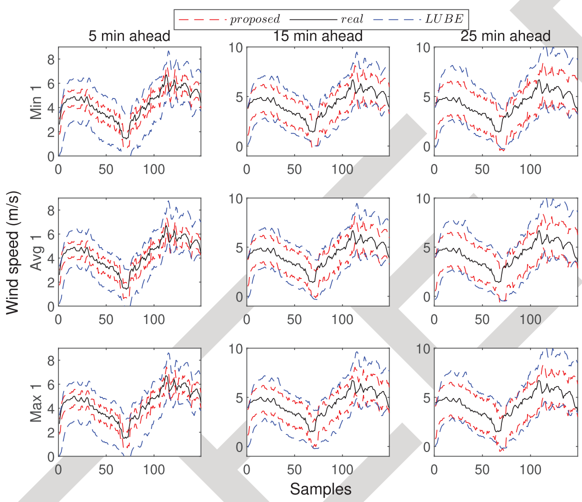


Fig. 10. Comparison of the PIs between the proposed PI construction method and LUBE method for the second three wind data sets from North China. Each wind speed data set covers from 17:30, 5 July 2019, to 22:15, 10 July 2019, with a total of 1498 samples. Blue dots and red dots are the PIs constructed by LUBE and the proposed method, respectively. Black dots are the test set samples.

Therefore, our PI construction method, which is based on the proposed RARLNN, can construct a PI with good performance for wind-speed forecasting.

F. Results and Analysis for the Second Group of Data Sets

In this section, we present the experimental results of all benchmark models for the second group of the data set, including the PIs constructed by the proposed method and the LUBE method. In this group of experiments, the confidence level u is selected as 0.9 to obtain PIs with a 0.9 confidence level. Detailed results are listed in Tables V and VII.

To begin with, the convergence of the hyper-parameter ζ is shown in the right of Fig. 6, which can also be found in the

TABLE V
Error Comparison of the Second Group of the Data Set

Model	1 hour ahead		3 hours ahead		5 hours ahead	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
ELM	3.5035	4.7732	4.6872	5.9894	5.3202	6.5930
OS-ELM-MC	3.4570	4.7528	4.5968	5.9289	**5.1358**	6.4469
LSTM	3.4788	4.8461	4,5601	5.9391	5.2100	6.4817
LSTM-MSNet-DS	3.6000	4.9077	4.5688	6.0046	5.1795	6.6277
LSTM-MSNet-SE	3.6022	4.9177	**4.5569**	6.0000	5.1818	6.6325
Informer	3.4771	4.7439	4.5724	5.9390	5.1432	**6.3623**
Proposed.	**3.4498**	**4.7329**	4.5668	**5.9179**	5.1389	6.4269

TABLE VI
Comparison of l_2, l_1, Huber's Loss, and the Adaptive Rescaled Lncosh Loss in Terms of the Vancouver Data Set

Criterion	RARLNN					Proposed
	l_2	l_1	Huber	Llncosh	lncosh ($\zeta = 1$)	
MAE	3.4935	3.4756	3.4918	3.4639	3.4864	**3.4498**

TABLE VII
Comparisons Between the Proposed PI Construction Method and LUBE Method

Step	LUBE			The Proposed method		
	PICP(%)	NMPIW	CWC	PICP(%)	NMPIW	CWC
1 hour ahead	96.83	0.38	0.39	93.92	0.32	0.36
3 hours ahead	96.29	0.46	0.48	93.91	0.40	0.46
5 hours ahead	97.08	0.49	0.51	95.34	0.46	0.49

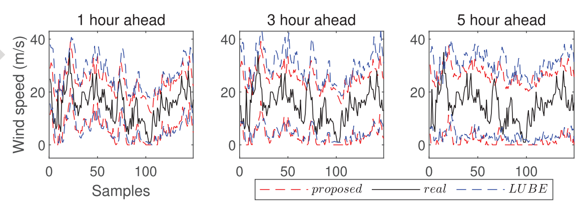


Fig. 11. Comparison of the PIs between the proposed PI construction method and LUBE method for the Vancouver wind data set. Blue dots and red dots are the PIs constructed by LUBE and the proposed method, respectively. Black dots are the test set samples. The confidence level is set to 0.9.

experiments for the first six data sets. In the 1, 3, and 5 h ahead experiments, the proposed model can achieve best forecasting performance than other benchmark models (OS-ELM-MC, LSTM, LSTM-MSNet-DS, LSTM-MSNet-SE, informer). This phenomenon proves the conclusion in Section IV-E that the proposed adaptive rescaled lncosh loss can appropriately reduce the effects of outliers and that the proposed RARLNN can achieve accurate predictions when facing time series with unknown prior information. The better overall performance of the proposed model suggests that our models are good at obtaining short-term predictions. This finding is consistent with the conclusion in Section IV-E. As for the constructed PIs, the proposed method performs better than LUBE based on the values of the NMPIW and CWC criteria in Table VII. Specifically, LUBE performs slightly better than our method in terms of PICP, and these two methods can both achieve the confidence level u. However, the boundary ranges of the PIs constructed by the LUBE method are larger than that of our method. Detailed PIs constructed by the two methods are shown in Fig. 11. As a result, our method can perform better than LUBE in terms of comprehensive CWC criterion.

TABLE VIII
TABLE OF HYPERPARAMETERS OF BENCHMARK MODELS

Models	Parameters	Min 1	Avg 1	Max 1	Min 2	Avg 2	Max 2	Vancouver
ELM	(a, h)[1]	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)
OS-ELM-MC	(a, h)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)
	(λ_0, A_0, h_0)	(8,10,1)	(8,10,1)	(8,10,1)	(8,10,1)	(8,10,1)	(8,10,1)	(8,10,1)
LSTM	(a, h, batch)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)
LSTM-MSNet-DS	(a, h, batch)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)
LSTM-MSNet-SE	(a, h, batch)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)	(6,20,32)
Informer	(seq_len,label_len, batch)	(6,3,32)	(6,3,32)	(6,3,32)	(6,3,32)	(6,3,32)	(6,3,32)	(6,3,32)
	(factor,e_layers,d_layers)	(5,2,1)	(5,2,1)	(5,2,1)	(5,2,1)	(5,2,1)	(5,2,1)	(5,2,1)
Proposed.	(a,h)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)	(6,20)

[1] The a and h represent the input length and the number of nodes in the hidden layer.

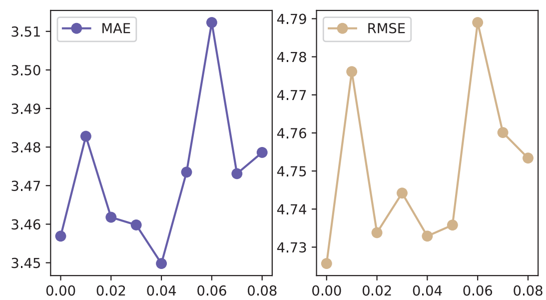


Fig. 12. Changes of MAE and RMSE criterion with the increase of k in (13) in terms of 1-h ahead predictions for Vancouver data set.

As with the first set of experiments, we conduct ablation studies for several classic loss functions (l_2 loss, l_1 loss, Huber's loss, and traditional lncosh ($\zeta = 1$) loss) and a novel optimization method named Llncosh for lncosh loss. As shown in Table VI, the proposed adaptive rescaled lncosh loss achieves the best performance, and the novel Llncosh has the second-best performance. This suggests the ability of the proposed adaptive rescaled lncosh loss in fitting complex unknown noise distribution. The Llncosh has a similar but slightly worse effect than the proposed adaptive rescaled lncosh loss in this group of experiments. This phenomenon is consistent with that in the first group of experiments. In addition, the MAE values of the proposed RARLNN with different random numbers are shown in Fig. 12. We can find that the performance of the proposed RARLNN decreases first and then increases with the increasing of parameter k in (13). The decrease represents the effect of adding random numbers conforming to lncosh loss to fit the noise distribution; the increase represents that the prediction results of the model will be affected by the increase of random numbers.

In summary, the main points of the specific analysis are included in the following.

1) The empirical analysis suggests that the proposed adaptive rescaled lncosh loss can reduce the drawback of outliers. The proposed adaptive rescaled lncosh loss function shows more stable and accurate performance than l_2, l_1, and Huber's loss function.
2) When encountering a data set whose prior knowledge is difficult to obtain, our proposed adaptive rescaled lncosh loss function can neatly approach the real noise distribution because it can approach the distributions of several common loss functions (l_2, l_1, and Huber's loss).
3) Compared with the common models, the proposed RARLNN has better robustness on complex time series because it can restrain outliers and approach unknown noise distribution in unknown data sets.
4) Compared with the results of the LUBE method, the empirical analysis indicates that the proposed PI construction method can construct PIs that are more accurate and reliable.

V. CONCLUSION

To obtain highly accurate predictions for time series with outliers, a novel RARLNN with an adaptive rescaled lncosh loss has been proposed. Specifically, an adaptive rescaled lncosh loss function has been proposed to approximate the unknown noise distribution and reduce the influence of outliers in complex time series. Based on the RARLNN, a novel PI construction method has been proposed to describe the prediction results at a specific confidence level. The experimental results show that this approach can construct PIs with high quality when compared with the traditional approach.

The adaptive rescaled lncosh loss function may be improved to integrate other distributions, not limited to a normal distribution and Laplacian distribution in the adaptive rescaled lncosh loss. At the same time, the problem of distribution shifting between the training set and the test set (or offline data and online data) can be considered and solved in the future. In future work, an alternative to the "working" likelihood approach is the Bayesian approach, which can integrate the hyper-parameter in the robust prediction. This Bayesian approach also has the advantage of resultant data-dependent tuning parameters. For the Bayesian approach, the key issue is to build a surrogate model of the objective function.

APPENDIX

ABLATION EXPERIMENTS

In the current work, the authors have considered six benchmark algorithms in total, including ELM, OS-ELM-MC, LSTM, LSTM-MSNet-SE, LSTM-MSNet-DS, and Informer. The input nodes of these models are determined by PACF, as shown in Table VIII. The parameter settings of the informer are according to our empirical trials. The hyper-parameter of Huber's loss is set to 1.345 according to [12].

REFERENCES

[1] M. Khodayar and J. Wang, “Spatio-temporal graph deep neural network for short-term wind speed forecasting,” *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.

[2] C.-Y. Zhang, C. P. Chen, M. Gan, and L. Chen, “Predictive deep Boltzmann machine for multiperiod wind speed forecasting,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1416–1425, Oct. 2015.

[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, Jan. 2022.

[4] S. Sun, S. Wang, Y. Wei, and G. Zhang, “A clustering-based nonlinear ensemble approach for exchange rates forecasting,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 6, pp. 2284–2292, Jun. 2020.

[5] Y. Yang, Q. M. J. Wu, Y. Wang, K. M. Zeeshan, X. Lin, and X. Yuan, “Data partition learning with multiple extreme learning machines,” *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1463–1475, Aug. 2015.

[6] Y. Yang and Q. J. Wu, “Extreme learning machine with subnetwork hidden nodes for regression and classification,” *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2885–2898, Dec. 2016.

[7] T. Li, Y. Pan, K. Tong, C. E. Ventura, and C. W. de Silva, “Attention-based sequence-to-sequence learning for online structural response forecasting under seismic excitation,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2184–2200, Apr. 2022.

[8] J. Li et al., “A novel hybrid short-term load forecasting method of smart grid using MLR and LSTM neural network,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2443–2452, Apr. 2021.

[9] J. Zhang, K. Zhang, Y. An, H. Luo, and S. Yin, “An integrated multitasking intelligent bearing fault diagnosis scheme based on representation learning under imbalanced sample condition,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 6, 2023, doi: 10.1109/TNNLS.2022.3232147.

[10] K. Bandara, C. Bergmeir, and H. Hewamalage, “LSTM-MSNET: Leveraging forecasts on sets of related time series with multiple seasonal patterns,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1586–1599, Apr. 2021.

[11] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1997, pp. 155–161.

[12] P. J. Huber, “Robust regression: Asymptotics, conjectures and monte carlo,” *Ann. Stat.*, vol. 1, no. 5, pp. 799–821, 1973.

[13] A. Esmaeili and F. Marvasti, “A novel approach to quantized matrix completion using Huber loss measure,” *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 337–341, Feb. 2019.

[14] O. Karal, “Maximum likelihood optimal and robust support vector regression with lncosh loss function,” *Neural Netw.*, vol. 94, pp. 1–12, Oct. 2017.

[15] K. Ning, M. Liu, M. Dong, C. Wu, and Z. Wu, “Two efficient twin elm methods with prediction interval,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2058–2071, Sep. 2015.

[16] 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 Netw.*, vol. 22, no. 3, pp. 337–346, Mar. 2011.

[17] C. Li, G. Tang, X. Xue, A. Saeed, and X. Hu, “Short-term wind speed interval prediction based on ensemble GRU model,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1370–1380, Jul. 2020.

[18] M. Moness and A. M. Moustafa, “A survey of cyber-physical advances and challenges of wind energy conversion systems: Prospects for Internet of energy,” *IEEE Internet Things J.*, vol. 3, no. 2, pp. 134–145, Apr. 2016.

[19] X. He, X. Fang, and J. Yu, “Distributed energy management strategy for reaching cost-driven optimal operation integrated with wind forecasting in multimicrogrids system,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 8, pp. 1643–1651, Aug. 2019.

[20] Q. Xu et al., “A short-term wind power forecasting approach with adjustment of numerical weather prediction input by data mining,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1283–1291, Oct. 2015.

[21] K. Yunus, T. Thiringer, and P. Chen, “ARIMA-based frequency-decomposed modeling of wind speed time series,” *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2546–2556, Jul. 2016.

[22] M. Khodayar, J. Wang, and M. Manthouri, “Interval deep generative neural network for wind speed forecasting,” *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3974–3989, Jul. 2019.

[23] M.-R. Chen, G.-Q. Zeng, K.-D. Lu, and J. Weng, “A two-layer nonlinear combination method for short-term wind speed prediction based on ELM, ENN, and LSTM,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6997–7010, Aug. 2019.

[24] J. Zhang, X. Li, J. Tian, Y. Jiang, H. Luo, and S. Yin, “A variational local weighted deep sub-domain adaptation network for remaining useful life prediction facing cross-domain condition,” *Rel. Eng. Syst. Safety*, vol. 231, Mar. 2023, Art. no. 108986.

[25] J. Zhang, X. Li, J. Tian, H. Luo, and S. Yin, “An integrated multi-head dual sparse self-attention network for remaining useful life prediction,” *Rel. Eng. Syst. Safety*, vol. 233, May 2023, Art. no. 109096.

[26] W. Wang, C. Shi, W. Wang, L. Dang, S. Wang, and S. Duan, “Online sequential extreme learning machine algorithms based on maximum correntropy citerion,” in *Proc. 20th Int. Conf. Inf. Fusion (Fusion)*, 2017, pp. 1–7.

[27] R. J. Carroll and D. Ruppert, “Robust estimation in heteroscedastic linear models,” *Ann. Stat.*, vol. 10, no. 2, pp. 429–441, 1982.

[28] J. Wu and Y.-G. Wang, “A working likelihood approach to support vector regression with a data-driven insensitivity parameter,” *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 3, pp. 929–945, 2023.

[29] L. Fu, Y.-G. Wang, and F. Cai, “A working likelihood approach for robust regression,” *Stat. Methods Med. Res.*, vol. 29, no. 12, pp. 3641–3652, 2020.

[30] Y.-G. Wang and Y. Zhao, “A modified pseudolikelihood approach for analysis of longitudinal data,” *Biometrics*, vol. 63, no. 3, pp. 681–689, 2007.

[31] Y.-G. Wang, J. Wu, Z.-H. Hu, and G. J. McLachlan, “A new algorithm for support vector regression with automatic selection of hyperparameters,” *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 108989.

[32] J. Wu and Y.-G. Wang, “Iterative learning in support vector regression with heterogeneous variances,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 2, pp. 513–522, Apr. 2023.

[33] X. Lu, H. Zou, H. Zhou, L. Xie, and G.-B. Huang, “Robust extreme learning machine with its application to indoor positioning,” *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 194–205, Jan. 2016.

[34] P. Horata, S. Chiewchanwattana, and K. Sunat, “Robust extreme learning machine,” *Neurocomputing*, vol. 102, pp. 31–44, Feb. 2013.

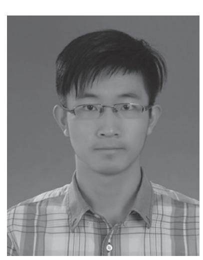
[35] R. J. Hyndman and Y. Khandakar, “Automatic time series forecasting: The forecast package for R,” *J. Stat. Softw.*, vol. 27, no. 1, pp. 1–22, 2008.

[36] H. Zhou et al., “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 11106–11115.

[37] C. Liu and M. Jiang, “Robust adaptive filter with lncosh cost,” *Signal Process.*, vol. 168, Mar. 2020, Art. no. 107348.

Yang Yang (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in automatic control from Dalian Maritime University, Dalian, China, in 2008, 2010, and 2013, respectively.

From 2018 to 2019, he held a Visiting Research Fellow position with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, Australia. He is currently an Associate Professor with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include nonlinear control theory and intelligent control.



Hu Zhou received the bachelor’s degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2020, where he is currently pursuing the M.E. degree in electronic information.

He is interested in energy forecasting for smart grids.

Jinran Wu received the Ph.D. degree in statistics from the Queensland University of Technology (QUT), Brisbane, QLD, Australia, in 2022.

He is currently a Research Fellow with the Institute for Learning Sciences and Teacher Education, Australian Catholic University, Brisbane. Prior to that, he was an Associate Lecturer with the School of Mathematical Sciences, QUT. He is interested in statistical machine learning and optimizations with applications.

Zhe Ding received the Master of Research degree in information technology from the Queensland University of Technology, Brisbane, QLD, Australia, in 2017, where he is currently pursuing the Ph.D. degree with the School of Computer Science.

His research interests include big data computing, resource virtualization, cloud computing, and data center optimization.

Yu-Chu Tian (Senior Member, IEEE) received the first Ph.D. degree in industrial automation from Zhejiang University, Hangzhou, China, in 1993, and the second Ph.D. degree in computer and software engineering from the University of Sydney, Sydney, NSW, Australia, in 2019.

He is currently a Professor of Computer Science with the School of Computer Science, Queensland University of Technology, Brisbane, QLD, Australia. His research interests include big data computing, cloud computing, computer networks, smart grid communications and control, networked control systems, cyber–physical security, and optimization and machine learning.

Dong Yue (Fellow, IEEE) received the Ph.D. degree in automation engineering from the South China University of Technology, Guangzhou, China, in 1995.

He is currently a Professor and the Dean of the Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, and a Changjiang Professor with the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. His research interests include the analysis and synthesis of networked control systems, multiagent systems, optimal control of power systems, and Internet of Things.

You-Gan Wang received the Ph.D. degree in statistics from Oxford University, Oxford, U.K., in 1991.

He is currently a Professor of Data Analytics with the Australian Catholic University, Brisbane, QLD, Australia. His research interests include developing machine learning algorithms and statistical methodology for analyzing longitudinal data and time series to solve important problems in education, social sciences, medical sciences, environmental research, and natural resource management.