Probabilistic Prediction of Solar Generation Based on Stacked Autoencoder and Lower Upper Bound Estimation Method

Cheng Pan*
Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Beijing 100190, P. R. China
pancheng2014@ia.ac.cn

Jie Tan
Institute of Automation, Chinese Academy of Sciences
Chinese Academy of Sciences
Beijing 100190, P. R. China
tan.jie@tom.com

***Abstract*—The lower upper bound estimation method is an important probabilistic prediction method and has been applied to the solar generation forecasting. However, when the input dimension of the lower upper bound estimation method is large, its performance will be seriously affected. To overcome this challenge, a novel probabilistic prediction of solar generation based on stacked autoencoder and lower upper bound estimation method is proposed. In this method, stacked autoencoder is first used to obtain highly compressed features, which are utilized as the input of the lower upper bound estimation method. Besides, to make the target value in the center of the prediction interval as much as possible, inspired by the idea of support vector machine, the mean squared error of prediction interval is introduced to the loss function, which keeps the target value as far as possible from the lower and upper bounds of the prediction interval. To verify the performance of the proposed method, a large number of experiments have been carried out on the freely available dataset. The results show that the proposed method has better forecasting performance.**

***Index Terms*—stacked autoencoder, lower upper bound estimation, solar generation forecasting, probabilistic prediction**

I. Introduction

Recently, in order to cope with global climate change, countries around the world actively carry out emission reduction actions [1], [2]. This has led to a significant increase in the percentage of photovoltaic (PV) in annual electricity consumption [3]. However, solar generation has randomness, volatility, and intermittence. Its large-scale integration into the grid will bring great challenges to the safe and stable operation of the grid [4]. Solar generation forecasting, as an important way to overcome these challenges, has become the focus of academia and industry.

Most previous works in solar forecasting have mainly focus on the deterministic prediction (point forecast) [5]–[7]. However, due to the influence of meteorological factors, solar generation has strong randomness. When the weather changes greatly, the deterministic prediction accuracy is poor. Probabilistic prediction can provide all possible solar generation conditions and corresponding probabilities at the next moment, which can provide more comprehensive prediction information. Therefore, the probabilistic prediction is more conducive to the safe and stable operation of power systems, which has attracted more and more attention. Quantile regression [8], quantile regression forests [9], Gaussian process [10], lower upper bound estimate (LUBE) [11], and other probabilistic prediction methods, have been proposed and applied in solar forecasting.

LUBE method is a probabilistic prediction method based on a neural network. Its output layer has two nodes, which are the upper bound of the prediction interval and the lower bound of the prediction interval. Ni et al. [12] proposed a novel ensemble approach based on extreme learning machines (ELM) and LUBE for short-term PV power forecasting. In [13], the LUBE method was used to quantify potential uncertainties of electrical demands and wind power generation. Because the objective function of the LUBE method is not differentiable, it is usually solved by evolutionary algorithm, such as particle optimization (PSO), ant colony algorithm (ACA), genetic algorithm (GA) and so on [14], [15]. Taking PSO algorithm as an example, when the dimension of particles is too high, the solution speed of the LUBE method will be reduced, and it may lead to a complex loss function and fall into a local optimum. Reducing the dimension of particles is an important way to improve the performance of the LUBE method.

Deep learning, a powerful machine learning method, has been widely used in computer vision [16], [17], natural language processing [18], [19], robotics [20], [21], and other fields, and achieved accepting results. As an efficient algorithm of deep learning, stacked autoencoder (SAE) is made up of several autoencoders (AEs). The basic idea of AE is to use neural networks for unsupervised learning, that is, samples are used as input and output of neural networks at the same time. In essence, it is hoped that it can obtain the representation of input samples. SAE uses multiple AEs to improve its ability to learn the representation of input samples, and has been successfully applied in wind speed prediction [22], image classification [23], drug-target interactions prediction [24], synthetic aperture radar target recognition [25] and other fields. In [26], the combination of Deep Belief Network (DBN),

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

AE, and LSTM shows the predictive advantage in predicting the energy output of 21 solar power plants compared to standard Multi-Layer Perceptron (MLP) and physical prediction models. However, SAE has not been combined with the LUBE method to conduct the probabilistic prediction of solar generation.

Following the motivation above, we propose a novel method of solar generation probabilistic prediction based on SAE and LUBE. In the proposed method, highly compressed features obtained by SAE are used as the input of the LUBE method, which can quickly and efficiently constructs the prediction interval of solar generation. Besides, the mean squared error of prediction interval is added to the loss function of the LUBE method to make the target value in the center of the prediction interval as much as possible.

The remainder of the paper is organized as follows. In Section II, the proposed method is described in detail. The experimental results and analysis are presented in Section III. Finally, the conclusion of the paper is summarized in Section IV.

II. Proposed method

The flowchart of the proposed method is shown in Fig. 1. In the proposed method, we first use a stacked autoencoder to transform the raw data to obtain highly compressed features. Then, highly compressed features are used as the input of the LUBE method to construct the prediction interval, and particle swarm optimization is used to find the optimal parameters of the LUBE method. In the following section, we will describe the main content of the proposed method in detail.

A. Autoencoder and Stacked Autoencoder

AE is a three-layer network model, which consists of the encoder and the decoder, as shown in Fig. 2. Suppose X is the input vector, h is the hidden vector, and Y is the output vector (i.e., reconstructed data). The encoder maps the input layer data X to a hidden vector h, which can be represented as follows:

$$h = f_\theta(W_i X + b_i), \tag{1}$$

where θ is the parameters of encoder, including W_i and b_i, W_i is the weight matrix between the hidden layer and the input layer, b_i is the bias vector of hidden layer.

The decoder maps the hidden vector h to the reconstruction vector y of the input vector X. This process can be defined as follows:

$$Y = g_{\theta'}(W_o h + b_o), \tag{2}$$

where $\theta^{'}$ is the parameters of decoder, including W_o and b_o, W_o is the weight matrix between the hidden layer and the output layer, b_o is the bias vector of output layer.

As defined in Eq (3), the learning task of AE is to find the optimal parameters to minimize reconstruction errors.

$$(\theta, \theta^{'}) = \arg\min_{(\theta,\theta')} L(X, Y), \tag{3}$$

where L is the loss function of AE, which can be represent as:

$$L(X, Y) = \frac{1}{2N}\sum_{i=1}^{N} ||X - Y||_2, \tag{4}$$

where N is the number of samples, $||\circ||_2$ is the *Euclidean norm*.

When the reconstruction error between input vector X and output vector Y is small enough, the hidden vector h can be regarded as the feature representation of the input vector X. Therefor, we can set the dimension of hidden layer less than that of input vector to achieve the purpose of feature extraction.

SAE is a typical deep learning model, which is composed of multiple AE. By increasing the hidden depth of SAE, the ability of feature extraction and training effect can be improved. The training process of SAE can be divided into two parts: pre-training and fine tuning. The pre-training process is to train the parameters of each AE, and fine tuning process is to train the parameters of the SAE.

B. Lower Upper Bound Estimation

LUBE based on neural network is an important and effective method to directly construct prediction intervals. As shown in Fig. 3, it has two outputs, i.e., upper bound and lower bound. The learning task of LUBE is to find the optimal parameters of neural network to minimize its loss function and obtain the acceptable forecasting performance. In the LUBE method, its loss function comprehensively consider reliability and sharpness.

In probabilistic prediction, reliability can be evaluated by the average coverage error (ACE), which can be defined as

$$ACE = PINC - PICP, \tag{5}$$

where PINC (prediction interval nominal confidence) represents the expected confidence level that target values lie in the prediction intervals, PICP (prediction interval coverage probability) gives the actual probability that target values will be covered by the predicted intervals. PICP can be defined as follows:

$$PICP = \frac{1}{N}\sum_{i=1}^{N} \epsilon_i, \tag{6}$$

where N is the number of samples, ϵ_i is a Boolean variable, which indicates the coverage behavior of prediction intervals. Mathematically, ϵ_i can be calculated as follows:

$$\epsilon_i = \begin{cases} 1 & \text{if } Y_i \in [L_i^{(\alpha)}, U_i^{(\alpha)}] \\ 0 & \text{if } Y_i \notin [L_i^{(\alpha)}, U_i^{(\alpha)}] \end{cases}, \tag{7}$$

where α is the nominal confidence level associated with prediction interval, $L_i^{(\alpha)}$ and $U_i^{(\alpha)}$ are the lower bound and upper bound of the prediction interval, respectively.

The Winkler score can be used to evaluate the sharpness of prediction interval. It can be defined as follows:

$$S_i^{(\alpha)} = \begin{cases} -2\alpha\vartheta_i^{(\alpha)} - 4(L_i^{(\alpha)} - Y_i), & \text{if } Y_i < L_i^{(\alpha)} \\ -2\alpha\vartheta_i^{(\alpha)}, & \text{if } Y_i \in \vartheta_i^{(\alpha)} \\ -2\alpha\vartheta_i^{(\alpha)} - 4(Y_i - U_i^{(\alpha)}), & \text{if } Y_i > U_i^{(\alpha)} \end{cases}, \tag{8}$$

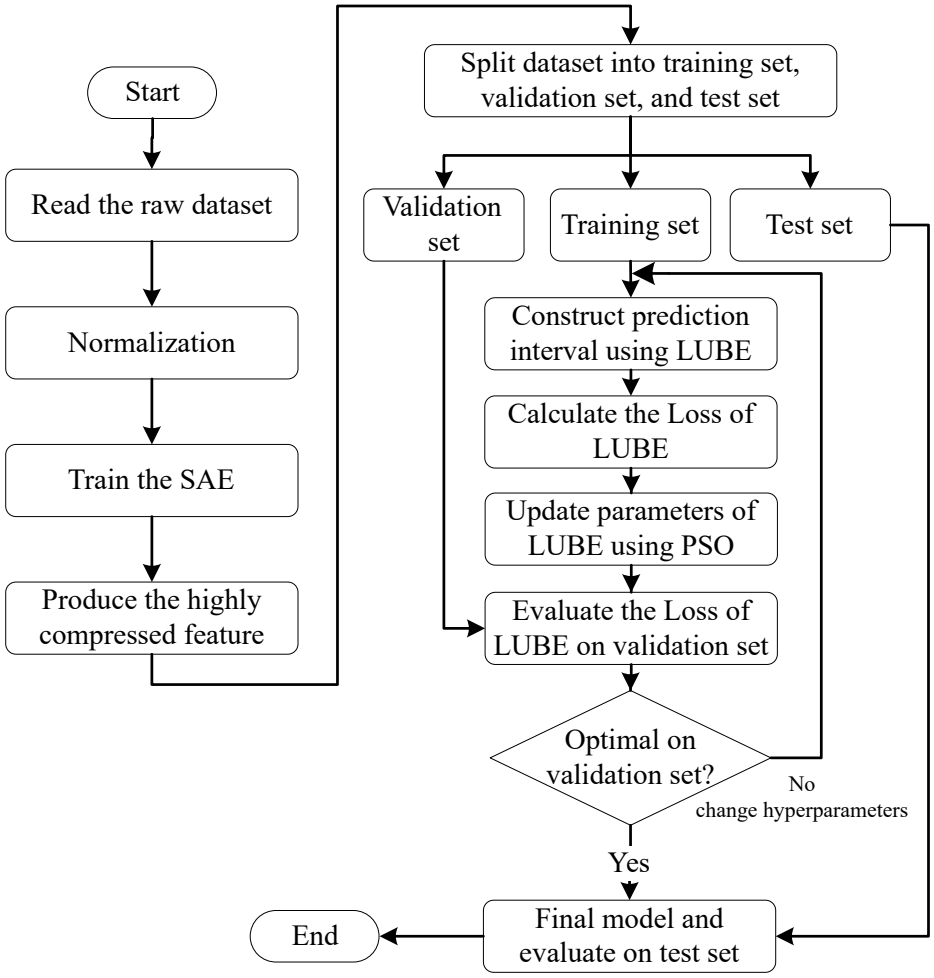

Fig. 1. Flowchart of the proposed method.

where $\vartheta_i^{(\alpha)}$ is the prediction interval width, as shown in Eq (9).

$$\vartheta_i^{(\alpha)} = U_i^{(\alpha)} - L_i^{(\alpha)}. \tag{9}$$

Then, the Winkler score on the whole dataset can be represented as

$$S^{(\alpha)} = \frac{1}{N}\sum_{i=1}^{N} S_i^{(\alpha)}. \tag{10}$$

In [15], the loss function combining ACE and Winkler score was defined as follows:

$$Loss = \gamma|ACE| + \lambda|S^{(\alpha)}|. \tag{11}$$

However, it may face a problem that the target value is not in the center of the prediction interval, when the loss function mentioned above is used to train the LUBE method. Thus, the target value that changes drastically is easily not included in the prediction interval. In order to avoid this problem, inspired by the idea of support vector machine (SVM), we introduce the mean squared error of prediction interval (PIMSE) to the loss function. This keeps the target value as far as possible from the lower and upper bounds of the prediction interval. Mathematically, PIMSE can be calculated as follows:

$$PIMSE = \frac{1}{N}\sum_{i=1}^{N}[(U_i^{(\alpha)} - Y_i)^2 + (L_i^{(\alpha)} - Y_i)^2]. \tag{12}$$

Finally, the loss function of LUBE can be derived as follows:

$$Loss = \gamma|ACE| + \lambda|S^{(\alpha)}| + \eta PIMSE, \tag{13}$$

where γ, λ, and η are the controlling weights.

From the form of the loss function of the LUBE method, it can be seen that the LUBE method can not be optimized by the error backpropagation algorithm. Therefore, the PSO

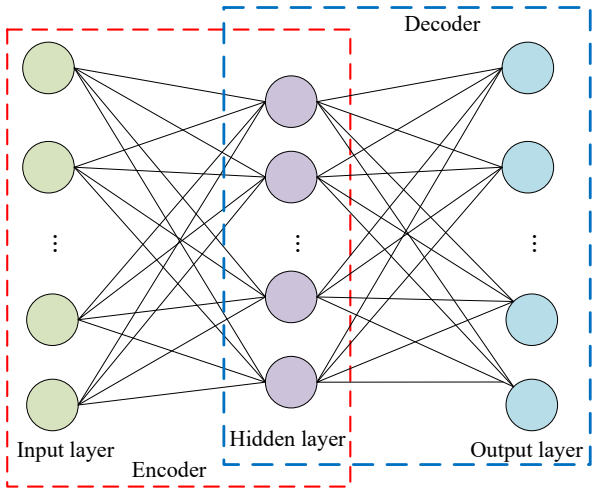

Fig. 2. Structure of autoencoder.

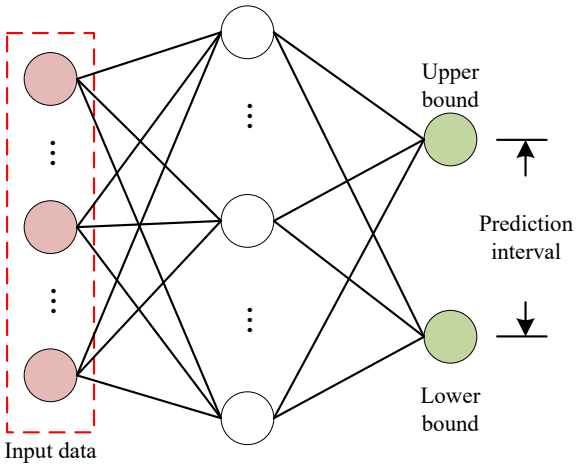

Fig. 3. illustration of LUBE method.

described in the next section is used to solve the optimal parameters of the LUBE method.

C. Particle swarm optimization

PSO was first proposed in [27]. It is a population-based optimization technique inspired by predatory behavior of birds. The basic core of PSO is that individual swarm members have establish social networks that can benefit from the discoveries and prior experience of other swarm members.

Suppose particles are located in D-dimension space, and the ith particle position is $X_i = (X_{i1}, X_{i2}, \cdots, X_{iD})$, ith particle velocity is $V_i = (V_{i1}, V_{i2}, \cdots, V_{iD})$. In addition, we define the optimal position of the ith particle as $pbest_i$, the optimal position of particle swarm as $gbest$. Then, the position and velocity of the particle will be updated by the equation defined as follows:

$$\begin{aligned} V_i &= wV_i + c_1 r_1 (pbest_i - X_i) + c_2 r_2 (gbest - X_i) \\ X_i &= X_i + V_i \end{aligned}, \quad (14)$$

where w is the inertia weight, r_1 and r_2 are random numbers between 0 and 1, c_1 and c_2 are the cognitive and social coefficient, respectively.

In our work, we set the position of particles as the parameters of the neural network in the LUBE method, including the weights and biases of the input layer to the hidden layer and the hidden layer to the output layer. At the same time, we set the optimization goal of the PSO (i.e., fitness function) as the loss function of LUBE method, defined as Eq (13).

III. Case Study

A. Data

The Desert Knowledge Australia Solar Centre (DKASC) [28] is an open platform for sharing high-quality solar related data and knowledge in the Northern Territory, Australia and beyond. The diverse arrays installed in DKASC are shown in Fig. 4. In our case, the dataset from site 31 TDG, DKASC is used to verify the performance of the proposed method. Its installation parameters are described in Table I. The provided dataset contains solar generation and weather information with a resolution of 5-min. The weather information includes six variables, namely, wind speed, weather temperature, weather relative humidity, global horizontal radiation, diffuse horizontal radiation, and wind direction. Due to the large number of outliers in the wind direction, this weather information is discard in our case. Here, the data from 1 April 2014 to 31 October 2015 are used in our work.

In this work, we aim to use the past sequence of solar generation and weather information to quantify the prediction interval of solar generation in the next 5-min. To that end, sliding window is used here to construct the dataset. The window length should be first determined. According to Fig. 5, we can find that the first four lags have larger partial correlation coefficients, and subsequent lags are less than 0.1. Thus, the window length is set to four. It means that we use the solar generation and weather information (include five weather variables) at time $t-3, t-2, t-1, t$ to forecast the prediction interval of solar generation at time $t+1$. Thus the input dimension of the proposed method is 24. Besides, there some missing data in some days, for convenience, we delete these days directly. Finally, we use the 443 days data from 1 April 2014 to 30 June 2015 as training set, the 61 days data from 1 July 2015 to 31 August 2015 as validation set, and the 61 days data from 1 September 2015 to 31 October 2015 as test set. Because the solar panels do not have power output during night, the solar generation at 5:30-19:00 is predicted in our case. Hence, we can obtain the size of training set is 71766 ($443 * 162$, 162 is the number of samples can be construct in one day), the size of validation set is 9882 ($61 * 162$), the size of test set is 9882 ($61 * 162$).

TABLE I
INSTALLATION PARAMETERS OF SITE NUMBER 31 TDG.

Parameter	Value
Array Rating	5kW
Panel Rating	250W
Number Of Panels	20
Panel Type	TDG T250M606
PV Technology	mono-Si
Array Structure	Fixed: Ground Mount
Array Area	$33.5m^2$
Inverter Size / Type	APS Micro-inverter YC500 (x 10)
Installation Completed	Wed, 7 Aug 2013
Array Tilt/Azimuth	Tilt = 20, Azi = 0 (Solar North)

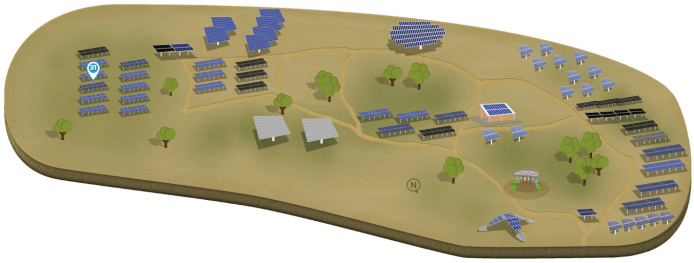

Fig. 4. The DKASC is located at the Desert Knowledge Precinct in Alice Springs, a town in the Northern Territory that enjoys one of the country's highest solar resources in an arid desert environment.

B. Experimental Results and Analysis

To verify the performance of the proposed method, a large number of experiments are carried out and analyzed in this section. In our case, all experiments are implemented with Python 3.7 on a standard i7 3.4-GHz computer.

1) Hyperparameter Setting: In the proposed method, the purpose of the SAE is to obtain highly compressed features. Therefore, we set the number of AEs in the SAE to 2, and set the number of nodes in the hidden layer of the first AE to 15, and the number of nodes in the second hidden layer to 4. That means the first AE compresses the feature with dimensions 24 to 15, and the second AE compresses the feature with dimensions 15 to 4. According to Fig. 6, we can find that pre-training loss and fine tuning loss converge at 600 iterations. Besides, we can observe that the MSE loss of the first AE is closer to 0, compared with the MSE loss of the second AE and the fine tune. It is evident that the feature compression of data inevitably brings losses.

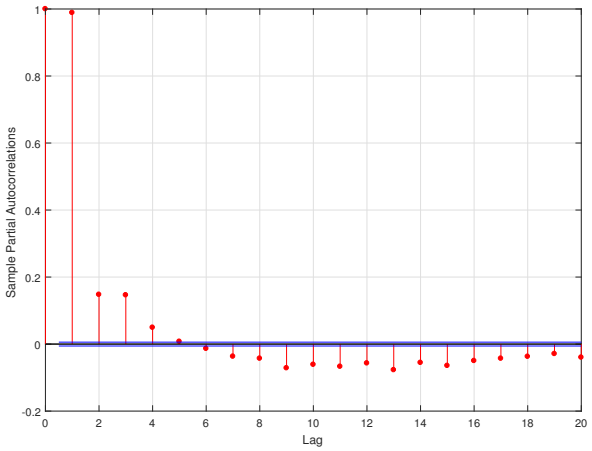

Fig. 5. Partial autocorrelation graph of the solar generation.

TABLE II
PARAMETERS USED IN THE EXPERIMENTS.

	Parameter	Numerical value
SAE	input size of SAE	24
	number of hidden layer	2
	number of nodes (1st hidden layer)	15
	number of nodes (2nd hidden layer)	4
	optimizer	Adam
	learning rate	0.001
LUBE	input size of LUBE	4
	number of hidden layer	1
	number of nodes (hidden layer)	3
	output size of LUBE	2
	controlling weight γ, λ, η	$1, 0.05, 0.05$
PSO	size of particle	23
	number of particles	60
	inertia weight w	0.5
	cognitive coefficient	2
	social coefficient	2
	number of iteration	300

Once obtain the highly compressed features, the PSO algorithm is adopted to find the optimal parameters of the LUBE method to construct the prediction interval. The forecasting performance of the LUBE method depends on the structure of the neural network. In this paper, the single hidden layer neural network is considered. In addition, the number of nodes in the hidden layer is set to 3. According to the data representation obtained by the SAE and the neural network structure of the LUBE method, there are 23 parameters to be trained in the LUBE method. Thus, the size of particle is set to 23. According to Eq (13), γ, λ, and η are three important hyperparameters of the LUBE method. To determine the proper hyperparameters, the grid search method on the validation set is adopted. After many experiments, the three hyperparameters, namely, γ, λ, and η, are set to 1, 0.05, and 0.05, respectively. As shown in Fig. 7, the LUBE loss of $gbest$ drops sharply at the beginning and simply converges in 50 iterations. After several iterations, the LUBE loss of $gbest$ makes little changes, which implies that the optimal parameters of the LUBE method are found. Finally, the typical hyperparameters of our case for SAE, LUBE, and PSO are summarized in Table II.

2) Method comparison: To verify the forecasting performance of the proposed method, we compare it with the persistence ensemble (PeEn). The PeEn is widely used as a referenced model in solar probabilistic forecasting, which assumes that the forecast error is a random normal distribution [12], [29], [30]. Table III shows the probabilistic prediction performance of the proposed method (denoted as SAE-LUBE) and the compared method in terms of PICP, PIMSE, Winkler score, and loss. For PICP, the proposed method can meet the

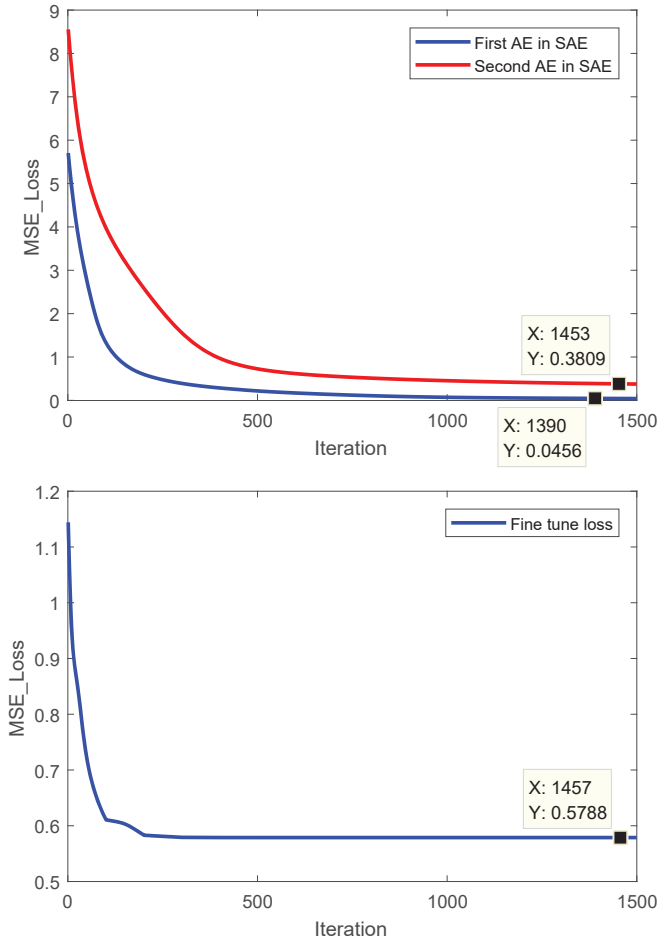

Fig. 6. Pre-training loss and fine tuning loss.

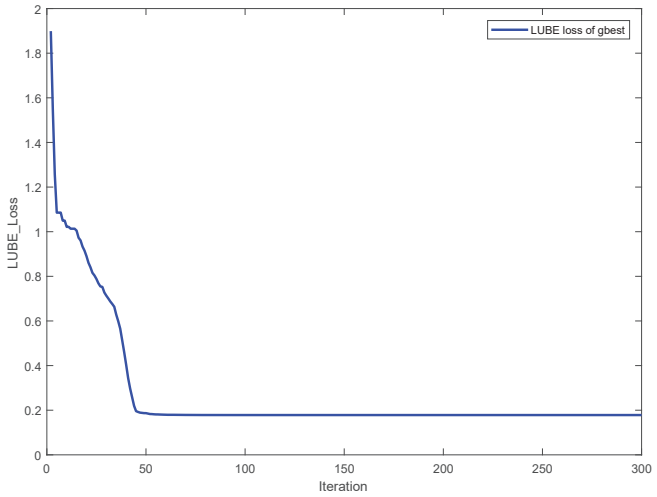

Fig. 7. LUBE loss of *gbest* particles during training process.

TABLE III
PROBABILISTIC PREDICTION PERFORMANCE AT DIFFERENT CONFIDENCE LEVELS.

PINC	Method	PICP	PIMSE	Winkler score	Loss
85%	PeEn	70.8%	16.13	-9.28	1.41
	SAE-LUBE	92.2%	4.79	-4.05	0.51
90%	PeEn	75.8%	16.17	-9.30	1.42
	SAE-LUBE	94.8%	0.91	-2.07	0.20
95%	PeEn	81.5%	16.26	-9.33	1.41
	SAE-LUBE	96.3%	1.03	-2.28	0.18

requirements of different confidence levels. For example, when the confidence level (i.e., PINC) is 90%, the PICP of our method is 94.8%, which meets the requirement of confidence level (i.e., PINC). However, the PICP of the PeEn is 75.8%, which does not meet the requirement of PINC. This shows that the proposed method has better reliability. In addition, at this confidence level, the Winkler score of the proposed method is -2.07, which is better than the compared method (Winkler score: -9.30). This indicates that the proposed method has better sharpness. Combing the reliability and sharpness, we can conclude that the proposed method has better forecasting performance. By analyzing the reliability and sharpness at different confidence levels, similar results can be obtained.

Fig 8 and Fig 9 show the prediction interval of solar generation constructed by the LUBE method with and without PIMSE in the loss function, respectively. It is clear that the target values are well covered by the prediction interval. Moreover, the target values are more in the center of the prediction interval constructed by the LUBE method when PIMSE is introduced to the loss function. This shows that the introduction of PIMSE in the loss function makes the LUBE method more robust. From the solar generation of 16 September 2015 and 17 September 2015, it can be seen that the solar generation has highly variable patterns, which indicates that it is difficult to accurately predict the PV power generation by deterministic prediction. This also reflects the importance of the probabilistic prediction of solar generation.

IV. CONCLUSION

In this paper, we propose a novel probabilistic forecasting method based on the SAE and LUBE. In the proposed method, SAE is first utilized to obtain the highly compressed representation of the input data. Then highly compressed features are used as the input of the LUBE method. In addition, PIMSE is introduced to the loss function of the LUBE method to ensure the target value can be located in the center of the prediction interval of solar generation. The proposed method is evaluated by the real solar generation data from DKASC. In terms of PICP, PIMSE, Winkler score, and loss of LUBE, the proposed method has better performance than the compared forecasting method.

The future research work includes the following two aspects: (1) Denoising stacked autoencoders (DSAE) is a variant of AE, which adds noise to the original sample to improve its generalization ability. We will use the DSAE to get highly compressed features to verify the prediction performance of

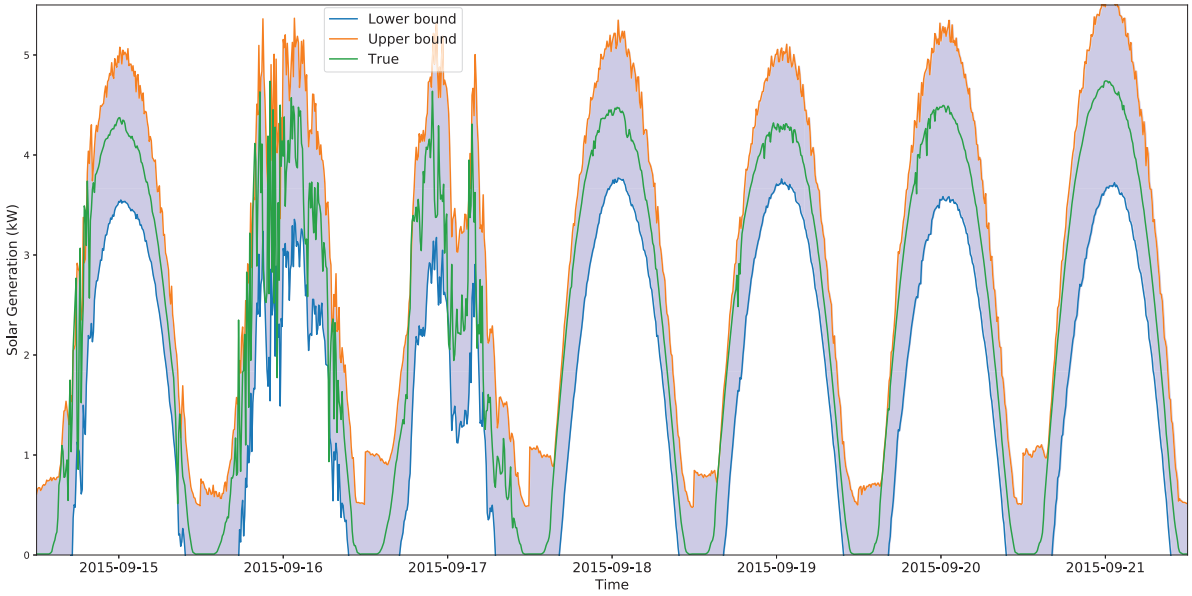

Fig. 8. Prediction interval constructed by the proposed method (PINC=95%, PIMSE is introduced to the loss function of the LUBE method).

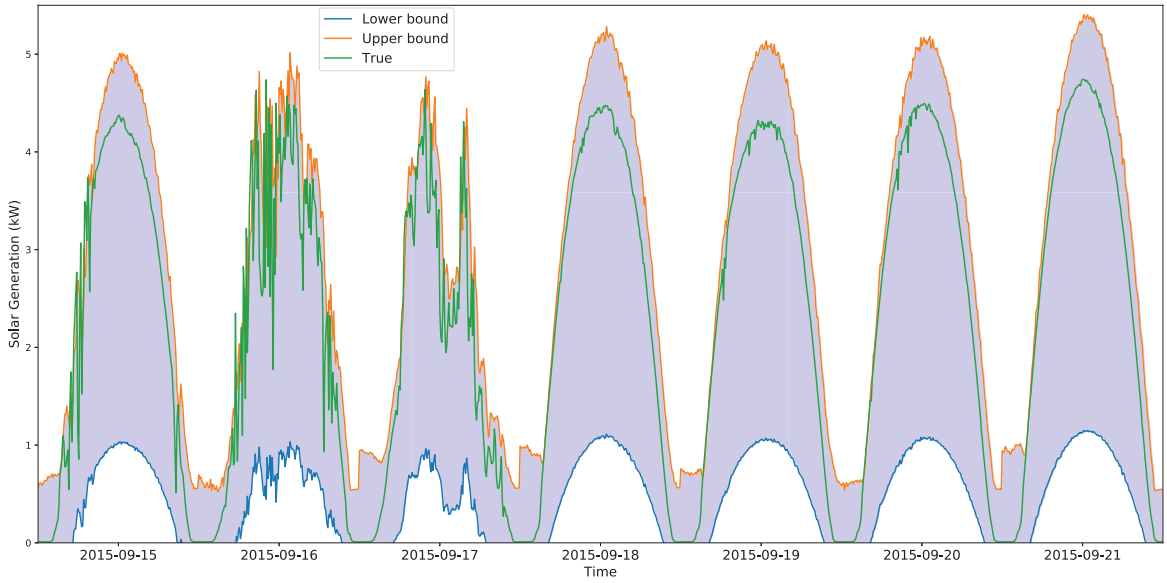

Fig. 9. Prediction interval constructed by the proposed method (PINC=95%, PIMSE is not introduced to the loss function of the LUBE method).

the proposed method. (2) The proposed method only optimizes a specific trade-off between interval coverage and width. We will use a multi-objective optimization [31] to determine the prediction interval.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant U1701262 and U1801263.

REFERENCES

[1] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569–582, 2017.

[2] Y. Gala, Á. Fernández, J. Díaz, and J. R. Dorronsoro, "Hybrid machine learning forecasting of solar radiation values," *Neurocomputing*, vol. 176, pp. 48–59, 2016.

[3] Y. Liu, S. Shimada, J. Yoshino, T. Kobayashi, Y. Miwa, and K. Furuta, "Ensemble forecasting of solar irradiance by applying a mesoscale meteorological model," *Solar Energy*, vol. 136, pp. 597–605, 2016.

[4] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," *CSEE Journal of Power and Energy Systems*, vol. 1, no. 4, pp. 38–46, 2015.

[5] D. Liu and K. Sun, "Random forest solar power forecast based on classification optimization," *Energy*, vol. 187, p. 115940, 2019.

[6] M. Paulescu and E. Paulescu, "Short-term forecasting of solar irradiance," *Renewable Energy*, vol. 143, pp. 985 – 994, 2019.

[7] J. Dong, M. M. Olama, T. Kuruganti, A. M. Melin, S. M. Djouadi, Y. Zhang, and Y. Xue, "Novel stochastic methods to predict short-term solar radiation and photovoltaic power," *Renewable Energy*, vol. 145, pp. 333 – 346, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148119307463

[8] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[9] N. Meinshausen, "Quantile regression forests," *Journal of Machine Learning Research*, vol. 7, no. Jun, pp. 983–999, 2006.

[10] C. Rusmassen and C. Williams, "Gaussian process for machine learning," 2005.

[11] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE transactions on neural networks*, vol. 22, no. 3, pp. 337–346, 2010.

[12] Q. Ni, S. Zhuang, H. Sheng, G. Kang, and J. Xiao, "An ensemble prediction intervals approach for short-term pv power forecasting," *Solar Energy*, vol. 155, pp. 1072–1083, 2017.

[13] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 303–315, 2013.

[14] A. Khosravi, S. Nahavandi, and D. Creighton, "Improving prediction interval quality: A genetic algorithm-based method applied to neural networks," in *International Conference on Neural Information Processing*. Springer, 2009, pp. 141–149.

[15] C. Wan, Z. Xu, and P. Pinson, "Direct interval forecasting of wind power," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4877–4878, 2013.

[16] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[17] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[18] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[19] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International conference on machine learning*, 2016, pp. 1378–1387.

[20] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[21] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[22] H. Liu and C. Chen, "Multi-objective data-ensemble wind speed forecasting model with stacked sparse autoencoder and adaptive decomposition-based error correction," *Applied Energy*, vol. 254, p. 113686, 2019.

[23] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[24] L. Wang, Z.-H. You, X. Chen, S.-X. Xia, F. Liu, X. Yan, Y. Zhou, and K.-J. Song, "A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network," *Journal of Computational Biology*, vol. 25, no. 3, pp. 361–373, 2018.

[25] M. Kang, K. Ji, X. Leng, X. Xing, and H. Zou, "Synthetic aperture radar target recognition with feature fusion based on a stacked autoencoder," *Sensors*, vol. 17, no. 1, p. 192, 2017.

[26] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting — an approach using autoencoder and lstm neural networks," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2016, pp. 002 858–002 865.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization (pso)," in *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, 1995, pp. 1942–1948.

[28] D. K. A. S. Centre, http://dkasolarcentre.com.au/, accessed 2019-12-30.

[29] S. Alessandrini, L. Delle Monache, S. Sperati, and G. Cervone, "An analog ensemble for short-term probabilistic solar power forecast," *Applied energy*, vol. 157, pp. 95–110, 2015.

[30] J. Zhang, A. Florita, B.-M. Hodge, S. Lu, H. F. Hamann, V. Banunarayanan, and A. M. Brockway, "A suite of metrics for assessing the performance of solar power forecasting," *Solar Energy*, vol. 111, pp. 157–175, 2015.

[31] P. Cortez, P. J. Pereira, and R. Mendes, "Multi-step time series prediction intervals using neuroevolution," *Neural Computing and Applications*, pp. 1–15, 2019.