Theory-guided deep-learning for wind power forecasting (TgDPF) via long short-term memory

Abstract

Wind power plays a critical role in achieving carbon neutrality as one of the key renewable energy sources. Accurate wind power forecasting helps to better manage wind farms and wind application systems. In this paper, we propose theory-guided deep-learning wind power forecasting (TgDPF) to solve this issue. TgDPF skillfully combines the wind power curve which represents human knowledge with a classical deep-learning model long short-term memory (LSTM). In TgDPF, the actual wind power curve is calculated by the method of kernel density estimation at first, then the gap between the wind power curve generated by the LSTM model and the actual wind power curve (JS divergence is used to measure the gap between two curves) is introduced into the training process of the LSTM. During the training process, the forecasts that do not conform to the wind power curve gradually decrease. Compared with the origin mean square error (MSE) loss trained LSTM, the predicted power of the TgDPF conforms to the pre-calculated wind power curve, which improves the reliability of the predictions. Experiments on different wind turbines show that the performance of TgDPF is obviously better than that of LSTM when adding noise of different proportions to wind speed. Furthermore, TgDPF becomes more robust after adding human knowledge, which does not over-fit the noise in the wind speed.

**Keywords:** wind power forecasting; theory-guided; deep-learning; LSTM; human knowledge; wind power curve; kernel density estimation; JS divergence.

1. Introduction

Wind power is one of the fastest growing forms of energy globally, with an important impact on both industrial production and daily life [1]. As a kind of energy obtained from wind, wind power has the characteristics of discontinuity, variability, and non-schedulability. At the same time, it is vulnerable to short-term extreme weather, such as storm and rainstorm. Meanwhile, if there is no energy storage system, wind power must be used immediately after being produced. Based on the above characteristics, accurate wind power forecasting is essential for efficient management of wind farms and wind application systems [2]. Wind power forecasting is mainly divided into long-term, medium-term and short-term. The forecasting of the next day (6h to 1 day ahead) is usually called medium-term forecasting [3]. The demand for medium-term forecasting is reflected in the market operation of wind farms, which is more important for the utilization of wind power [4]. Therefore, this study focuses on medium-term forecasting of wind power. Futhermore, forecasting of renewable energy often relies on weather forecasts. However, compared to other renewable energy sources such as photovoltaic, wind speed fluctuations are larger and more pronounced than irradiance, which results in wind speed forecasts having more noise and lower accuracy. Therefore, wind power forecasting is more challenging due to the greater noise associated with weather forecast data. It is thus crucial to leverage prior information such as wind power curves to enhance model predictive capability for noisy data.

Researchers are increasingly interested in wind power forecasting, and many forecasting models have been developed based on it. These models can be broadly separated into two categories: domain knowledge-based models and data-driven models.

In knowledge-based models, physical features are employed to simulate and forecast wind power. In [46], a novel smoothing method was proposed which addresses three physical problems associated with wind power. Shao et al. utilized the wavelet decomposition and AdaBoost technique to develop a wind power forecasting model in [47]. Furthermore, Knowledge-based models include many Numerical Weather Prediction (NWP) models [5, 6]. By simulating the local climate and meteorology, various initial states, and boundary information, future wind speed and direction are predicted by NWP models. The wind power is then projected using the wind power curve [7]. Although these knowledge-based models have good interpretability, but in order to perform well, they frequently require to collect and extract a huge number of physical features, and they cannot efficiently utilise the readily available historical data.

Data-driven models can forecast future wind power only by using historical data. Data-driven models can be further divided into traditional statistical models and deep-learning models. For the traditional statistical models, Wang et al. used autoregressive moving average (ARMA) [8] and Chen et al. used autoregressive integrated moving average (ARIMA) to model and forecast wind power [9]. In contrast to knowledge-based models, these time series models are typically used to characterize the linear fluctuation of wind power at various locations, and they exhibit good performance in short-term wind power forecasting. However, for medium-term and long-term wind power forecasting, the capabilities of these traditional statistical models are insufficient. With the development of deep learning technology, many deep-learning models are also applied to wind power forecasting. Singh et al. utilized a multilayer perceptron (MLP) network for wind power forecasting [10], which is the simplest form of deep-learning model. RNN [11] based models are also widely used in wind power forecasting, such as LSTM [12], GRU [13], bidirectional LSTM (BiLSTM) [14], and bidirectional GRU (BiGRU) [15]. Furthermore, Yu et al. applied CNN [16] and Mezaache et al. applied auto-encoder (AE) [17] to wind power forecasting. Deep-learning models show better wind power forecasting performance than traditional statistical models, thanks to their strong ability to deal with complex nonlinear problems. Whether traditional statistical models or deep-learning models, these data-driven models need a large amount of data to achieve good results. What’s more, the data-driven models may be trapped in a local minimum and fail to achieve the highest accuracy during training without the assistance of domain knowledge.

Some research has attempted to mix domain knowledge with data-driven models because pure domain knowledge-based models and pure data-driven models are often insufficient to handle complicated situations. On the other hand, domain knowledge is routinely used for feature engineering, but it has not yet been fully integrated with deep-learning models [18]. In this situation, domain knowledge is frequently underused. To deal with this issue, a theory-guided framework was suggested and achieved good performance in many fields. For instance, theory-guided neural networks (TgNNs) were successfully applied in the field of hydrology by Wang et al. and and He et al. [19, 20]. Karpatne et al. suggested five methods for combining scientific knowledge and data science in their theory-guided data science (TGDS) proposal [21]. For computational fluid dynamics, Raissi et al. used the physics-informed neural network (PINN) [22], which is essentially a type of theory-guided neural network (TgNN), and achieved good results. As a prediction model for oil/water phase flow, Li et al. developed a TgNN [23]. To verify that the model outputs respect known governing equations, Chen et al. created a sort of hard constraint model under the theory-guided framework [24]. TgDLF was developed in the area of electrical load forecasting referring to the idea of theory-guided methods [25]. In the field of wind power forecasting, Huang et al. proposed a Tg-OFNN [26]. In Tg-OFNN, wind power is divided into two parts, the first of which can be learned from human knowledge and the second of which can be approximated by a deep-learning model. Additionally, they developed a priori-guided and data-driven hybrid model for wind power forecasting [27], which divides the forecasting process into one theory-guided stage and two data-driven stages. The two theory-guided methods mentioned above have used the deterministic domain knowledge, however, many prior information exist in the form of probability distributions, such as wind power curves. However, current methods are unable to effectively embed such probability distributions into the wind power forecasting model, leading to the insufficient utilization of domain knowledge. In this study, we propose theory-guided deep-learning wind power forecasting (TgDPF), in which human knowledge is applied in the form of embedded probability distributions to aid the training of deep-learning model.

In TgDPF, the probability distribution of wind power is embedded into the training process of the deep-learning model LSTM. Specifically, the wind power curve representing the probability distribution of wind power can be calculated by the method of kernel density estimation [36], and the difference of probability distribution can be obtained by JS divergence [43]. The loss of LSTM can be divided into two parts, one part represents the difference between the wind power distribution predicted by the model and the actual wind power distribution, and this part of loss comes from human knowledge; The other part of the loss is the MSE loss, which is purely data-driven loss. These two losses can be effectively combined to make the final trained model more accurate and robust.

To the best of the authors’ knowledge, not only in the field of wind power, but also in the field of energy, probability distribution is the first time to be embedded into the training process of deep-learning model as domain knowledge. In the previous research, the knowledge embedded is generally the deterministic formula information. Adding probability distribution to the training process of deep-learning model broadens the scope of knowledge embedding, which makes some weakly related or uncertain knowledge can assist training. However, adding probability distributions to models poses certain challenges, such as ensuring the differentiability of models after distribution embedding and measuring the similarity between distributions. This method can also be easily applied to other problems in the energy field.

The contribution of this study is three-fold:

* (1) This study proposes an accurate and robust theory-guided model for wind power forecasting.
* (2) This study firstly presents the exploration of introducing probability distribution constraints in deep learning models and embedding the probability distribution of wind power curves into the wind power forecasting model in the energy field.
* (3) This study uses the kernel density estimation method to calculate the wind power curve, and uses JS divergence to calculate the distance between the wind power distributions. This process ensures that the calculation graph of the deep-learning model is derivable.

1. Methodology

In this work, theory-guided deep-learning wind power forecasting (TgDPF) is used to forecast the wind power. In this section, the deep-learning model LSTM is introduced first, followed by a demonstration of the wind power curve and kernel density estimation. Next, we introduce the Jensen-Shannon divergence (JSD) metric that can measure the difference of different wind power curves. Finally, we show how to combine the domain knowledge and deep-learning model in TgDPF.

* 1. *LSTM*

Long short-term memory (LSTM) [28] is classical neural network for processing sequential data, such as weather data [29], voice data [30] and oil well data [31]. It was developed to overcome the gradient vanishing and gradient explosion problem of the traditional recurrent neural network (RNN). The internal structure of a LSTM cell is shown in Fig. 1. The core of a LSTM cell is cell state *Ct*, which can pass through the LSTM cells, carrying information of previous steps. The interior of a LSTM Cell contains a forget gate *ft*, an input gate *it* and an output gate *ot*, which work together with the hidden state h of the previous step and the input X of this step to determine to remove or add information to the cell state.

The forget gate determines what information to forget, and it can be described by Eq. (1). The output *ft* is a number with a range of [0,1], which will be multiplied with the cell state from previous step in Eq. (4).

(1)

The input gate determines what new information should be stored in the cell state, and it can be described by Eq. (2). The output *it* is also a number with a range of [0,1], which will be multiplied with the new candidate cell state in Eq. (4).

(2)

And the new candidate cell state which contains the information of the hidden state h and the input X is defined by Eq. (3):

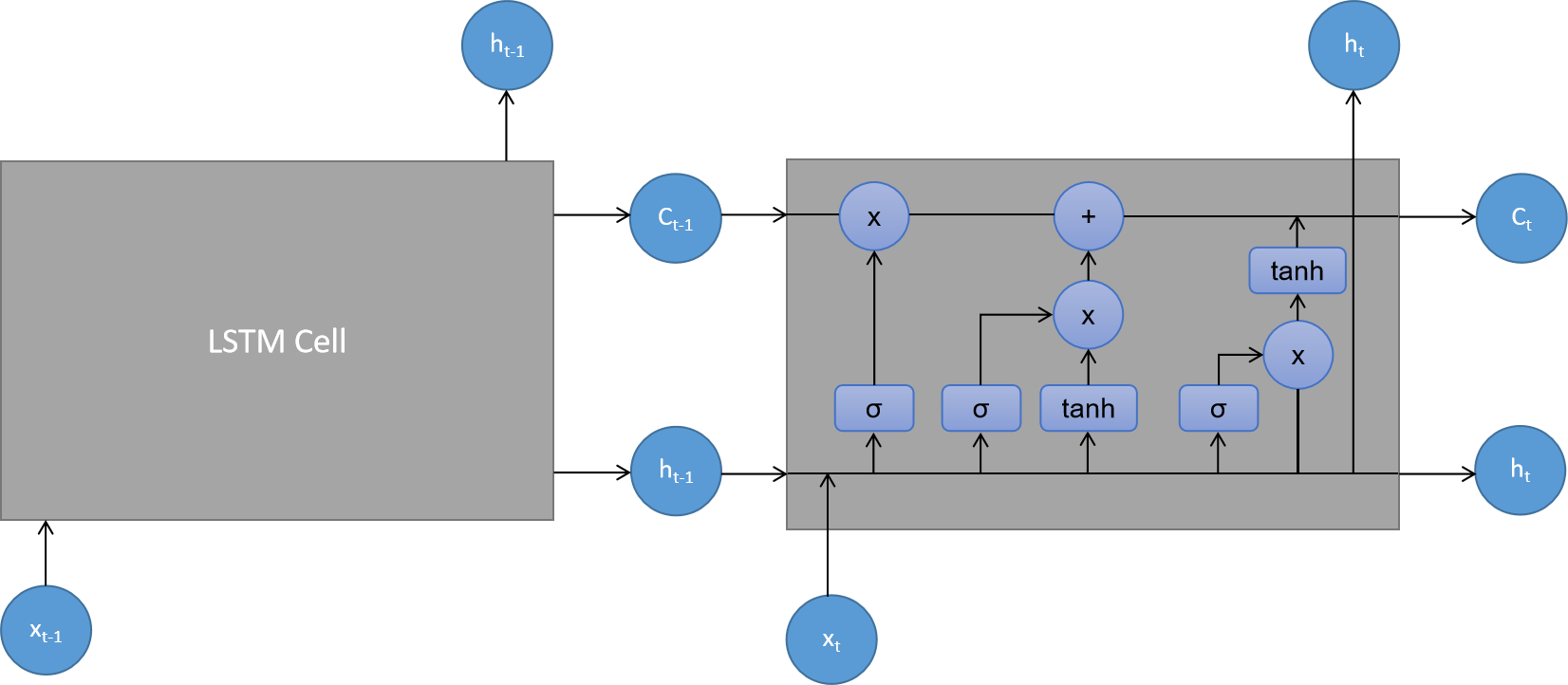
(3)

The final updated cell state *Ct* is composed of the cell state in the previous step and the new candidate cell state, as expressed in Eq. (4). The forget gate *ft* and the input gate *it* control the deletion and retention of information here.

(4)

The output gate *ot* determines the output of the LSTM cell, and it can generate the output *ht* (it will also be transmitted to the next LSTM cell as a hidden state) with the updated cell state *Ct*. The whole process can be described by Eq. (5).

(5)



**Fig. 1.** Structure of LSTM.

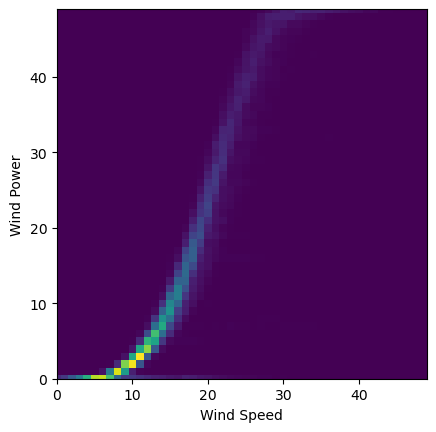
Although the internal structure of LSTM is complex, its internal complex components have been proven to alleviate the gradient vanishing and gradient explosion problem [32]. LSTM cells at different time steps share parameter weights, and the information of LSTM cells from previous time steps is transmitted to later time steps through hidden states and cell states. LSTM can not only extract short-term features from series data like the traditional RNN, but also capture long-term dependencies in series data [33]. Based on the above features, LSTM is an ideal deep-learning model for wind power forecasting.

* 1. Wind power curve and kernel density estimation

Wind turbines can convert wind energy into electric energy, and the wind power curve can describe the conversion capacity of the wind turbine. The wind power curve refers to the curve corresponding to the output power of the wind turbine and the wind speed, which reflects the mechanical capabilities of the wind turbine. The power generation of a wind turbine is largely affected by the wind speed, and different types of wind turbines have different performances. Through the wind power curve, we can intuitively understand the power output of the wind turbine without knowing its technical details.

The wind power curve can express the relationship between the wind speed and the wind power generated by wind turbines, as shown in Fig. 2.It describes the joint probability distribution of wind speed and wind power. The minimum wind speed that drives the turbine to generate power is called “cut in speed”. When the turbine power generation reaches the maximum, in order to protect the turbine, even if the wind speed continues to increase, the power generated by the turbine will not increase. The maximum speed at which the turbine functions normally is called “cut off speed” [34].

Different methods available for wind power modeling can be divided into parametric methods and non-parametric methods [35]. The parametric method has a pre-assumption on the distribution of data. The parameter quantity of the parametric method is fixed and does not change with the number of training samples, so it usually consumes relatively less computing resources. Experiences and theories show that there is often a big gap between the pre-assumption of data distribution in parametric models and the actual data distribution, and these methods cannot always achieve satisfactory results. On the other hand, the non-parametric method does not make any assumptions about the distribution of data, so it is usually more flexible and accurate. However, when the amount of data is too large, non-parametric methods consume a lot of computing resources. With the massive promotion of renewable energy, many wind turbines have limited operating time and have accumulated relatively little historical data that can be used for training. As a result, wind power forecasting models often face the challenge of insufficient data. In this study, the number of training samples is only tens of thousands, so we use a non-parametric method called kernel density estimation to model the wind power.



**Fig. 2.** Wind power curve.

Kernel density estimation (KDE), which is also known as Parzen's window [36], is used to estimate unknown density function. It can learn the shape of the density from the data automatically. Let (X1, X2, … , Xn) be independent, identically distributed random samples from an unknown distribution P with density function f, the kernel density estimation can be expressed as:

(6)

where *n* represents the number of samples, *h* is called bandwidth, and *K* is the kernel function. The Gaussian kernel function is usually adopted, and its formula is:

(7)

Research shows that the selection of kernel function has little influence on the effect of kernel density estimation. On the other side, the selection of bandwidth has a great impact on the estimation results [37, 38]. The bandwidth is also called the smoothing coefficient. If the bandwidth is small, there will be many fluctuations in the density estimation function, and if the bandwidth is too large, a lot of details will be missing. The kernel density estimation can reflect the real distribution only if the appropriate bandwidth is selected. There are many methods to determine bandwidth, including urle fo thumb [39], least square cross validation [40], biased cross-validation [41], plug-in method [42], etc. In this study, a method similar to Scott's rule of thumb [38] is used to select bandwidth, and the bandwidth *h* is calculated as follows:

(8)

where is the unbiased standard deviation of data, and *n* represents number of bins, and is an adjustable parameter.

Every step of bandwidth calculation and kernel density estimation is derivable, so the gradient here is easy to obtain.

* 1. JS Loss

Since the wind power curve as the optimization target is a distribution, Jensen-Shannon divergence (JSD) [43] is introduced to calculate the distance between the model predicted distribution and the actual distribution. Before talking about JSD, let's first introduce Kullback-Leibler divergence (KLD) [44], which is also a basic component of JSD.

KLD is also called information distance or relative entropy, which represents the difference of information entropy of probability distribution. Assuming that P and Q are two probability distributions, KLD can be expressed as:

(9)

However, KLD does not satisfy symmetry, that is, *KL(P||Q)* isnot equal to *KL(Q||P)*, so many researchers do not think it is a good metric to measure distribution distance.

JSD is based on KSD, and its formula is:

(10)

JSD is a bounded measurement with a range of [0, 1]. If the two distributions are identical, the JSD is equal to 0. At the same time, JSD has symmetry, so it is a relatively ideal metric. If P and Q of the Eq. (10) are respectively taken as wind power curve generated by the deep-learning model and the actual wind power curve, JS loss will be generated.

However, there is a disadvantage of JS divergence: The calculation process of JSD is originally derivable, but if the two distributions do not overlap at all, JSD will become a fixed value log2 [45]. In this case, the gradient is 0 and the model cannot be updated according to the difference of distributions. In the training process of the deep-learning model, the distribution of the output of the model in the initial state may not overlap with the real distribution at all, this is problem to be considered in the training process.

* 1. TgDPF

In this study, based on the expressive time series prediction capability of LSTM and the physical knowledge contained in the wind power curve, we propose theory-guided deep-learning wind power forecasting (TgDPF), which combines the deep-learning model LSTM and wind power curve.

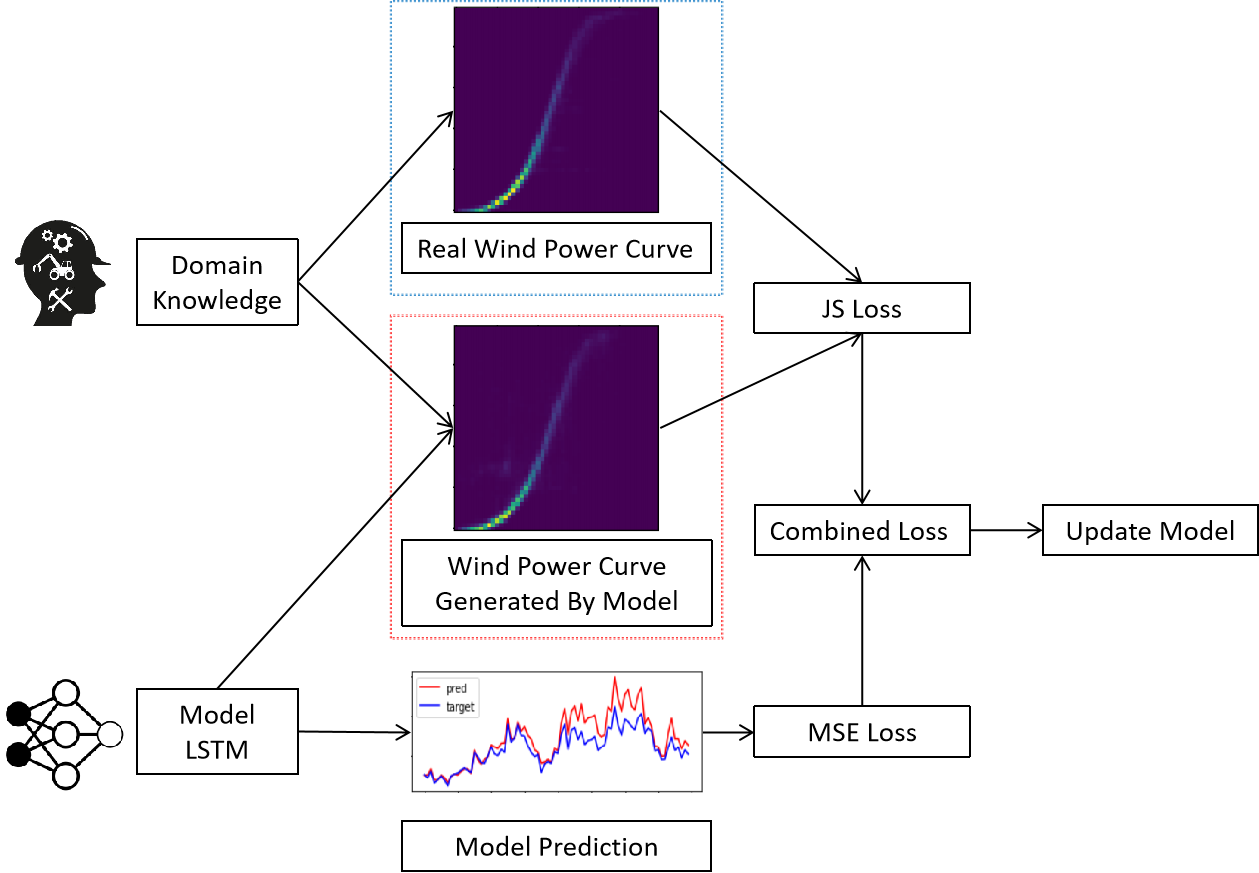
The input of the model LSTM used in TgDPF has three dimensions: (1) batch size, which is the number of samples used in each training iteration, and the batch size used during training is 300; (2) time step, which refers to how long the historical data is used for training, and the time step used here is 576 (the training uses 4-days historical data, and the data is sampled at 10 minutes intervals, so the length of the time step is 576); and (3) the feature dimension, which contains a historical wind power feature together with a wind speed and a pitch angle of the wind turbine, for a total of three features. The model LSTM contains 2 LSTM layers, and the dimension of the hidden layer is 20. Two fully connection layers are connected behind the LSTM layer to enhance the expression ability of the model.

Before training the LSTM model, it is also necessary to obtain the actual wind power curve. As the physical properties of different wind turbines in the experiments are very similar, this study uses the average wind power curve as the actual wind power curve to simplify the calculation. In order to get the average wind power curve, we need to calculate the average wind speed and average power of all wind turbines at the same time step, and then we can get the average wind power curve according to the KDE method mentioned above.

In TgDPF, the loss consists of two parts: the first part is the aforementioned JS loss, and it is obtained from the wind power curve generated by the model and the actual wind power curve, and this part of loss can be called domain knowledge loss; the second part is the MSE loss obtained from the predicted wind power and the actual wind power, and this part of loss can be called deep-learning loss. During training, these two losses are combined linearly to update the weight of the deep-learning model.

In the training process of the model LSTM, the distribution of the output of the model in the initial state may not overlap with the real distribution at all, which will cause the JS loss to become a fixed value and its gradient to disappear. Our solution is to set some threshold conditions, including the prediction range of the model, the standard deviation of the prediction value, etc. When the predictions of the model LSTM are within the threshold range, the JS ratio (the weight of JS loss) is set to 1, and the combined loss of MSE loss and JS loss is used to update the model. Otherwise, the JS ratio is set to 0, and the pure MSE loss is used to update the model LSTM. Through testing, it is found that the JS ratio may be set to 0 only in the first few iterations, and then it will always be set to 1, and JS loss can be used to guide the optimization of the model after that.

The process of TgDPF is shown in Fig. 3.



**Fig. 3.** Flowchart of TgDPF.

1. Experiment
   1. Data description and experiment setting

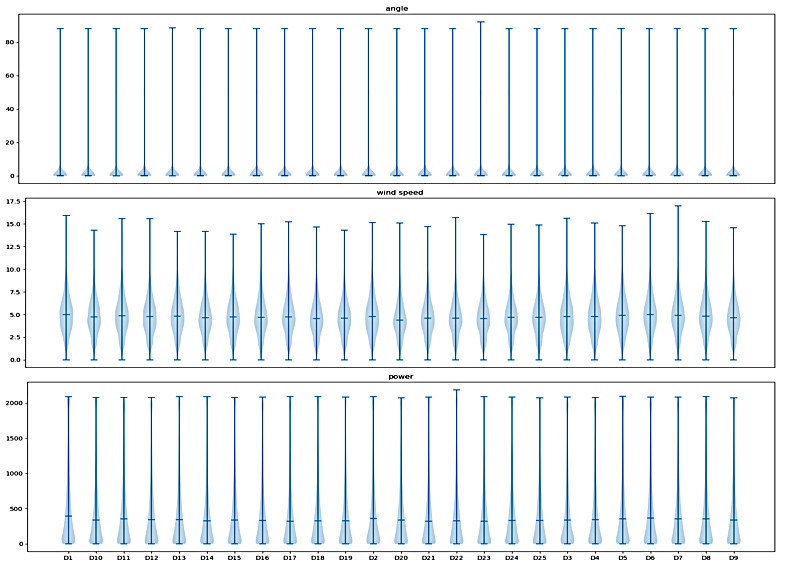
In this study, we take the data of wind turbines in Jiangsu Province, China, as a study case. The whole data set consists of data of 25 wind turbines in 2020, which includes three features of pitch angle, wind speed and wind power. All features are sampled at a frequency of 10 minutes, so each wind turbine contains about 50000 data. Features distributions of these 25 wind turbines are shown in Fig. 4. The figure illustrates that the features distributions of different wind turbines are very similar.

We use the data from January to October for training, and the data from November and December for testing. The length of input data is four days (4 d), including the three features of pitch angle, wind speed and wind power of the wind turbine. The wind speed data is the historical wind speed data of three days and the data of the next 24 hours (equivalent to wind speed forecast data). The wind speed forecast data is simulated with real wind speed with noise added. The formula for adding random noise to wind speed is as follows:

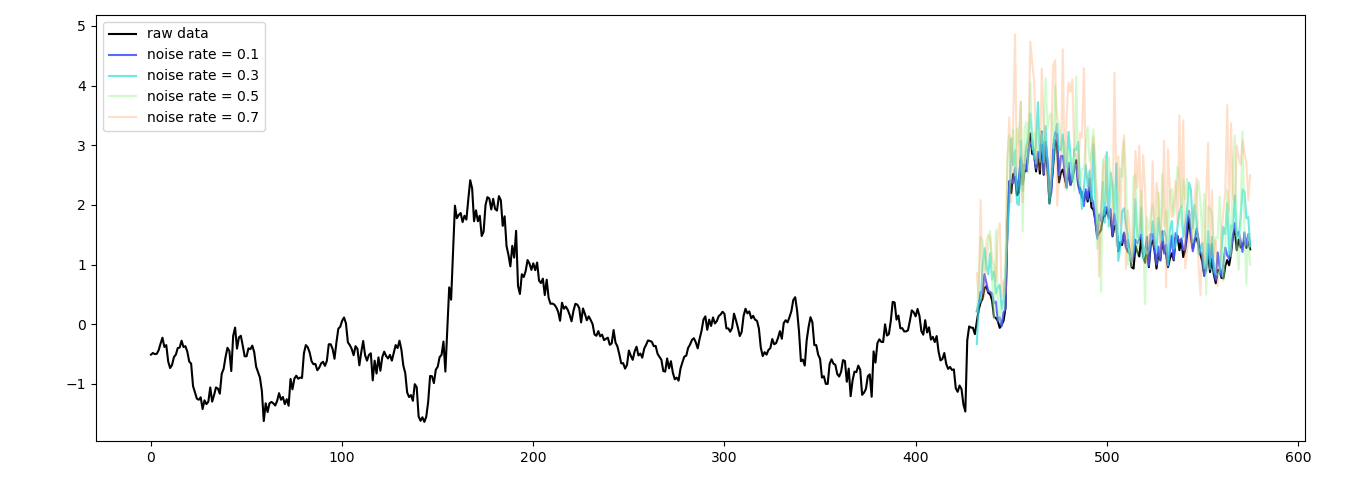
(11)

where *S’t+1* and *St+1* represent the noisy wind speed data and standardized real wind speed data at time *t+1*, respectively; *Noise* represents normally distributed noise, two kinds of noise are added. One is *N* (0, x) (normal distribution with 0 as the mean and x as the standard deviation), which is unbiased noise. It represents the scene where the actual wind speed is the predicted wind speed plus random disturbance; The other is *N* (x, x) (normal distribution with x as the mean and x as the standard deviation), which is biased noise. It represents the scene where the actual wind speed is offset from the predicted wind speed. Fig. 5 shows the wind speed after adding *N* (x, x) noise. It can be seen from Fig. 5 that the wind speed data fluctuates greatly after adding high biased noise, which is obviously unreasonable. Unfortunately, due to the instability of wind speed, the forecast wind speed is often quite different from the real wind speed (the forecast wind speed is the real wind speed adding high noise).

The samples for training and testing in TgDPF are produced using a moving window approach. The initial input window contains the first 4 d of historical wind power data, pitch angle, and wind speed (3 d of historical data and 1 d of forecast data), and the initial output window contains the wind power data for the fifth day. The initial sample is produced in this manner. More samples are generated each time the input window slides forward 1 d, and the output window does the same.



**Fig. 4.** Pitch angle, wind speed and wind power distribution of wind turbines.



**Fig. 5.** Wind speed after adding *N* (x, x) noise.

* 1. Wind power forecasting experiments

In wind power forecasting experiments, we add different proportions of noise to the wind speed of training set to measure the effects of TgDPF under different degrees of interference, and the results of test sets are shown in Table 1 and Table 2. As a contrast, the training results of LSTM trained only with MSE loss are also listed in the table. Table 1 shows the experimental results of adding *N* (0, x) noise to training set, and Table 2 shows the experimental results of adding *N* (x, x) noise to training set. Each large column, such as *N* (0, 0.1), indicates the experimental results of adding such noise to the wind speed in the training set during the training process. In experiments, MSE losses in test set are used to evaluate models. The number (10, 20, …) in the small rows show how many epochs are run; LSTMs in the small columns show the results of LSTM trained only with MSE loss, TgDPFs in the small columns show results of TgDPF, Impros show the improvements of TgDPF over LSTM, and the results in each grid are the average of five independent experiments. In Table 1 and Table 2, no noise is added to the wind speed data of the test sets while testing. It is shown in the Table 1 and Table 2 that the performance of TgDPF is obviously better than that of LSTM when adding noise of different proportions. Fig. 6 intuitively shows the prediction performance of LSTM and TgDPF. It can also be seen from the Fig. 6 that the predicted wind power of TgDPF is closer to the real wind power than that of LSTM.

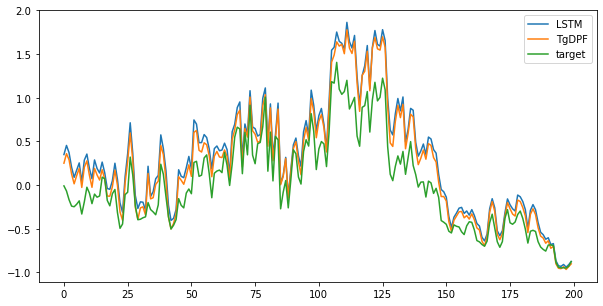


Fig. 6. Wind power forecasting of LSTM and TgDPF.

**Table 1.** Wind power forecasting experimental results of adding *N* (0, x) noise to training set.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Epoch | *N*(0, 0.1) | | | *N*(0, 0.3) | | | *N*(0, 0.5) | | | *N*(0, 0.7) | | |
| LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) | LSTM | TgDPF(%) | Impro | LSTM | TgDPF | Impro(%) |
| 10 | 0.0368 | 0.0322 | 12.5 | 0.0442 | 0.0374 | 15.4 | 0.0604 | 0.0466 | 22.8 | 0.0768 | 0.0620 | 19.3 |
| 20 | 0.0372 | 0.0307 | 17.5 | 0.0454 | 0.0356 | 21.6 | 0.0589 | 0.0464 | 21.2 | 0.0770 | 0.0562 | 27.0 |
| 30 | 0.0329 | 0.0300 | 8.8 | 0.0417 | 0.0372 | 10.8 | 0.0589 | 0.0463 | 21.4 | 0.0820 | 0.0504 | 38.5 |
| 40 | 0.0315 | 0.0301 | 4.4 | 0.0412 | 0.0376 | 8.7 | 0.0616 | 0.0443 | 28.1 | 0.0881 | 0.0470 | 46.7 |
| 50 | 0.0327 | 0.0319 | 2.4 | 0.0423 | 0.0385 | 9.0 | 0.0626 | 0.0439 | 29.9 | 0.0903 | 0.0475 | 47.4 |

**Table 2.** Wind power forecasting experimental results of adding *N* (x, x) noise to training set.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Epoch | *N*(0.1, 0.1) | | | *N*(0.3, 0.3) | | | *N*(0.5, 0.5) | | | *N*(0.7, 0.7) | | |
| LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) |
| 10 | 0.0310 | 0.0318 | -2.6 | 0.1017 | 0.0596 | 41.4 | 0.2409 | 0.1069 | 55.6 | 0.4242 | 0.1126 | 73.5 |
| 20 | 0.0310 | 0.0300 | 3.3 | 0.0956 | 0.0544 | 43.1 | 0.2352 | 0.0764 | 67.5 | 0.3994 | 0.0968 | 75.8 |
| 30 | 0.0306 | 0.0294 | 3.9 | 0.1013 | 0.0530 | 47.7 | 0.2548 | 0.0721 | 71.7 | 0.4051 | 0.1001 | 75.3 |
| 40 | 0.0324 | 0.0300 | 7.4 | 0.1218 | 0.0560 | 54.0 | 0.2663 | 0.0870 | 67.3 | 0.3603 | 0.0845 | 76.5 |
| 50 | 0.0331 | 0.0298 | 10.0 | 0.1081 | 0.0492 | 54.5 | 0.2609 | 0.0682 | 73.9 | 0.3104 | 0.0984 | 68.3 |

**Table 3.** Wind power forecasting experimental results of adding the same high noise to training set and test set.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Epoch | *N*(0, 0.5) | | | *N*(0, 0.7) | | | *N*(0.5, 0.5) | | | *N*(0.7, 0.7) | | |
| LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) | LSTM | TgDPF | Impro(%) |
| 10 | 0.1131 | 0.1199 | -6.0 | 0.1457 | 0.1596 | -9.5 | 0.1134 | 0.1551 | -36.8 | 0.1466 | 0.2044 | -39.4 |
| 20 | 0.1082 | 0.1163 | -7.5 | 0.1371 | 0.1553 | -13.3 | 0.1096 | 0.1269 | -15.8 | 0.1379 | 0.1500 | -8.8 |
| 30 | 0.1053 | 0.1165 | -10.6 | 0.1366 | 0.1515 | -10.9 | 0.1048 | 0.1162 | -10.9 | 0.1358 | 0.1445 | -6.4 |
| 40 | 0.1042 | 0.1136 | -9.0 | 0.1374 | 0.1454 | -5.8 | 0.1045 | 0.1111 | -6.3 | 0.1401 | 0.1428 | -1.93 |
| 50 | 0.1057 | 0.1097 | -3.8 | 0.1357 | 0.1438 | -6.0 | 0.1059 | 0.1129 | -6.6 | 0.1420 | 0.1418 | 0.1 |

Furthermore, when the noise ratio is high, the prediction effect of LSTM will be significantly worse, while the effect of TgDPF is relatively stable. We think that the reason for this phenomenon is that LSTM has over-fitted the noise in the wind speed, so that when the model is tested with test set data without noise, the effect of LSTM will decline a lot. In order to verify our idea, we also tested the effect of the model when adding the same high noise to the wind speed of the test sets as the training sets, and the results are shown in Table 3. Each large column, such as *N* (0, 0.5), indicates the experimental result of adding such noise to the wind speed in the training set during the training process and to the wind speed in the test set during the testing process, and the number (10, 20, …) in the small rows also indicate how many epochs are run. It can be seen from Table 3 that when the same high noise as the training set is added to the test set, the effect of LSTM immediately improves a lot, and in most cases is significantly better than that of TgDPF, which is obviously not what we want to see. In contrast, human knowledge in TgDPF will help the model find that the wind speed with high noise in the test set is unreasonable, so it is impossible to obtain accurate wind power based on unreasonable wind speed, which further proves the robustness of TgDPF.

During the training, it can be seen that the wind power curve generated by TgDPF is gradually approaching the actual wind power curve, as shown in Fig. 7. The abscissa of each sub figure in Fig. 7 represents the wind speed, and the ordinate represents the wind power. Fig. 7 selects four different stages of training according to the training process: the first stage is the initial stage, and the weights of the model are randomized. The wind power curve generated by the model is obviously not reliable in this stage; In the second stage, the model has learned some knowledge, and the wind power curve generated by the model has been very similar to the actual wind power curve in the low wind speed part. According to our hypothesis, this is because the distribution of wind power in the low wind speed part is more concentrated, making it simpler for the model to learn the wind power distribution of this part. In the third stage, the wind power curve generated by the model is very similar to the actual wind power curve in profile, but slightly different from the actual curve in detail; In the fourth stage, which is also the last stage of training, the wind power curve generated by the model is almost consistent with the actual wind power curve.

We also tried updating the model with solely JS loss, but the results weren't great. We believe the cause is that the JS loss only represents the difference of probability distributions, and it does represent the difference of actual predicted values (even if the two probability distributions are the same, their sampling values may differ greatly). Therefore, it is not sufficient to update the LSTM model alone with JS loss; MSE loss, which indicates the difference between actual predicted values, should also be used.

1. Discussion

In this study, we proposed TgDPF, an accurate and robust model for middle-term wind power forecasting. In TgDPF, human knowledge is embedded into deep-learning model training process as a part of the loss function. TgDPF achieves positive results and contributes the following: (1) TgDPF achieves better results than pure data-driven LSTM and does not over-fit the noise in the wind speed, demonstrating its effectiveness and robustness; (2) This study is the first in the field of wind energy, and even the entire energy field, to embed domain knowledge into the training process of a deep-learning model in the form of probability distribution, which broadens the scope of knowledge embedding; and (3) at the implementation level, this study calculates the wind power curve using the kernel density estimation method and the distance between the wind power distributions using JS divergence. This procedure ensures that the calculation graph of the deep-learning model is derivable. Researchers can apply the implementation details of this knowledge embedding to other energy-related problems.

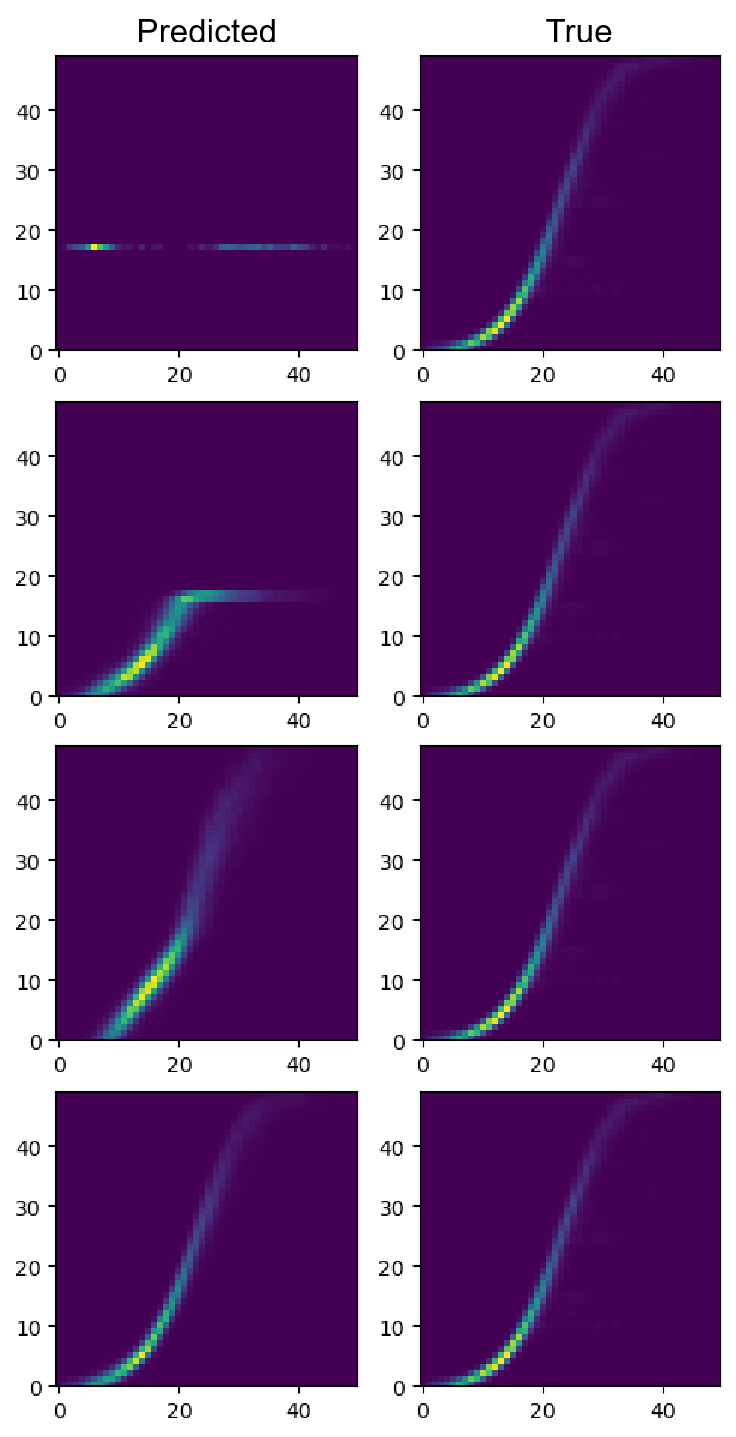


Fig. 7. Wind power curves generated by TgDPF during training

We consider the following as future work for TgDPF: (1)The current preprocessing of wind power data only involves a simple standardization; additional preprocessing of wind power data, such as ratio, decomposition of wind power data into dimensionless trends and local variations, etc., can be performed using human expertise; (2) the current deep-learning model for training is LSTM, which may be replaced by a more efficient model, such as GRU, Transformer, etc; and (3) our experiments have proved that TgDPF has a good performance on medium-term wind power forecasting. It’s also possible to lengthen or shorten the forecasting duration (historical data should also be adjusted accordingly) to test the performance of TgDPF on short-term or long-term wind power forecasting, and further improve the model based on these results. If one of these proposals is proven to work, TgDPF will achieve better performance or have a wider scope of application.

Acknowledgment

This work is funded by the National Natural Science Foundation of China (Grant No. 62106116), the Shenzhen Key Laboratory of Natural Gas Hydrates (Grant No. ZDSYS20200421111201738), and the SUSTech - Qingdao New Energy Technology Research Institute.

REFERENCES

1. Liu H, Chen C. Data processing strategies in wind energy forecasting models and applications: A comprehensive review[J]. Applied Energy, 2019, 249: 392-408.
2. Chen C, Duan S, Cai T, et al. Smart energy management system for optimal microgrid economic operation[J]. IET renewable power generation, 2011, 5(3): 258-267.
3. Hanifi S, Liu X, Lin Z, et al. A critical review of wind power forecasting methods—past, present and future[J]. Energies, 2020, 13(15): 3764.
4. Wang Y, Zou R, Liu F, et al. A review of wind speed and wind power forecasting with deep neural networks[J]. Applied Energy, 2021, 304: 117766.
5. Jung J, Broadwater R P. Current status and future advances for wind speed and power forecasting[J]. Renewable and Sustainable Energy Reviews, 2014, 31: 762-777.
6. Hu J, Heng J, Wen J, et al. Deterministic and probabilistic wind speed forecasting with de-noising-reconstruction strategy and quantile regression based algorithm[J]. Renewable Energy, 2020, 162: 1208-1226.
7. Lydia M, Kumar S S, Selvakumar A I, et al. A comprehensive review on wind turbine power curve modeling techniques[J]. Renewable and Sustainable Energy Reviews, 2014, 30: 452-460.
8. Wang J, Zhou Q, Zhang X. Wind power forecasting based on time series ARMA model[C]//IOP Conference Series: Earth and Environmental Science. IOP Publishing, 2018, 199(2): 022015.
9. Chen P, Pedersen T, Bak-Jensen B, et al. ARIMA-based time series model of stochastic wind power generation[J]. IEEE transactions on power systems, 2009, 25(2): 667-676.
10. Singh S, Bhatti T S, Kothari D P. Wind power estimation using artificial neural network[J]. Journal of Energy Engineering, 2007, 133(1): 46-52.
11. Medsker L R, Jain L C. Recurrent neural networks[J]. Design and Applications, 2001, 5: 64-67.
12. Zhang J, Jiang X, Chen X, et al. Wind power generation prediction based on LSTM[C]//Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence. 2019: 85-89.
13. Kisvari A, Lin Z, Liu X. Wind power forecasting–A data-driven method along with gated recurrent neural network[J]. Renewable Energy, 2021, 163: 1895-1909.
14. Neshat M, Nezhad M M, Abbasnejad E, et al. A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm[J]. Energy Conversion and Management, 2021, 236: 114002.
15. Ding M, Zhou H, Xie H, et al. A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting[J]. Neurocomputing, 2019, 365: 54-61.
16. Yu Y, Han X, Yang M, et al. Probabilistic prediction of regional wind power based on spatiotemporal quantile regression[J]. IEEE Transactions on Industry Applications, 2020, 56(6): 6117-6127.
17. Mezaache H, Bouzgou H. Auto-encoder with neural networks for wind speed forecasting[C]//2018 International Conference on Communications and Electrical Engineering (ICCEE). IEEE, 2018: 1-5.
18. Wei W, Hu X, Liu H, et al. Towards Integration of Domain Knowledge-Guided Feature Engineering and Deep Feature Learning in Surface Electromyography-Based Hand Movement Recognition[J]. Computational Intelligence and Neuroscience, 2021.
19. Wang N, Zhang D, Chang H, et al. Deep learning of subsurface flow via theory-guided neural network[J]. Journal of Hydrology, 2020, 584: 124700.
20. He T, Zhang D. Deep learning of dynamic subsurface flow via theory-guided generative adversarial network[J]. Journal of Hydrology, 2021, 601: 126626.
21. Karpatne A, Atluri G, Faghmous J H, et al. Theory-guided data science: A new paradigm for scientific discovery from data[J]. IEEE Transactions on knowledge and data engineering, 2017, 29(10): 2318-2331.
22. Raissi M, Perdikaris P, Karniadakis G E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. Journal of Computational physics, 2019, 378: 686-707.
23. Li J, Zhang D, Wang N, et al. Deep Learning of Two-Phase Flow in Porous Media via Theory-Guided Neural Networks[J]. SPE Journal, 2022, 27(02): 1176-1194.
24. Chen Y, Huang D, Zhang D, et al. Theory-guided hard constraint projection (HCP): A knowledge-based data-driven scientific machine learning method[J]. Journal of Computational Physics, 2021, 445: 110624.
25. Chen Y, Zhang D. Theory-guided deep-learning for electrical load forecasting (TgDLF) via ensemble long short-term memory[J]. Advances in Applied Energy, 2021, 1: 100004.
26. Huang Y, Liu G P, Hu W. Theory-guided Output Feedback Neural Network (Tg-OFNN) for Short-term Wind Power Forecasting[C]//2021 40th Chinese Control Conference (CCC). IEEE, 2021: 5951-5956.
27. Huang Y, Liu G P, Hu W. Priori-guided and data-driven hybrid model for wind power forecasting[J]. ISA transactions, 2022.
28. Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
29. Karevan Z, Suykens J A K. Transductive LSTM for time-series prediction: An application to weather forecasting[J]. Neural Networks, 2020, 125: 1-9.
30. Eyben F, Weninger F, Squartini S, et al. Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies[C]//2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013: 483-487.
31. Chen Y, Zhang D. Well log generation via ensemble long short‐term memory (EnLSTM) network[J]. Geophysical Research Letters, 2020, 47(23): e2020GL087685.
32. Grosse R. Lecture 15: Exploding and vanishing gradients[J]. University of Toronto Computer Science, 2017.
33. Chakraborty S, Banik J, Addhya S, et al. Study of Dependency on number of LSTM units for Character based Text Generation models[C]//2020 International Conference on Computer Science, Engineering and Applications (ICCSEA). IEEE, 2020: 1-5.
34. Akpinar E K, Akpinar S. An assessment on seasonal analysis of wind energy characteristics and wind turbine characteristics[J]. Energy conversion and management, 2005, 46(11-12): 1848-1867.
35. Panahi D, Deilami S, Masoum M A S. Evaluation of parametric and non-parametric methods for power curve modelling of wind turbines[C]//2015 9th International Conference on Electrical and Electronics Engineering (ELECO). IEEE, 2015: 996-1000.
36. Parzen E. On estimation of a probability density function and mode[J]. The annals of mathematical statistics, 1962, 33(3): 1065-1076.
37. Sheather S J. Density estimation[J]. Statistical science, 2004: 588-597.
38. Scott D W. Multivariate density estimation: theory, practice, and visualization[M]. John Wiley & Sons, 2015.
39. Rudemo M. Empirical choice of histograms and kernel density estimators[J]. Scandinavian Journal of Statistics, 1982: 65-78.
40. Bowman A W. An alternative method of cross-validation for the smoothing of density estimates[J]. Biometrika, 1984, 71(2): 353-360.
41. Scott D W, Terrell G R. Biased and unbiased cross-validation in density estimation[J]. Journal of the american Statistical association, 1987, 82(400): 1131-1146.
42. Sheather S J, Jones M C. A reliable data‐based bandwidth selection method for kernel density estimation[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1991, 53(3): 683-690.
43. Menéndez M L, Pardo J A, Pardo L, et al. The jensen-shannon divergence[J]. Journal of the Franklin Institute, 1997, 334(2): 307-318.
44. Joyce J M. Kullback-leibler divergence[M]//International encyclopedia of statistical science. Springer, Berlin, Heidelberg, 2011: 720-722.
45. Nielsen F. On a generalization of the Jensen–Shannon divergence and the Jensen–Shannon centroid[J]. Entropy, 2020, 22(2): 221.
46. Lobo M G, Sanchez I. Regional wind power forecasting based on smoothing techniques, with application to the Spanish peninsular system[J]. IEEE Transactions on Power Systems, 2012, 27(4): 1990-1997.
47. Shao H, Deng X, Cui F. Short‐term wind speed forecasting using the wavelet decomposition and AdaBoost technique in wind farm of East China[J]. IET Generation, Transmission & Distribution, 2016, 10(11): 2585-2592.