Citation: Zhongda Tian, Xinru Shao. Short-term wind speed prediction based on CEEMDAN-PSO-BiLSTM-attention. *Journal of Harbin Institute of Technology(New Series)*. DOI:10.11916/j.issn.1005-9113.24080

Short-Term Wind Speed Prediction Based on CEEMDAN-PSO-BiLSTM-Attention

Zhongda Tian and Xinru Shao*

(*School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110870, China*)

Abstract: One of the cornerstones for guaranteeing the stability of wind generation and electric power system operation is wind speed prediction. This research offers a method based on Particle Swarm Optimization (PSO) to optimize the Bidirectional Long Short-term Memory Network (BiLSTM) in order to improve the wind speed prediction accuracy, taking into account the highly stochastic and regular aspects of wind speed. Firstly, the wind speed time sequence is subjected to the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN). The complexity of the wind speed pattern is reduced by decomposing it into components with different local feature information. The BiLSTM model, which incorporates the attention mechanism for prediction, is then fitted with the decomposed data, and optimization of BiLSTM parameters using the particle swarm technique, cutting down on mistakes in predictive modeling. To get the final prediction, the components are finally superimposed. The empirical evidence shows that the CEEMDAN-PSO-BiLSTM-attention model decreases the RMSE(root-mean-square-error) by 15%-44%, the MAE by 18%-45%, the MAPE by 24%-52%, and the R^2 by 1.4%-2.7% in comparison to the BiLSTM and other models, which the validation of CEEMDAN-PSO-BiLSTM-Attention model in Short-Term wind speed prediction is verified.

Keywords: short-term wind speed; prediction; Particle Swarm Optimization (PSO); attention mechanism; Bidirectional Short-Term Memory Network(BiLSTM)

CLC number: TM614 **Document code**: A **Article ID**: 1005-9113(2025)00-0000-11

0 Introduction

The Global Wind Energy Council (GWEC) reported a record 117 GW of new wind power capacity installed globally in 2023, marking the highest annual growth to date[1]. Despite abundant wind resources, its unpredictable variability leads to unstable energy output, challenging power grid frequency regulation and voltage control. Exceeding voltage limits threatens grid stability[2], highlighting the critical need for accurate wind speed forecasting in power system management.

There are a number of wind speed prediction algorithms and models, including physical models, artificial intelligence models, statistical models and combined prediction models[3]. Physical models usually make predictions by solving N-S equations from real-time data such as on-site speed and wind direction measurements, and numerical weather prediction; AI models, encompassing both deep learning and supervised studying techniques, such as decision trees and support vector machines, make use of the nonlinear relationship between wind speeds to make models[4], statistical models predict wind speed depending on historical statistical data[5], and the combined prediction model takes the prediction effect of a single model into account and establishes a statistical analysis model depending on the prediction consequences of each model, which improves the combined prediction model.

The combination of combined predictive model is enhanced through the development of statistical analysis frameworks based on individual model outputs[6]. Long Short-Term Memory (LSTM) neural networks[7] were employed in comparative studies, demonstrating superior prediction accuracy over conventional time series methods, though constrained by high computational complexity and extended training durations. To address these limitations,

Received 2024-12-24.

Sponsored by the Science Research Project of Liaoning Education Department (Grant No.LJKZ0143), and the Open Project of State Key Laboratory of Synthetical Automation for Process Industries (Grant No.2023-kfkt-01).

* Corresponding author: Zhongda Tian, Ph.D, Professor. Email: tianzhongda@126.com.

Kisvari et al.[8] presented a Gated Recurrent Unit (GRU)-based architecture, with comparative analyses confirming GRU's advantages in both predictive performance and computational efficiency. While each methodology demonstrates scenario-specific adaptability, achieving optimal predictions with single-model architectures remains challenging. Sabri et al.[9] developed a CNN-GRU hybrid model wherein convolutional layers perform feature extraction to enhance prediction accuracy, while gated recurrent units maintain temporal information storage. Experimental validation confirmed this architecture's superiority over benchmark models, though GRU implementations require intensive hyper-parameter optimization across diverse data types.

Liu et al.[10] put forward a wind power prediction technique based on VMD and LSTM, and discovered that VMD processing greatly increased the wind power forecast accuracy. However, the computing complexity of VMD is high, especially for signals with long time series, the computation time may be substantial. Liu et al.[11] employed empirical modal decomposition (EMD) to decompose the time series and subsequently uses radial basis function (RBF) neural network for prediction. This method reduces the difficulty of prediction and also enhances the accuracy of time series prediction. However, the EMD method is susceptible to module blending or endpoint effects. He et al.[12] combined the improved EEMD (ensemble empirical mode decomposition) with the LASSO-QRNN model to overcome the modal aliasing problem and enhance the prediction accuracy. However, there may be interference noise in the decomposition sequence, which affects the prediction accuracy. To address this problem, Xiong et al.[13] proposed to decompose the wind energy sequence by CEEMD (complete ensemble empirical mode decomposition), which effectively eliminates the interference noise in the decomposed sequence, thus increasing the wind power forecast accuracy. However, the disadvantage of CEEMD is that if the delivered white noise amplitude and the number of iterations are not properly selected, redundant intrinsic mode function (IMF) components are generated, which need to be restructured or processed.

In this study, complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) will be used to deconstruct the original wind velocity sequence. During the decomposition process, the final reconstructed signal has less noise residue than the EEMD result, thus minimising the number of screenings. This is achieved by summing the IMF components of each order resulting from the white noise EMD decomposition. Meanwhile, the parameters of the BiLSTM model introducing the attention mechanism are optimised by the particle swarm algorithm to enhance the forecasting accuracy. Simulation comparisons demonstrate the great prediction accuracy of the method presented in this research.

1 Basic Theory

1.1 CEEMDAN

CEEMDAN algorithm enhances signal decomposition by introducing adaptive white noise to original data, leveraging its properties to differentiate modal components across multiple scales. Initial preprocessing first addresses extreme values or monotonic trends. Adaptive noise sequences are constructed and added to the signal, generating multiple trial sequences. EMD decomposition was then performed on each set of sequences to derive the IMF. These IMFs are aggregated and weighted, then re-decomposed via EMD. The process iteratively verifies convergence criteria: if unmet, adaptive noise is adjusted; if satisfied, iterations continue until IMF quantities stabilize. This method improves decomposition accuracy and stability through signal-adaptive noise modulation, ensuring minimal interference during modal separation[14]. It successfully solves the problem of too many averaging time sets of EEMD algorithm and improves the decomposition efficiency of the approach. The following are the steps in the CEEMDAN decomposition:

1) A new signal sequence is generated by adding white noise $\omega_i(t)$ to the primordial wind speed signal $y(t)$, i.e

$$y_i(t) = y(t) + \varepsilon\omega_i(t) \tag{1}$$

where ε is the amplitude of white noise.

2) The EMD decomposition of the new signal sequence $y_i(t)$ yields the first modal component IMF_1 and the residual $r_1(t)$, i.e

$$\mathrm{IMF}_1(t) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{IMF}_{i1}(t) \tag{2}$$

$$r_1(t) = y(t) - \mathrm{IMF}_1(t) \tag{3}$$

where n is the number of times white noise is added.

3) After adding the white noise $\varepsilon_1 E_1(\omega_i(t))$ to

the residuals $r_1(t)$, EMD decomposition is carried out to obtain the second modal component IMF_2 and residuals $r_2(t)$, i.e

$$IMF_2(t) = \frac{1}{n}\sum_{i=1}^{1}(E_1(r_1(t)) + \varepsilon_1 E_1(\omega_i(t))) \tag{4}$$

$$r_2(t) = r_1(t) - IMF_2(t) \tag{5}$$

4) The raw wind speed signature is decomposed into Eq. (6) and the above steps are repeated for each extra white noise decomposition until the resulting remnants cannot be further decomposed.

$$y(t) = \sum_{j=1}^{J} IMF_j(t) + r_j(t) \tag{6}$$

1.2 BiLSTM with the Introduction of Attention Mechanism

Unidirectional LSTM has low learning efficiency and low prediction accuracy because it can only mine feature information of time sequence data that came into the past and cannot mine information of time sequence data that never came into the past. BiLSTM[15] can not only make use of the information of the meteorological and hydrological data in the past moment, but also learn the features of the time sequence in the future moment by mining the message of the time series data in both directions, which enhances the learning capability of the neural network prediction model and the utilization charge of the time series data, thus improving the precision of prediction. The principle of BiLSTM is shown in Fig. 1.

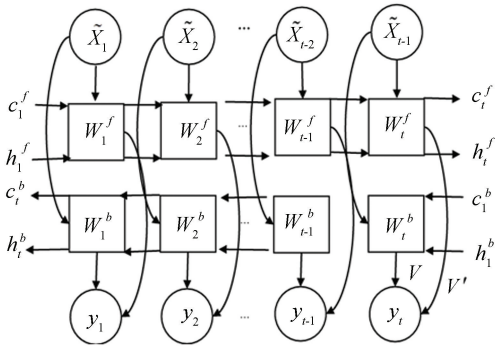

Fig.1 Structure of BiLSTM

The attention mechanism[16] is a way for a model to focus on essential components of the input data. When working with multivariate time series, the model can deal with the most relevant time points by assigning a weight to each input time point. In this way, the model can make accurate predictions even when data are missing or abnormal at certain time points.

Adding an attention layer to the basic BiLSTM network model structure is the fundamental principle behind a BiLSTM model that incorporates an attention mechanism. This makes it possible to sample the input time series data to determine its importance, and then feed the sampled data into the BiLSTM model as input data for training, modeling and prediction. A BiLSTM model with an attention mechanism can handle not only importance-based sampling, but also the long-term dependence of sequences on historical time steps. The BiLSTM in Fig. 1 introduces the attention mechanism by defining an attention layer, where the attention layer weight is denoted by W, $W = (w^1, w^2, \ldots, w^L)$. Through this layer of attention weights W, the BiLSTM that presents the attention mechanism samples the significance of the incoming sequential data $X = (X_1, X_2, \ldots, X_T)$. The sampled data is defined as $\tilde{X}_t$, where $\tilde{X}_t = (x_t^1 w^1, x_t^2 w^2, \ldots, x_t^L w^L)$, after which the importance-sampled data $\tilde{X}_t$ is inserted into the BiLSTM network to obtain the prediction results.

1.3 Particle Swarm Optimization

Some of the parameters in the BiLSTM algorithm affect the prediction accuracy of the model such as: number of neurons, learning rate, batch size, number of training iterations, regularisation parameters[17]. Optimization algorithms can be applied to optimize the above parameters, Particle Swarm Optimization (PSO) can be used to search for the optimal learning rate to speed up training and improve model performance, assist in deciding the proper variety of neurons for most desirable performance, and find the most efficient wide variety of iterations to stability education velocity and performance. PSO can help to find the optimal batch size.

PSO is an evolutionary computational approach based on the feeding behaviour of fowl flocks[18]. The problem of finding food for bird flocks in one-dimensional space is extended to multi-dimensional space to solve. Assuming that the position of a particle in D-dimensional space is denoted as $x(x_{1,D}, x_{2,D}, \ldots, x_{N,D})$, where N is the complete variety of particles; the flight speed of a particle in area is denoted as $v(v_{1,D}, v_{2,D}, \ldots, v_{N,D})$, the best position of an individual particle among all the positions $p_i(p_{1,D}, p_{2,D}, \ldots, p_{i,D})$ it passes through is p_{best}, and the best position among all the positions $p_g(p_{1,D}, \ldots, p_{2,D}, \ldots, p_{g,D})$ it passes through is g_{best}. The updating formulas for the flight speed and position of a particle can be

expressed as follows:

$$v_{N,D} = a \cdot v_{N,D} + c_1 \cdot \mathrm{rand}(p_{i,D} - x_{N,D}) + c_2 \cdot \mathrm{rand}(p_{g,D} - x_{N,D}) \quad (7)$$

$$x_{N,D} = x_{N,D} + v_{N,D} \quad (8)$$

where a denotes the inertia coefficient, which takes a non-negative value; c_1 and c_2 denote the learning factor; and rand is a random generating function between [0, 1].

The PSO optimization search process is shown below:

Step 1: Initialize the velocity and role of the population, set the number of population size N and the variety of iterations M, and set the historic exceptional function p_{best} of an person as the contemporary position, and the most efficient character in the populace as the contemporary g_{best}.

Step 2: At each evolutionary stage, the health feature for every particle is derived.

Step 3: p_{best} is up to date when the cutting-edge fitness characteristic cost is higher than the historic optimum.

Step 4: When the contemporary health characteristic price is higher than the historic optimum, g_{best} is updated.

Step 5: The position and flight velocity of the particles are up to date according to the above Eqs. (7) and (8).

Step 6: Repeat Steps 2 to 5 to continue searching for the global optimum. Stop iteration after finding the global optimum location or reaching the maximum number of iterations.

2 Designed Model

2.1 Prediction Model Construction

The flow diagram of the prediction model is shown in Fig. 2. The specific prediction steps can be described as follows:

Step 1: The unique wind velocity sequence is decomposed into IMF factors with different local feature information, and the IMF factors are smoothed by CEEMDAN decomposition;

Step 2: The normalized IMF components were modeled using BiLSTM with the introduction of an attention mechanism, where the top 80% of the data was used for training and the bottom 20% for prediction;

Step 3: Parameters such as learning rate, number of hidden neurons and batch size are optimized in BiLSTM model using PSO algorithm;

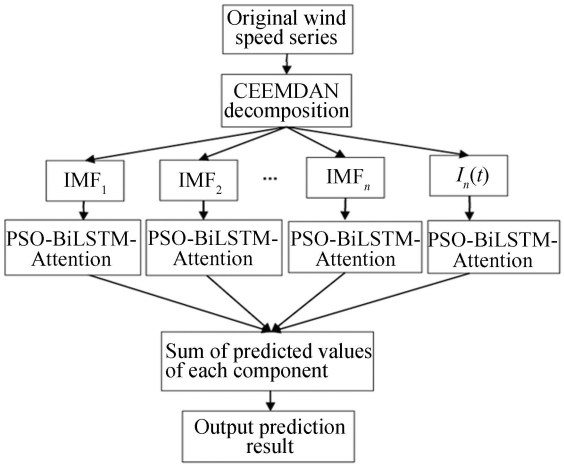

Fig. 2 Flowchart of wind speed prediction model

Step 4: Predictions are made for each component using the optimized BiLSTM model, and then all factors are summed and inverted to normalize to acquire the ultimate prediction.

2.2 Hyper-parameter Optimization

The prediction accuracy of BiLSTM models is closely related to the values of hyper-parameters. At present, the values of hyper-parameters are mostly based on experience, and sometimes even through multiple trials to achieve better prediction accuracy. In this paper, particle swarm optimization BiLSTM model hyper-parameters are cited. Greff[19] pointed out that the learning rate and the number of hidden neurons are the two hyper-parameters that have the best influence on the prediction performance of LSTM. Therefore, in this paper, the learning rate, the number of hidden neurons and the batch size are used as the search parameters for particle swarm optimization. The process is shown in Fig. 3.

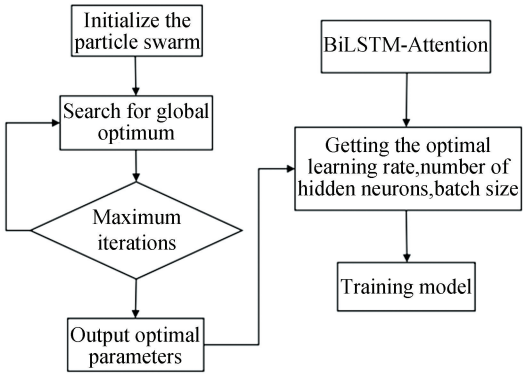

Fig. 3 PSO optimize BiLSTM-Attention flowchart

3 Case Studies

In this article, Short-Term wind velocity data from wind turbine Scada in Turkey in 2018 are selected to constitute datasets *A* and *B*. In dataset *A*, the collection period is from 00∶00 on 1 February 2018 to 23∶50 on 28 February 2018, the total number of samples is 4032, and the sampling period is 10 min; In dataset *B*, the collection period is from 00∶00 on 1 August 2018 to 23∶30 on 31 August 2018, with a total number of samples of 1488 and a sampling period of 30 min. The top 80% of the samples are taken as the training set and the bottom 20% of the samples are taken as the test set, respectively. The original wind speed sequences are shown in Figs. 4 and 5.

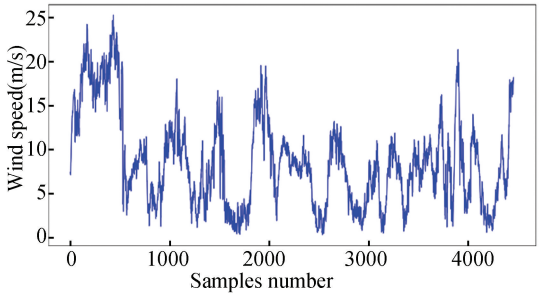

Fig.4 Original wind speed series of dataset *A*

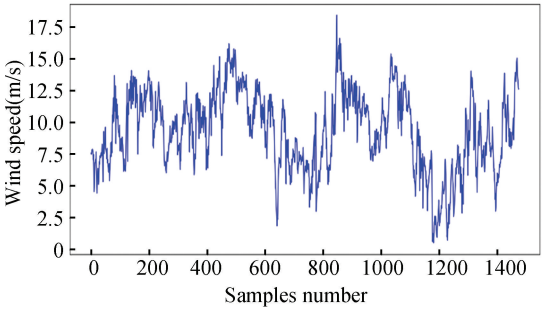

Fig. 5 Original wind speed series of dataset *B*

From Figs. 4 and 5, it can be observed that the amplitude of the data fluctuates a lot, so it is necessary to normalize the wind speed series shown in Eq. (9) to increase the precision of the wind speed forecasting model.

$$x^* = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \tag{9}$$

where x^* is the wind velocity value after data normalization, x is the primordial wind velocity value, $X_{\min}$ is the minimum wind speed data in the original wind velocity data, and $X_{\max}$ is the maximum wind velocity data in the primordial wind velocity data.

The wind velocity data are first decomposed by CEEMDAN, which adaptively adds white noise of a specific intensity at each decomposition stage to extract the IMF through multiple ensemble averaging. The residual term is updated after each decomposition as an input to the next stage until the residual becomes a monotonic trend term. Taking dataset *A* as an example, Fig. 6 shows the wind speed sequence of dataset *A* after CEEMDAN decomposition. The primordial wind speed sequence is collapsed into nine IMF components (IMF_0-IMF_8) and one residual term (IMF_9). As can be seen from Fig. 6, the image tends to be more smooth curves as the number of decompositions increases. Components IMF_1-IMF_5 have higher frequency and high volatility, and components IMF_6-IMF_9 have overall lower frequency and smooth. Therefore CEEMDAN decomposition improves the decomposition efficiency and reduces the error for subsequent wind speed prediction.

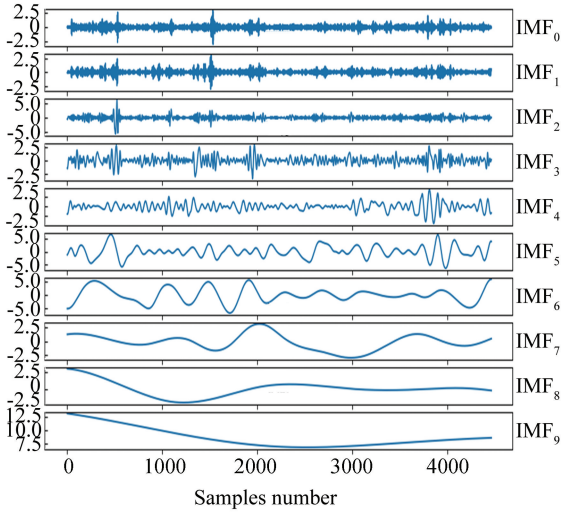

Fig. 6 CEEMDAN decomposition (dataset *A*)

Afterwards, the corresponding BiLSTM-Attention forecasting model is built for each IMF component and the BiLSTM parameters are optimized using PSO. The particle swarm optimization algorithm is calculated according to the above parameter settings and constraints, and the specific parameters are shown in Table 1. The optimal hyper-parameters of each subsequence obtained after optimization are shown in Tables 2 and 3. Finally, the predicted values for each wind speed component are totaled to give the final prediction.

For a better comparison of predictive effects, this

paper adopts root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and coefficient of determination (R^2) as the evaluation indexes, which are calculated by the following formulas:

Table 1 PSO parameter setting

Population size N	Maximum number of iterations M	Parameter upper limit	Parameter lower limit	Learning factor, social factor	Random factor	Inertia weight a
25	**30**	[0.01,100,128]	[0.001,10,64]	$c_1=1.5$, $c_2=1.7$	$r_1=0.8$, $r_2=0.3$	0.5

Table 2 Optimal hyper-parameters of each subsequence model of dataset *A*

Subsequence	Learning rate	Number of hidden neurons	Batch size
IMF_1	0.00974	49	86
IMF_2	0.00816	82	102
IMF_3	0.00770	64	92
IMF_4	0.00713	70	93
IMF_5	0.00810	25	105
IMF_6	0.00986	54	95
IMF_7	0.00695	69	86
IMF_8	0.00558	54	107
IMF_9	0.00492	74	78

Table 3 Optimal hyper-parameters of each subsequence model of dataset *B*

Subsequence	Learning rate	Number of hidden neurons	Batch size
IMF_1	0.00186	74	105
IMF_2	0.00942	63	94
IMF_3	0.00754	23	80
IMF_4	0.00778	46	95
IMF_5	0.00803	70	87
IMF_6	0.00875	14	89
IMF_7	0.00838	35	96
IMF_8	0.00555	59	87

RMSE(root mean square error):

$$\mathrm{RMSE}=\sqrt{\frac{1}{k}\sum_{i=1}^{k}(y_i-\hat{y}_i)^2} \tag{10}$$

MAE(mean absolute error):

$$\mathrm{MAE}=\frac{1}{k}\sum_{i=1}^{k}|y_i-\hat{y}_i| \tag{11}$$

MAPE(mean absolute percentage error):

$$\mathrm{MAPE}=\frac{1}{k}\sum_{i=1}^{k}\frac{|y_i-\hat{y}_i|}{y_i}\times 100\% \tag{12}$$

R^2(coefficient of determination):

$$R^2=1-\frac{\sum_{i=1}^{k}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{k}(y_i-\bar{y})^2} \tag{13}$$

where y_i is true value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean value, and k is the sample size used for training or testing.

3.1 Comparison with Combination Models

BiLSTM model, CEEMDAN-BiLSTM model, CEEMDAN-BiLSTM-Attention model, and CEEMDAN-PSO-BiLSTM-Attention model are respectively built to compare with the original wind speed to verify their effectiveness and accuracy, as shown in Figs.7-8. Among them, the BiLSTM hidden neurons are set to 32 and the batch size is 64. CEEMDAN divides the raw wind speed sequence into nine IMF components and a residual term.

Comparing the actual and predicted value curves of models in Fig. 7 and Fig. 8, it can be observed that

the expected values of each model are nearer to the actual values, but the estimated profile of the CEEMDAN-PSO-BiLSTM-attention model has the largest overlap with the actual value curve.

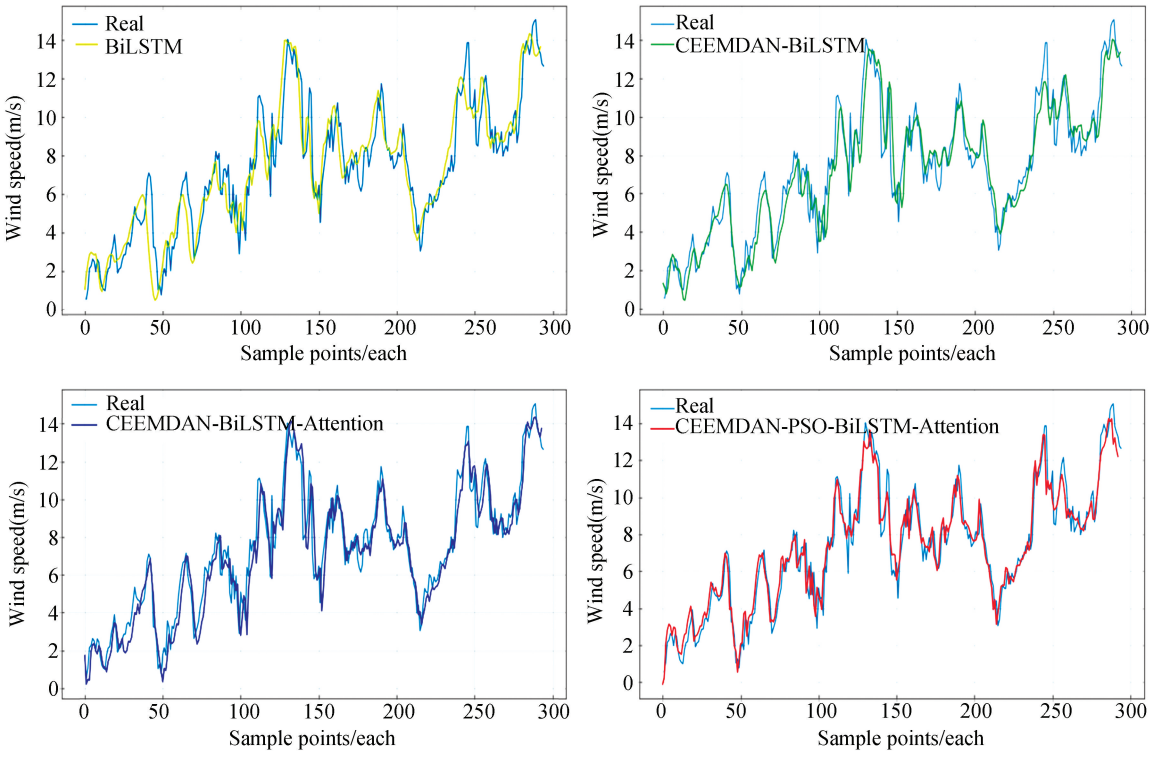

Fig.7 Comparison of predicted and actual wind speeds of dataset *A*

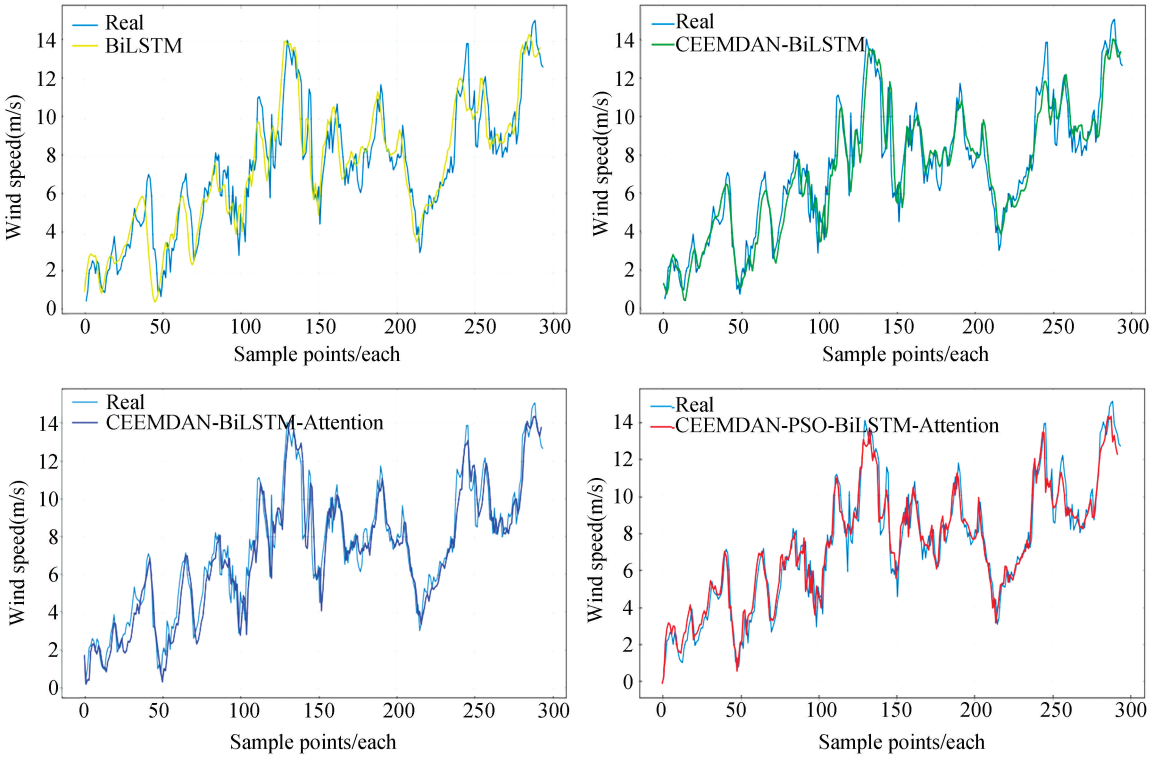

Fig. 8 Comparison of predicted and actual wind speeds of dataset *B*

Tables 4 and 5 show the comparison of the performance metrics such as RMSE, MAE, MAPE, and R^2 of several prediction methods in dataset *A* and dataset *B* respectively. As shown in Table 4, the CEEMDAN-BiLSTM-Attention model reduces the RMSE, MAE, and MAPE error metrics by 24.2%, 27.9%, and 20.8%, respectively, with respect to the CEEMDAN-BiLSTM model, and the R^2 improves by 0.62%. The CEEMDAN-PSO-BiLSTM-attention model also reduces the RMSE, MAE, and MAPE error metrics by 15.5%, 18.0%, and 24.1%, respectively, relative to the CEEMDAN-BiLSTM-Attention model. R^2 improves by 1.41%.

Table 5 similarly demonstrates that the CEEMDAN-PSO-BiLSTM-attention model outperforms other models. In conclusion, the prediction approach in this paper is superior to other prediction methods in terms of performance metrics of error.

Table 4 Comparison of Performance Metrics of Dataset *A*

Forecasting method	Training time (s)	Evaluation indicators			
		RMSE(m/s)	MAE(m/s)	MAPE(%)	R^2
BiLSTM	61.57	0.871	0.630	5.779	0.961
CEEMDAN-BiLSTM	127.89	0.764	0.583	4.860	0.967
CEEMDAN-BiLSTM-Attention	187.35	0.579	0.420	3.849	0.973
CEEMDAN-PSO-BiLSTM-Attention	10706.31	0.489	0.344	2.920	0.987

Table 5 Comparison of Performance Metrics of Dataset *B*

Forecasting method	Training time (s)	Evaluation indicators			
		RMSE(m/s)	MAE(m/s)	MAPE(%)	R^2
BiLSTM	16.52	1.036	0.776	8.649	0.902
CEEMDAN-BiLSTM	31.40	0.820	0.636	6.045	0.951
CEEMDAN-BiLSTM-Attention	173.95	0.716	0.523	5.437	0.953
CEEMDAN-PSO-BiLSTM-Attention	9629.68	0.555	0.412	3.932	0.971

3.2 Comparison with Classical Models

To equally illustrate the excellence of the model in this study, the classical model is choosen for comparison experiments. The number of 1D convolutions of CNN is set to be 64, the size of convolution kernel is 3×3, and the number of GRU implied neurons is 32. EMD decomposes the primordial wind speed data into 8 IMF components and a residual term, and the performance indexes of the several prediction approaches for dataset *A* and dataset *B* are given in Tables 6 and 7 for the comparison, respectively. As shown in Table 6, the RMSE, MAE, and MAPE error metrics of this paper's model are reduced by 44.1%, 44.9%, and 45.4%, respectively, in contrast with those of the CNN-GRU model, and the R^2 improves by 2.6%; and the RMSE, MAE, MAPE error metrics of this paper's model are reduced by 40.7%, 43.1%, and 52.5%, respectively, in contrast with those of the EMD-LSTM model, and the R^2 increases by 2.3%. Table 7 additionally proves that the overall performance index of this model is higher than different models.

Table 6 Comparison of classical model performance metrics of dataset *A*

Forecasting method	Training time (s)	Evaluation indicators			
		RMSE(m/s)	MAE(m/s)	MAPE(%)	R^2
CNN-GRU	5.36	0.876	0.625	5.352	0.961
EMD-LSTM	107.67	0.826	0.605	6.156	0.964
CEEMDAN-PSO-BiLSTM-attention	10706.31	0.489	0.344	2.920	0.987

Table 7 Comparison of classical model performance metrics of dataset *B*

Forecasting method	Training time (s)	Evaluation indicators			
		RMSE(m/s)	MAE(m/s)	MAPE(%)	R^2
CNN-GRU	2.95	1.034	0.769	8.305	0.902
EMD-LSTM	23.47	0.904	0.683	6.785	0.923
CEEMDAN-PSO-BiLSTM-attention	9629.68	0.555	0.412	3.932	0.971

3.3 Multi-step Prediction

Multi-step forecasting is an positive approach to check the precision of forecast models [20]. In many forecasting applications, it is necessary to predict the trend in the coming period, for example, multi-step forecasting of wind speed is of great value in wind farm applications. For example, multi-step wind speed prediction with large step length can provide more abundant time for grid adjustment in wind farms. Therefore, the CEEMDAN-BiLSTM-Attention model and the model in this paper are selected to perform three-step, five-step and ten-step forecasting experiments, respectively. The multi-step forecasting results of dataset *A* and dataset *B* are shown in Figs. 9 and 10, respectively.

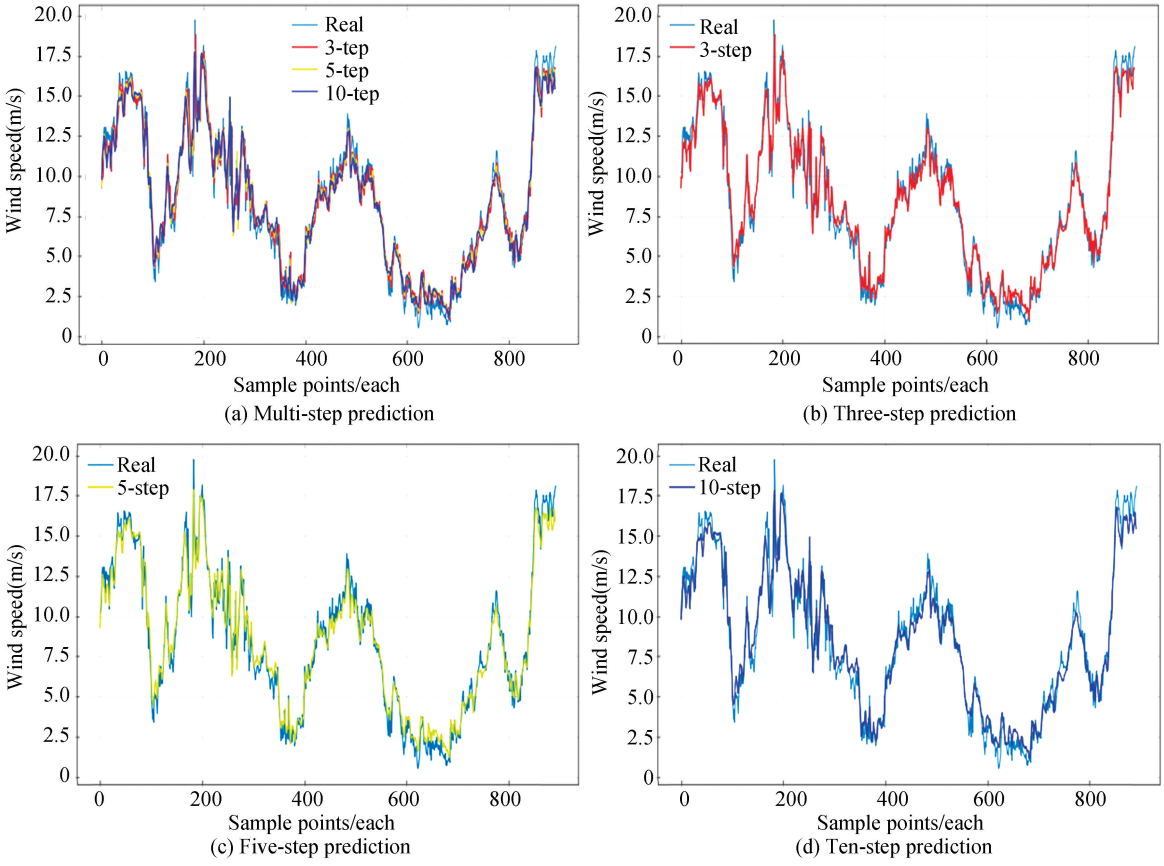

Fig. 9 Multi-step prediction results of dataset *A*

The experimental consequences are shown in Tables 8 – 9. The effects exhibit that the forecasting accuracy of multi-step forecasting is lower than that of single-step prediction. The fundamental motive is that the accuracy of multi-step forecasting is affected by the step length, and the forecasting of each step depends on the forecasting result of the previous step, even if the prediction of the first few steps is relatively accurate, the error may gradually accumulate with the passage of time, resulting in a larger overall prediction error. Therefore, the prediction results of three, five and ten steps gradually increase. The model proposed in this article has higher performance metrics than other models.

4 Conclusions

In this article, the CEEMDAN decomposition approach is adopted to process the fluctuating wind speed data, which is decomposed into smooth and regular signals to lessen the wind speed sequence's unpredictability and complexity. The decomposed subsequence is then modeled and predicted using BiLSTM with the introduction of an attention mechanism, which not only captures the long-term dependence on the historical time step in the sequence, but also handles importance-based sampling. In addition, the PSO algorithm is adopted

to find the optimal key parameters of BiLSTM, which decreases errors in prediction model and improves the prediction precision. The main observations drawn are as follows:

1) By contrast to the ablation experiments in Section 3.1, the CEEMDAN-PSO-BiLSTM-Attention model successfully reduces RMSE by 15% to 46%, MAE by 18% to 47% and MAPE by 24% to 55%, and improves R^2 by 1.4% to 7.6%.

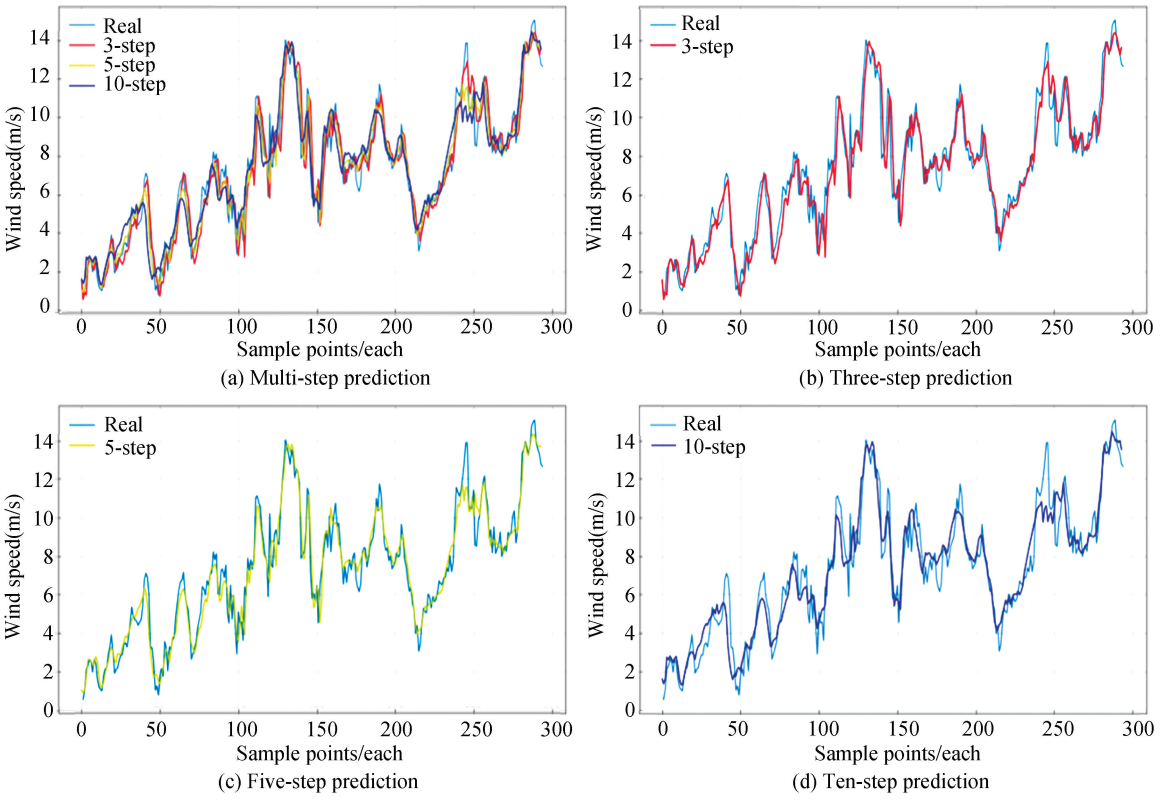

Fig.10 Multi-step prediction results of dataset *B*

Table 8 Comparison of different predicted steps performance metrics of dataset *A*

Predicted steps	Evaluation indicators	RMSE(m/s)	MAE(m/s)	MAPE(%)	R^2
Three-step forecast	CEEMDAN-BiLSTM-Attention	1.115	0.809	7.104	0.935
	CEEMDAN-PSO-BiLSTM-attention	0.977	0.703	6.686	0.951
Five-step forecast	CEEMDAN-BiLSTM-Attention	1.375	1.003	8.956	0.901
	CEEMDAN-PSO-BiLSTM-attention	1.200	0.896	8.504	0.925
Ten-step forecast	CEEMDAN-BiLSTM-Attention	1.735	1.289	11.597	0.844
	CEEMDAN-PSO-BiLSTM-attention	1.506	1.096	10.856	0.883

Table 9 Comparison of different predicted steps performance metrics of dataset *B*

Predicted steps	Evaluation indicators	RMSE(m/s)	MAE(m/s)	MAPE(%)	R^2
Three-step forecast	CEEMDAN-BiLSTM-Attention	1.215	0.917	10.400	0.859
	CEEMDAN-PSO-BiLSTM-attention	1.044	0.800	7.582	0.912
Five-step forecast	CEEMDAN-BiLSTM-Attention	1.436	1.112	11.788	0.806
	CEEMDAN-PSO-BiLSTM-Attention	1.278	0.992	9.358	0.846
Ten-step forecast	CEEMDAN-BiLSTM-Attention	1.732	1.393	15.295	0.715
	CEEMDAN-PSO-BiLSTM-Attention	1.685	1.339	12.373	0.788

2) Compared with the traditional CNN-GRU and EMD-LSTM models, the CEEMDAN-PSO-BiLSTM-Attention model reduces the prediction error and exhibits better performance enhancement in Short-Term wind velocity prediction. Specifically, the combined model successfully reduces the RMSE by 38% to 47%, MAE by 39% to 46% and MAPE by 42% to 53%, and improves the R^2 by 2.3% to 7.6%.

However, there are some problems with the current BiLSTM-Attention model, such as the computation of the attention weights may be affected by the length of the input sequences, leading to overfitting or underfitting. These issues can be further explored in future research.

References

[1] Joyce L, Feng Z. Global Wind Report 2024. Lisbon: Global Wind Energy Council, 2024.

[2] Mastoi S M, Zhuang S X, Haris M, et al. Large-scale wind power grid integration challenges and their solution: a detailed review. Environmental Science and Pollution Research, 2023, 30: 103424 - 103462. DOI: 10.1007/S11356 - 023-29653-9.

[3] Tian C S, Hao Y, Hu J M. A novel wind speed forecasting system based on hybrid data preprocessing and multi-objective optimization. Applied Energy, 2018, 231: 301 - 319. DOI: 10.1016/j.apenergy.2018.09.012.

[4] Xu Y Y, Yao T H, Yang G K. An EMD-SVM model with error compensation for Short-Term wind speed forecasting. International Journal of Information Technology and Management, 2019, 18(2/3): 171 - 181. DOI: 10.1504/IJITM.2019.099827.

[5] Ye L, Zhao Y N, Zeng C, et al. Short-term wind power prediction based on spatial model. Renewable Energy, 2017, 101: 1067-1074. DOI: 10.1016/j.renene.2016.09.069.

[6] Wang T L, Luo R, Ma T X, et al. Study and verification on an improved comprehensive prediction model of landslide displacement. Bulletin of Engineering Geology and the Environment, 2024, 83(3): 90. DOI: 10.1007/s10064-024-03581-5.

[7] Huang F, Li Z X, Xiang S C, et al. A new wind power forecasting algorithm based on long short-term memory neural network. International Transactions on Electrical Energy Systems, 2021, 31(12): e13233. DOI: 10.1002/2050-7038.13233.

[8] Kisvari A, Lin Z, Liu X L. Wind power forecasting-A data-driven method along with gated recurrent neural network. Renewable Energy, 2021, 163: 1895-1909. DOI: 10.1016/j.renene.2020.10.119.

[9] Sabri M, Hassouni M E. Accurate photovoltaic power prediction models based on deep convolutional neural networks and gated recurrent units. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 2022, 44(3): 6303 - 6320. DOI: 10.1080/15567036.2022.2097751.

[10] Liu X D, Zhang L, Zhang Z R, et al. Ultra-short-term wind power prediction model based on VMD decomposition and LSTM. IOP Conference Series: Earth and Environmental Science, 2021, 838(1): 012002. DOI: 10.1088/1755-1315/838/1/012002.

[11] Liu Y P, Wang Y, Wang Z. RBF prediction model based on EMD for forecasting GPS precipitable water vapor and annual precipitation. Advanced Materials Research, 2013, 765-767: 2830-2834. DOI: 10.4028/www.scientific.net/AMR.765-767.2830.

[12] He Y, Wang Y. Short-term wind power prediction based on EEMD-LASSO-QRNN model. Applied Soft Computing, 2021, 105: 107288. DOI: 10.1016/j.asoc.2021.107288.

[13] Xiong J L, Peng T, Tao Z H, et al. A dual-scale deep learning model based on ELM-BiLSTM and improved reptile search algorithm for wind power prediction. Energy, 2023, 266: 126419. DOI: 10.1016/j.energy.2022.126419.

[14] Zhang C Y, Wang S L, Yu C M, et al. A complete ensemble empirical mode decomposition with adaptive noise deep autoregressive recurrent neural network method for the whole life remaining useful life prediction of lithium-ion batteries. Ionics, 2023, 29(10): 4337-4349. DOI: 10.1007/S11581-023-05152-2.

[15] Zrira N, Kamal-Idrissi A, Farssi R, et al. Time series prediction of sea surface temperature based on BiLSTM model with attention mechanism. Journal of Sea Research, 2024, 198: 102472. DOI: 10.1016/J.SEARES.2024.102472.

[16] Gao Z, Kuruoğlu E E. Attention based hybrid parametric and neural network models for non-stationary time series prediction. Expert Systems, 2023, 41(2): e13419. DOI: 10.1111/EXSY.13419.

[17] Li J, Song Z, Wang X, et al. A novel offshore wind farm typhoon wind speed prediction model based on PSO-Bi-LSTM improved by VMD. Energy, 2022, 251: 123848. DOI: 10.1016/J.ENERGY.2022.123848.

[18] Jain N K, Nangia U, Jain J. A review of particle swarm optimization. Journal of The Institution of Engineers (India): Series B, 2018, 99: 407 - 411. DOI: 10.1007/s40031-018-0323-y.

[19] Greff K, Srivastava R K, Koutník J, et al. LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2222 - 2232. DOI: 10.1109/TNNLS.2016.2582924.

[20] Mirjalili S, Gandomi A H, Mirjalili S Z, et al. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 2017, 114: 163-191. DOI: 10.1016/j.advengsoft.2017.07.002.