An Improved Hybrid CNN-LSTM-Attention Model with Kepler Optimization Algorithm for Wind Speed Prediction

Yuesheng Huang, *Member*, *IAENG*, Jiawen Li, *Member*, *IAENG*, Yushan Li, Routing Lin, Jingru Wu, Leijun Wang, and Rongjun Chen

***Abstract*—The flexibility and intermittency of wind speed is a crucial factor for its prediction. Research in this domain necessitates integrating diverse data to enhance model precision and reliability to meet the growing demand for wind energy in both the energy sector and the global economy. To this end, this work adopts the Kepler optimization algorithm (KOA), which utilizes the position of each planet as a candidate solution. The optimization process then involves random updates to these candidate solutions, aiming to enhance wind speed prediction using the CNN-LSTM-Attention model. Specifically, it enhances the interpretability of the KOA for the global optimization problem. By using 1800 time moments of wind speed data with multidimensional features, the proposed hybrid model with KOA has been shown to outperform other optimization methods, demonstrating impressive performances in terms of mean absolute percentage error (MAPE), mean absolute error (MAE), runtime, root mean square error (RMSE), and R^2. Hence, these results indicate its potential for improving wind speed prediction, which could open a novel direction in this field.**

***Index Terms*—Deep learning, Kepler optimization algorithm, CNN-LSTM-Attention, Hybrid model, Wind speed prediction**

Manuscript received March 17, 2024; revised September 7, 2024. This work was supported in part by the Special Projects in Key Fields of Ordinary Universities of Guangdong Province under Grant 2021ZDZX1087, in part by the Guangzhou Science and Technology Plan Project under Grants 2024B03J1361, 2023B03J1327, and 2023A04J0361, in part by the Guangdong Province Ordinary Colleges and Universities Young Innovative Talents Project under Grant 2023KQNCX036, in part by the Special Fund for Science and Technology Innovation Strategy of Guangdong Province (Climbing Plan) under Grant pdjh2024a226, in part by the Key Discipline Improvement Project of Guangdong Province under Grant 2022ZDJS015, and in part by the Research Fund of Guangdong Polytechnic Normal University under Grant 2022SDKYA015.

Yuesheng Huang is an undergraduate student of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China (e-mail: yorksonhuang@gmail.com).

Jiawen Li is an assistant professor of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China. (corresponding author, phone:+86-020-38256730; fax: +86-020-38257901; e-mail: lijiawen@gpnu.edu.cn).

Yushan Li is an undergraduate student of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China (e-mail: leannlee133@gmail.com).

Routing Lin is an undergraduate student of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China (e-mail: linrouting96@gmail.com).

Jingru Wu is an undergraduate student of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China (e-mail: JingRuWu1023@gmail.com).

Leijun Wang is an assistant professor of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China (e-mail: wangleijun@gpnu.edu.cn).

Rongjun Chen is a professor of School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, 510665 China (e-mail: chenrongjun@gpnu.edu.cn).

I. Introduction

WITH the rapid development of the global economy, there is a growing need for solutions to address issues like energy shortages and environmental pollution [1]. Shifting from traditional energy structures and enhancing the utilization of renewable energy have become primary objectives for numerous countries. Wind energy is a renewable source that holds the potential for extensive development [2]. Therefore, the expansion of wind energy significantly contributes to addressing challenges in energy structure, managing atmospheric haze, and transforming modes of economic development [3–5]. Achieving high-precision wind speed prediction is essential for optimizing the utilization of wind power. Nevertheless, accurate prediction continues to be a challenge due to the flexibility and intermittency of wind speeds [6]. In this regard, synthesizing diverse data is vital to enhancing the accuracy and reliability of models, which is crucial for addressing the escalating demand for wind energy in both the energy sector and the global economy.

Time series forecasting (TSF) is a common approach to obtaining accurate predictions in real-time applications, and it is becoming increasingly popular in many areas of daily life [7]. It plays a key role in problems with a temporal characteristic, as forecasting the future through time series is beneficial in numerous areas, such as financial indices, weather, medical monitoring, and energy consumption [8], highlighting TSF's relevance for wind speed prediction. For example, Wu et al. [9] combined time fusion transformers and multivariate time series to predict wind speed. Yu et al. [10] utilized the time series cross-correlation network for wind speed prediction. Chen et al. [11] used multifunctional interpolation techniques on large-scale wind speed time series to predict offshore wind speed. Besides, heuristic optimization algorithms are extensively used in TSF due to their high predictive performance and ability to effectively reduce computational complexity. For instance, Chen and Zhang [12] introduced the gray wolf optimization algorithm to optimize the deep echo state model by minimizing learning errors. Kuranga et al. [13] proposed a model optimized by a multicluster particle swarm to improve prediction accuracy. Shen et al. [14] developed a hybrid dynamic ensemble pruning framework (DEPF) by combining heuristic optimization algorithms and the meta-learning paradigm to enhance prediction accuracy.

Furthermore, prediction methods are generally categorized into physical, statistical, neural network, and hybrid approaches [15–17]. For instance, artificial neural network (ANN) is a commonly used method for wind speed prediction, and approaches that incorporate statistical algorithms with ANNs have garnered significant attention [18]. Alternatively, by utilizing the automatic spatial feature extraction of convolutional neural network (CNN) and the robust capabilities of long short-term memory (LSTM) to obtain short-term and long-term temporal components, the CNN-LSTM model is a suitable solution for handling diverse prediction horizons. This hybrid model extracts features, processes variable-length sequences, and reduces computation and training time to achieve strong performance in wind speed prediction [19]. Meanwhile, an attention mechanism can be included [20], which has the key capability to process vital information by using limited resources and suppressing irrelevant data. Its essence lies in the allocation of input weights to prioritize important features. For example, Ren et al. [21] demonstrated that including the attention mechanism allows the assignment of distinct weights to feature vectors in the hidden layer, identifying key features during the training stage and ultimately enhancing prediction accuracy.

Recently, Abdel-Basset et al. [22] proposed the Kepler optimization algorithm (KOA), a population-based meta-heuristic technique designed for solving continuous optimization problems. It has achieved significant advances in tackling a diverse set of continuous optimization problems by efficiently exploring the solution space and converging toward near-optimal solutions. Motivated by the success of KOA in optimizing various tasks, we employ it to fine-tune the parameters of the CNN-LSTM-Attention model specifically tailored for wind speed prediction tasks. This technique has thus far been applied only to weather prediction [23] and has not yet been employed in wind speed prediction. By optimizing the model parameters using the KOA, we aim to achieve enhanced prediction accuracy and robustness in capturing the intricate spatiotemporal dynamics inherent in wind speed data. Therefore, this proposed hybrid model not only builds upon the advancements in deep learning methods for wind speed prediction but also further enhances model performance and applicability in TSF scenarios by integrating appropriate optimization algorithms, potentially opening a novel direction in this field.

The rest of this paper is organized as follows: Section II introduces the CNN-LSTM-Attention model with the KOA. Section III describes the experiment conducted for validating the method. Section IV presents the comprehensive results and evaluations. Finally, Section V presents the conclusion.

II. Proposed Methodology

A. Kepler Optimization Algorithm (KOA)

In the KOA, each planet and its position represent a candidate solution. The optimization process involves random updates to candidate solutions based on the best solution so far, enhancing the exploration and utilization of the search space. This process primarily involves initializing planetary positions and velocities, evaluating each planet using the fitness function, renewing the velocity and position of each planet, updating the position of the optimal solution, and iteratively repeating these steps until the stopping condition is met. These steps enable KOA to systematically explore and utilize the search space throughout the optimization process. Therefore, we include it to optimize hyperparameters within the CNN-LSTM-Attention model, enabling a reduction in the error of wind speed prediction. The details of KOA are described below:

The first stage is the initialization process, where N planets are distributed randomly in a d-dimensional space, forming a set known as the population. The set denotes the decision variables of the optimization problem:

$$X_i^j = X_{i,low}^j + rand_{[0,1]} \times \left(X_{i,up}^j - X_{i,low}^j \right), \begin{cases} i = 1,2,\ldots,N. \\ j = 1,2,\ldots,d. \end{cases} \quad (1)$$

where X_i is the *i-th* candidate solution in the search space. N denotes the total number of candidate solutions in the search space. d means the dimensionality of the problem to be optimized. $X_{i,up}^j$ and $X_{i,low}^j$ refer to the upper and lower bounds of the *j-th* decision variable, respectively, and $rand_{[0,\,1]}$ is a randomly generated number between 0 and 1. It is also used for the orbital eccentricity e of the i-th object, and the orbital period T_i is produced using normally distributed random numbers. By combining these variables, a series of candidate solutions with potential optimization values can be acquired. Then, the gravitational force between the X_S and the X_i is:

$$F_{g_i}(t) = e_i \times \mu(t) \times \frac{\overline{M}_s \times \overline{m}_i}{\overline{R}_i^2 + \varepsilon} + r_1 \quad (2)$$

where $\overline{M}_s$ and $\overline{m}_i$ are the normalized planetary masses of X_s and X_i, respectively, and ε denotes a small value used to control the range of gravitational influence. μ represents the universal gravitational constant, which is the fundamental constant of gravitational interaction between objects. e_i is the eccentricity of the planetary orbits, which takes a range of values between 0 and 1 and is used to introduce a certain stochastic character to the KOA. r_1 shows the Euclidean distance between X_S and X_i and is normalized to obtain the value of $\overline{R}_i^2$, a measure of the distance between planets.

Mathematically, the problem of calculating the speed of an object orbiting the Sun can be represented using a double equation. Then, to enhance the diversity of the search, the distance between a randomly chosen solution and the current solution can be utilized as a measure and incorporated into the velocity calculation. To prevent convergence to local minima, a step size determined by the distance between the lower and upper bounds of the optimization problem can be applied, ensuring the stability of planetary velocities [24]. Similarly, the distance between solutions can be employed as a measure and integrated into the velocity calculation. Moreover, to enhance the diversity of solutions, a second step size can be included, broadening the exploitability of the search space by reducing the magnitude of the velocity variation. The objective of these strategies is to sustain planetary velocity throughout the optimization process and minimize the risk of converging to local minima. Thus, by incorporating the distance between solutions and introducing varying step sizes, the diversity of the search can be enhanced:

$$V_{\mathrm{i}}(t) = \begin{cases} \ell \times \left(2r_4 \vec{\mathrm{X}}_i - \vec{\mathrm{X}}_b - \right) + \ddot{\ell} \times \left(\vec{\mathrm{X}}_a - \vec{\mathrm{X}}_b \right) + \left(1 - R_{i-norm}(t) \right) \\ \quad \times \mathcal{F} \times \vec{U}_1 \times \vec{r}_s \times \left(\vec{\mathrm{X}}_{i,up} - \vec{\mathrm{X}}_{i,low} \right), \text{if } R_{i-norm}(t) \le 0.5 \\ r_4 \times \mathcal{L} \times \left(\vec{\mathrm{X}}_a - \vec{\mathrm{X}}_i \right) + \left(1 - R_{i-norm}(t) \right) \\ \quad \times \mathcal{F} \times U_2 \times \vec{r}_s \times \left(r_3 \vec{\mathrm{X}}_{i,up} - \vec{\mathrm{X}}_{i,low} \right), \text{else} \end{cases} \quad (3)$$

$$\ell = \vec{U} \times \mathcal{M} \times \mathcal{L} \tag{4}$$

$$\mathcal{L} = \left[\mu(t) \times (M_S + m_i) \left| \frac{2}{R_i(t) + \varepsilon} - \frac{1}{a_{i(t)} + \varepsilon} \right| \right]^{\frac{1}{2}} \tag{5}$$

$$M = (r \times (1 - r) + r) \tag{6}$$

$$\vec{U} = \begin{cases} 0, & \vec{r}_5 \le \vec{r}_6 \\ 1, & \text{else} \end{cases} \tag{7}$$

$$F = \begin{cases} 1, & \text{if } r_4 \le 0.5 \\ -1, & \text{else} \end{cases} \tag{8}$$

$$\ddot{\ell} = (1 - \vec{U}) \times \vec{\mathcal{M}} \times \mathcal{L} \tag{9}$$

$$\vec{\mathcal{M}} = \left(r_3 \times \left(1 - \vec{r}_5 \right) + \vec{r}_5 \right) \tag{10}$$

$$\vec{U}_1 = \begin{cases} 0, & \vec{r}_5 \le r_4 \\ 1, & \text{else} \end{cases} \tag{11}$$

$$U_2 = \begin{cases} 0, & r_3 \le r_4 \\ 1, & \text{else} \end{cases} \tag{12}$$

where $V_i(t)$ denotes the velocity of object i at time t. r_3 and r_4 are random values between 0 and 1. Similarly, r_5 and r_6 are also the random values. X_a and X_b are randomly selected solutions from the population. $R_i(t)$ is the distance between the optimal solution X_S and the target X_i at time t. $R_{i-norm}(t)$ means the normalization of the Euclidean distance between X_S and X_i.

The second stage is to update the object position and the KOA accomplishes this aim via two primary steps: one is the exploration phase, and another is the exploitation phase. It searches the objects distant from the Sun to discover novel solutions, whereas utilizing solutions near the Sun is accurate because it seeks novel locations in proximity to the best solutions. After the preceding steps, (13) is adopted to renew the position of each object located away from the Sun:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + F \times \vec{V}_i(t) + \left(F_{gi}(t) + |r| \right) \times \vec{U} \times \left(\vec{X}_s(t) - \vec{X}_i(t) \right) \tag{13}$$

where $\vec{X}_i(t+1)$ means the novel position of object i at t+1, $X_S(t)$ denotes the best position of the Sun so far, $\vec{V}_i(t+1)$ refers to the velocity required for object i to attain the novel position, and F indicates a flag to alter the direction of search.

The above reveals the optimal solution when the distance between the planets and the Sun is minimal, and its core is randomly exchanged with (13) to enhance exploration and exploitation. The randomized exchange ensures that KOA can dynamically adjust its search strategy according to the local characteristics of the optimization, prompting convergence speed and solution quality. In detail, KOA optimizes the exploitation operator when the planet is near the Sun and the exploration operator when the Sun is at a greater distance. For a large value of h, the exploration operator is adopted to extend the distance of the planet's orbit from the Sun. On the other side, for a small value of h, the exploration operator is utilized to explore the region around the current best solution. That means the exploitation operator is employed to exploit the region around the best solution when the distances between the Sun and the planets are smaller. This way can be further enhanced by randomizing the exchange with (14) for the exploration and exploitation operators:

$$\vec{x}_i(t+1) = \vec{x}_i(t) \times \vec{U}_1 + \left(1 - \vec{U}_1\right) \times \left(\frac{\vec{x}_i(t) + \vec{x}_S + \vec{x}_a(t)}{3.0} + h \times \left(\frac{\vec{x}_i(t) + \vec{x}_S + \vec{x}_a(t)}{3.0} - \vec{x}_b(t) \right) \right) \tag{14}$$

where h means an adaptive factor controlling the distance between the current planet and the Sun at time t.

The third stage is to iterate the optimal solution, and employing an elitist strategy can beneficially direct the search in an evolutionary or optimization algorithm by preserving the current best solution or individual. In each generation, exclusively the top-performing individuals are designated as elites and seamlessly transition to the next generation, ensuring the optimal positioning of the Sun and the planets:

$$\vec{X}_{i,new}(t+1) = \begin{cases} \vec{X}_i(t+1), \text{if} & f(\vec{X}_i(t+1)) \le f(\vec{X}_i(t)) \\ \vec{X}_i(t), & \text{else} \end{cases} \tag{15}$$

Through the aforementioned optimization, parameters such as the learning rate and weights in the network model can be fine-tuned to minimize prediction errors.

B. CNN-LSTM-Attention Model

The CNN-LSTM-Attention model not only accounts for the relationship among the multidimensional features of wind speed but also focuses on the temporal dependence of ambient data across various periods. Particularly, it includes the attention mechanism to assign weights to the output of each LSTM step. As a result, this hybrid model can accurately predict variations in wind speed data. Its details are described as follows:

Firstly, the CNN is employed to extract features from multidimensional time series data containing wind speed data, as it effectively captures feature relationships across both spatial and temporal dimensions. A common approach in wind speed prediction involves utilizing the sliding window method to transform the problem into a supervised learning task. This process entails dividing the historical wind speed data into two segments: one for model training and the other for model validation. Additionally, this study examines the influence of various periods on wind speed. Therefore, the Pearson correlation coefficient is applied to identify the four time periods most strongly correlated with wind speed changes, which are then used as new features. To enhance prediction accuracy, new features derived from wind speed data are incorporated, and CNN is employed to extract these features as inputs for the subsequent stage. To this end, a single layer of convolution followed by a pooling operation is applied. The initial convolutional operation incorporates a total of 64 filters with a filter size of 10, and the output of the CNN serves as the input to the LSTM.

Secondly, the LSTM is employed to acquire the temporal dependencies inherent in the wind speed data. It mainly mitigates the issue of gradient vanishing present in traditional recurrent neural network (RNN) and is appropriate for capturing long-range dependencies in time series data. Utilizing the LSTM allows effective modeling of temporal features in wind speed data, allowing the use of past observations for predicting future variations. The LSTM mainly incorporates a memory cell and a gating mechanism, which facilitates automatic learning and adjustment of time-step information for extracting features essential for wind speed prediction. Its computation can be expressed as:

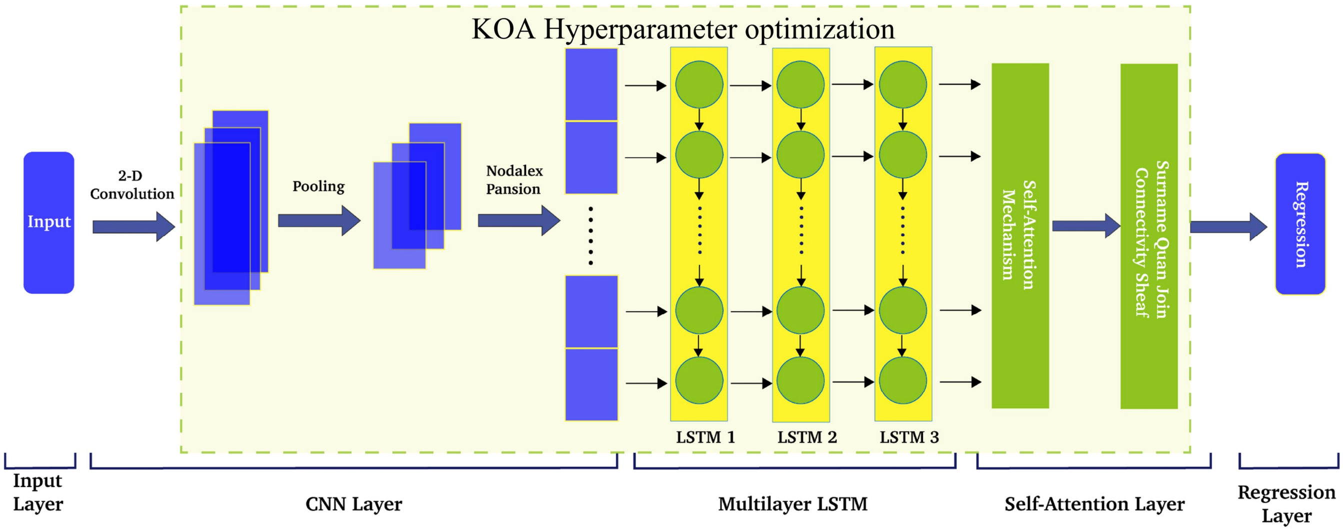

Fig. 1. The overall architecture of the proposed hybrid CNN-LSTM-Attention model with KOA.

$$f_t = \sigma\left(w_f \bullet [x_t, h_{t-1}] + b_f\right) \tag{16}$$

$$i_t = \sigma\left(w_i \bullet [x_t, h_{t-1}] + b_i\right) \tag{17}$$

$$o_t = \sigma\left(w_o \bullet [x_t, h_{t-1}] + b_o\right) \tag{18}$$

where f_t, i_t, and o_t represent the forgetting gate, input gate, and output gate, respectively.

Here, the forgetting gate f_t is for handling how much of the cell state c_t from the previous time is retained at the current time t. Similarly, the input gate i_t is for handling how much of the current transient state is input to the c_t, and the output gate o_t is for handling how much of the current cell state is output as a hidden layer. Then, the state of the current input unit $\widetilde{c}_t$ is derived from the input x_t of the network at the current time t and the h_{t-1} of the LSTM hidden layer output at the last time t-1 [25]:

$$\widetilde{c}_t = \tanh\left(w_c \bullet [x_t, h_{t-1}] + b_c\right) \tag{19}$$

where w_c is the weight matrix, b_c is the offset term, and tanh refers to the hyperbolic tangent function.

Now, the hidden layer output value of h_t and output x_t of the LSTM at the current time t is calculated by:

$$h_t = o_t \odot \tanh(c_t) \tag{20}$$

$$\widetilde{x}_{t+1} = \sigma\left(w \bullet h_t + b\right) \tag{21}$$

where $\odot$ means multiply by the element.

Thirdly, the attention mechanism is incorporated to assess the output of the LSTM at each time step. By prioritizing the significance of distinct time steps, the accuracy of predicting wind speed changes can be enhanced accordingly. The attention weights are automatically adjusted based on the significance of the features, and the attention-weighted LSTM outputs are used to forecast future data changes in wind speed via a fully connected network. Typically, the attention mechanism enhances the quality by allocating distinct weights to various parts of the model, directing more attention to components relevant to the task. Xie et al. [26] claimed that directly feeding data into the LSTM increases the complexity of the network model, and the data possess varying levels of importance for the prediction. In this context, with an increase in the prediction step size, a greater amount of historical data is needed. Consequently, a tailored attention layer is incorporated behind the LSTM to fine-tune the final output by introducing suitable weights to the outputs of the LSTM steps.

For better illustration, the overall architecture is presented in Fig. 1. The self-attention layer uses the output from each iteration of the LSTM as input and acquires the weights of each output vector following the Softmax operation. Subsequently, each LSTM output is multiplied by its weight and summed to generate the weighted vector. Finally, these weighted vectors are applied to the final outputs through the fully connected layer:

$$K = \tanh(X \cdot W + b) \tag{22}$$

$$A = \mathrm{softmax}(K) \tag{23}$$

$$out = sum(A \times X) \tag{24}$$

III. Experiment

As deep learning models typically require a specific input data format, the experimental data should be preprocessed, primarily involving the conversion of raw feature data and wind speed data into a 4-dimensional tensor suitable for model processing. In this experiment, wind speed data was collected from a region over 1800 time points across 18 different locations. The feature data contains 18 attributes pertinent to wind speed prediction, with wind speed data serving as the target variable. Subsequently, the data requires reshaping to adapt it to a format suitable for processing by the deep learning model. The MATLAB 'reshape' function is used. By specifying the appropriate dimensions, the feature data is transformed into a tensor with dimensions representing the number of samples, the number of features, the number of time steps, and the number of days. This transformation yields several benefits. Firstly, it allows the data to be fed directly into the deep learning model for training and prediction without additional processing or tuning. Secondly, representing the data as a 4-dimensional tensor better preserves the spatio-temporal information that can improve prediction accuracy. Finally, as the deep learning model is adept at handling specific input data formats, the transformed data can leverage the model's expressive power more effectively. To facilitate reproducible research and positively impact the academic community, all experimental data and source codes are available at https://github.com/Yorkson-huang/CNN-LSTM-Attention-Prediction.

IV. Results and Discussion

A. Prediction Results

Fig. 2 displays the variation of RMSE with the number of iterations and the loss function's change curve over iterations, using the CNN-LSTM-Attention with KOA. Here, the RMSE describes the standard deviation of the difference between the predicted values and the true values, and this curve shows the variation in prediction error in each iteration. Meanwhile, the loss function indicates the disparity between the predicted values and the true values, and the loss function's change curve provides insights into the convergence of the model during the training phase. It is common to evaluate models by examining convergence behaviors, and such metrics are vital as they provide a tangible measure of the model's accuracy and robustness over iterations, ensuring that the model generalizes well to unseen data.

Through the comparative analysis of the curves in Fig. 2, it is evident that during the model training process, compared to the loss function, the RMSE decreases more rapidly and significantly with the number of iterations. Thus, the significant decrease in RMSE observed is indicative of the hybrid model's superior predictive capabilities. This convergence pattern, with a more rapid reduction in RMSE compared to the loss function, demonstrates our approach's effectiveness in enhancing prediction accuracy and improving upon traditional methods.

To further investigate parameter optimization using KOA, it is necessary to establish an initial solution space by randomly generating initial planetary positions, with each planet's position corresponding to a candidate solution. Typically, increasing the number of initialized planetary positions broadens the search space, providing more initial candidate solutions. This practice enhances the likelihood of discovering the globally optimal solution, as a higher number of initial solutions allows for a more thorough exploration of the search space. However, increasing the number of initialized planetary positions may lead to higher computational costs, as each requires fitness calculations and iterative updates. Consequently, a balance between computational cost and search range must be maintained when determining the number of initialized planetary positions.

Fig. 3 illustrates the loss iteration curve for each case with varying initial numbers of planets (Nplan) and mean absolute percentage error (MAPE). The evaluation metrics include Nplan, the number of training iterations, and the value of the loss function. Here, Nplan indicates the number of initial planets used in the training process, the number of training iterations represents the frequency and extent of the training process, and the value of the loss function represents the loss at each iteration. Meanwhile, Fig. 4 depicts the adaptation curves under different initial planetary positions. The results validate that when Nplan is set to 7, the MAPE is only 0.0973, minimizing error and exhibiting optimal predictions. The choice of Nplan, similar to hyperparameter tuning in neural networks, plays a vital role in the model's training dynamics and overall performance. Thus, by setting Nplan to 7, the hybrid model achieves a MAPE of 0.0973, indicating high accuracy and demonstrating the model's robustness in minimizing prediction errors. This value is significant as it highlights the optimal configuration for the training process, akin to fine-tuning strategies in advanced deep learning models.

B. Ablation Experiment Results

To thoroughly investigate the inclusion or removal of specific attention mechanisms and optimization algorithms in the network model, an ablation experiment is conducted comparing CNN, CNN-LSTM, and CNN-LSTM-Attention in terms of MAPE, MAE, runtime, RMSE, and R^2. This evaluation comprehensively assesses the performance under different conditions and determines the impact of either the optimization algorithm (KOA) or the attention mechanism, and demonstrates that the CNN-LSTM-Attention model with KOA achieves superior performance on the evaluation metrics. The performance results of the different models are listed in Table I.

TABLE I
Ablation Experiment Results

Model	MAPE	MAE	Runtime (s)	RMSE	R^2
CNN	0.1772	19.2719	**12**	2.97	0.8613
CNN-LSTM	0.1648	18.3871	16	3.25	0.8626
CNN-LSTM-Attention	0.1330	23.8934	18	2.41	0.8953
CNN-LSTM-Attention with KOA	**0.0973**	**13.0910**	24	**1.78**	**0.9306**

As shown in Table I, concerning the MAPE, the proposed model achieves the lowest MAPE of 0.0973, while the CNN-LSTM-Attention model follows with a MAPE of 0.1330. Compared to the proposed model, the MAPE of both CNN-LSTM and CNN are higher. The MAE trends are consistent, with the proposed model achieving the best value at 13.0910. Such results validate that the combination of the attention mechanism and KOA optimization can result in a smaller average error and more accurate predictions. Subsequently, regarding the runtime, the proposed model takes 24 seconds, while the CNN-LSTM-Attention model uses 18 seconds, and the CNN model runs the fastest at only 12 seconds. Such differences are reasonable, as models with more components require relatively longer runtime. Additionally, in the evaluation based on RMSE, the proposed model achieves an RMSE of 1.78, whereas the CNN-LSTM-Attention model has an RMSE of 2.41, the CNN-LSTM model 3.25, and the CNN model 2.97. Also, the proposed model achieves the highest R^2 of 0.9306, indicating a better fitting effect than the others. Therefore, when considering the above metrics as overall concerns, the CNN-LSTM-Attention model with KOA performs well. From the ablation experiment, it is evident that the proposed model's impressive performance is due to its capability to provide highly accurate predictions, which is vital for applications requiring precision, such as wind speed prediction. These advancements also position the proposed model as a valuable contribution to deep learning, revealing its potential for impressive performance in related prediction applications.

C. *Performance Evaluation*

In the field of time-series data prediction, various algorithms have been used to optimize neural network hyperparameters, including evolutionary algorithms and metaheuristic methods such as simulated annealing (SA), genetic algorithm (GA), and particle swarm optimization (PSO). Each offers unique properties for navigating the complex search space of hyperparameters. In this paper, the KOA is applied to optimize the proposed model, and to validate its effectiveness, a comprehensive evaluation is conducted by comparing its optimization performance with the aforementioned algorithms, as listed in Table II.

TABLE II
PERFORMANCE EVALUATION OF DIFFERENT OPTIMIZATION ALGORITHMS

Optimization	MAPE	MAE	Runtime (s)	RMSE	R^2
CNN-LSTM-Attention with SA	0.1091	16.9459	25	1.97	0.8863
CNN-LSTM-Attention with GA	0.1107	17.0491	27	1.89	0.8728
CNN-LSTM-Attention with PSO	0.1075	14.9064	35	2.30	0.9135
CNN-LSTM-Attention with KOA	**0.0973**	**13.0910**	**24**	**1.78**	**0.9306**

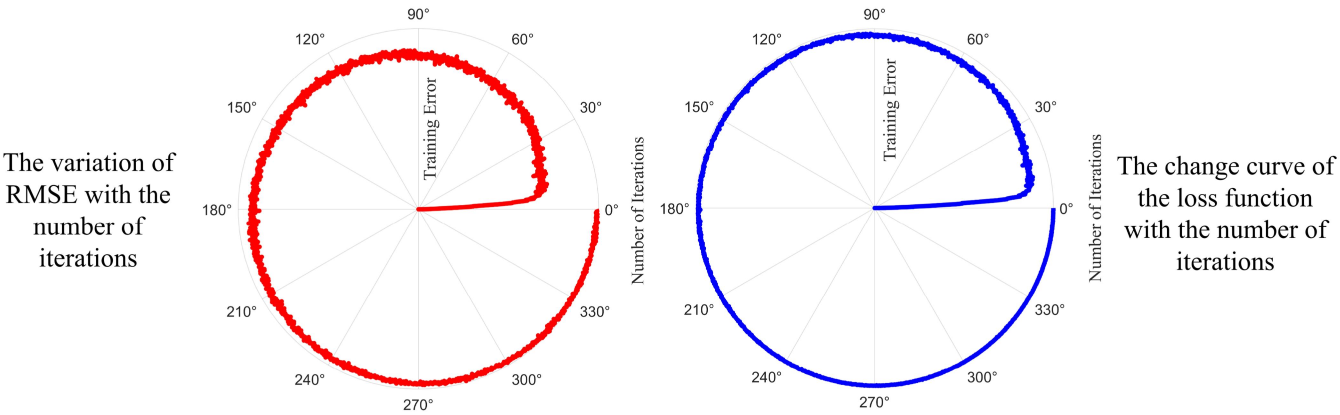

Fig. 2. Variation of RMSE and the loss function with the number of iterations using the proposed hybrid model.

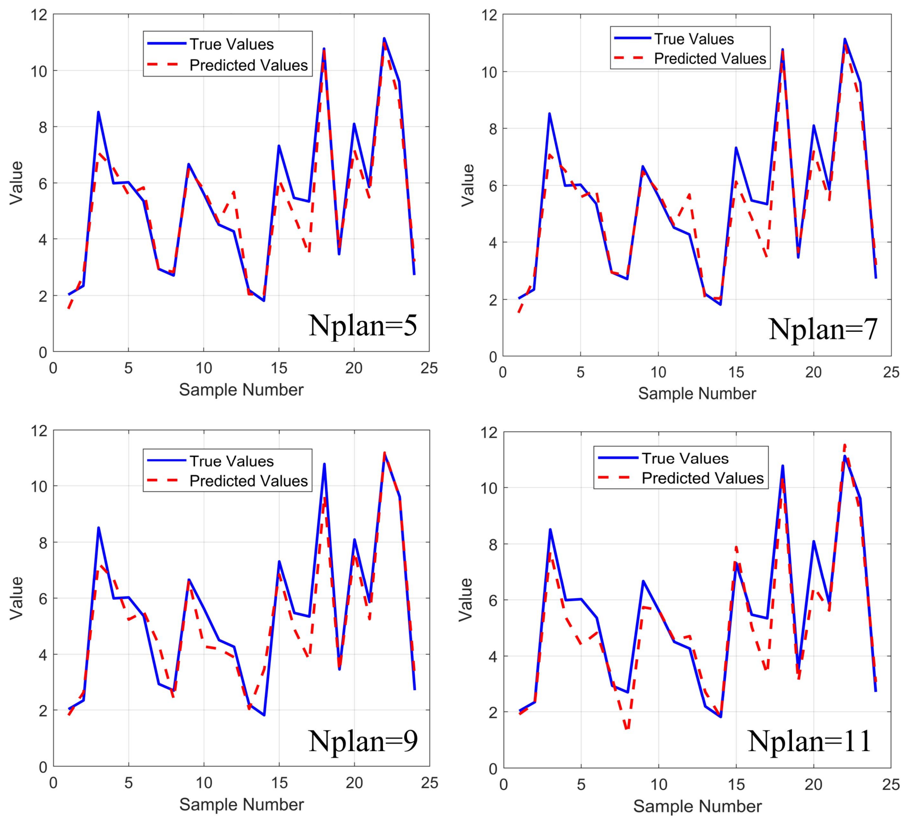

Fig. 3. The optimal predictions with different initial planets (Nplan).

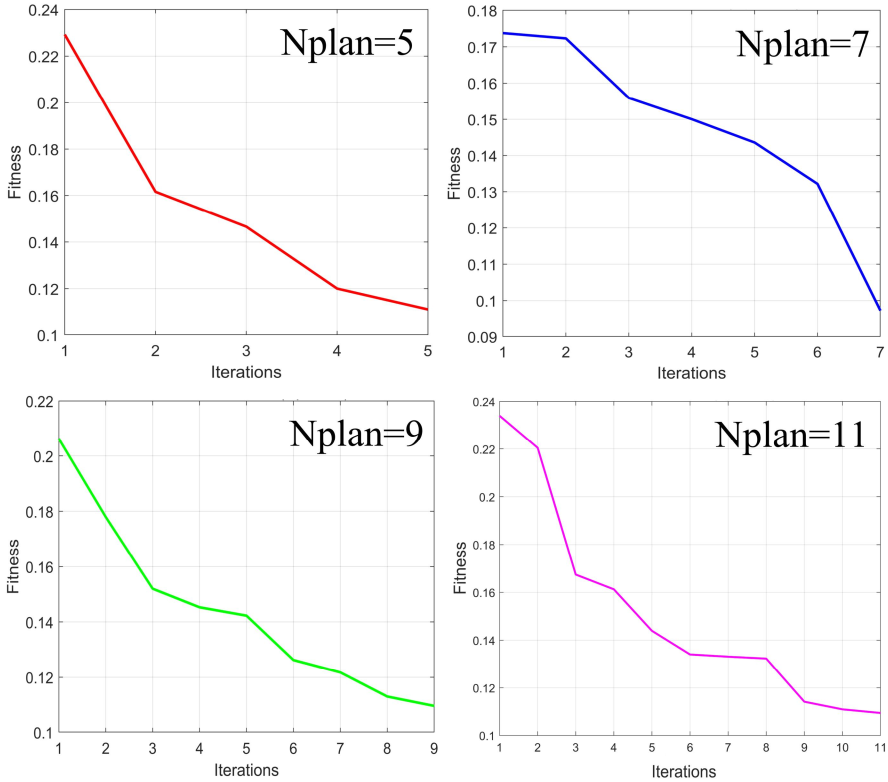

Fig. 4. The adaptation curves under different initial planets (Nplan).

From Table II, the KOA distinguishes itself across various metrics, indicating its effectiveness in optimizing model performance for wind speed prediction. On one hand, it achieves the best results in terms of MAPE and MAE, with the lowest values. This outcome indicates the KOA's exceptional capability in enhancing prediction accuracy, which is vital for ensuring reliable and precise forecasts in time-series applications. On the other hand, the KOA completes its run in just 24 seconds. This swift convergence highlights its computational effectiveness, making it well-suited for real-time environments where rapid model adaptation is imperative. Furthermore, the performance of the KOA extends to the evaluation of RMSE and R^2, displaying its superiority as a valid optimization approach for neural network models in wind speed prediction tasks. The remarkable outcomes indicate that the proposed model provides a significant improvement in wind speed prediction over existing optimization algorithms, reinforcing the value of the KOA in developing high-precision, efficient predictive models for time-series data. This property is also significant for deep learning applications, as the balance between accuracy and computational effectiveness is often crucial in dynamic environments.

D. Comparative Study

To extensively evaluate the effectiveness of the proposed method, we conduct a comparative study with previous works that adopt deep learning models, as presented in Table III. The results indicate that the proposed model yields the highest R^2, outperforming those models in previous works. This improvement can be attributed to the incorporation of the attention mechanism and the KOA optimization, where the attention mechanism dynamically prioritizes important features and the KOA optimization ensures efficient learning and convergence, leading to more accurate predictions. Therefore, such an advancement enriches neural network architectures by employing a robust optimization approach.

TABLE III
A COMPARATIVE STUDY WITH PREVIOUS WORKS

Work	Main methodology	Dataset	Highest R^2
Geng et al. [27]	Principal component analysis (PCA)-LSTM	From Fuyun meteorological station in China	0.9199
Guo et al. [28]	Wavelet neural network (WNN)	From Dabancheng wind farm in China	0.9030
Shu et al. [29]	Linear fast Fourier transform rank pooling multiple-layer perceptron/long short-term memory (LR-FFT-RP-MLP/LSTM)	From meteorological data collected at a wind farm in China	0.8430
Zheng et al. [30]	Legendre multi-wavelet neural network (LMWNN)	From the National Renewable Energy Laboratory (NREL)	0.8600
Parri and Teeparthi [31]	Successive variational mode decomposition with a transformer based model that incorporates a novel query selection mechanism (SVMD-TF-QS)	From two wind farms located in Leicester and Portland	0.9190
Jiao et al. [32]	Autoregressive moving average-support vector regression (ARMA-SVR) and error compensation	From a wind farm in China	0.9021
This work	**CNN-LSTM-Attention with KOA**	**From a wind farm in China**	**0.9306**

TABLE IV
BENCHMARKING USING OTHER STATE-OF-THE-ART PREDICTION MODELS

Method	MAPE	MAE	Runtime (s)	RMSE
Timesnet [33]	**0.0969**	14.8527	47	**0.9367**
SOFTS [34]	0.1106	15.1462	28	0.9155
UniTST [35]	0.1413	18.3697	39	0.8964
Itransformer [36]	0.1094	14.6681	30	0.9209
CNN-LSTM-Attention with KOA	0.0973	**13.0910**	**24**	0.9306

Finally, benchmarking using other state-of-the-art time series prediction models has been conducted on the same wind speed dataset evaluated in this work, and the results are summarized in Table IV. When comparing the performance evaluation of these studies, the proposed hybrid model exhibits an overall advantage. While it generates slight variation in terms of MAPE (+0.0004) and R^2 (-0.0061) compared to the best-performing model, Timesnet [33], these differences are minor, and our model remains highly efficient in practical applications because it completes in just 24 seconds, significantly faster than Timesnet's 47 seconds. This demonstrates that our model maintains operational efficiency with minimal compromise in prediction performance compared to the state-of-the-art. Such a comparison shows that the proposed method excels in prediction stability and reliability, optimizing time efficiency while maintaining satisfactory performance.

V. CONCLUSION

In this paper, we design a hybrid CNN-LSTM-Attention model that incorporates KOA hyperparameter optimization with the aim of improving wind speed prediction performance. This model effectively extracts relevant information from the original wind speed data and optimizes the neural network parameters accordingly. It also enables the development of an optimized prediction model that achieves remarkable results without compromising runtime. Notably, the proposed model demonstrates robust performance compared to other optimization algorithms, such as SA, GA, and PSO, outperforming across various metrics including MAPE, MAE, runtime, RMSE, and R^2. Additionally, benchmarking against other state-of-the-art time series prediction models shows that the proposed model demonstrates effective time efficiency while maintaining satisfactory results. In future work, we will explore its application in similar scenarios like wind power generation, which also involves high flexibility and intermittency. Meanwhile, several advanced neural network models in related deep learning fields [37–39] will be investigated and applied for wind speed prediction.

REFERENCES

[1] N. Abid, J. Wu, F. Ahmad, M. U. Draz, A. A. Chandio and H. Xu, “Incorporating environmental pollution and human development in the energy-growth nexus: A novel long run investigation for Pakistan,” *International Journal of Environmental Research and Public Health*, vol. 17, p. 5154, 2020.

[2] Y. Yan, X. Wang, F. Ren, Z. Shao and C. Tian, “Wind speed prediction using a hybrid model of EEMD and LSTM considering seasonal features,” *Energy Reports*, vol. 8, pp. 8965–8980, 2022.

[3] G. Chen, Z. Jiang, H. Tan, C. Zhang, J. Wei, W. Chen and H. Long, “Sliding-mode design with disturbance observer optimized by improved HBA algorithm for frequency regulation of multi-area power system,” *IAENG International Journal of Applied Mathematics*, vol. 52, no. 4, pp. 994–1013, 2022.

[4] S. Zhang, C. Zhu and X. Guo, “A novel combined model based on hybrid data decomposition, MSWOA and ENN for short-term wind speed forecasting,” *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 1000–1008, 2023.

[5] L. Wang, C. Li and C Li. “A short-term wind speed prediction method based on the DGA-BP neural network,” *IAENG International Journal of Computer Science*, vol. 51, no. 5, pp. 496–505, 2024.

[6] W. Ding and F. Y. Meng, “Point and interval forecasting for wind speed based on linear component extraction,” *Applied Soft Computing*, vol. 93, p. 106350, 2020.

[7] R. Corizzo and J. Rosen, “Stock market prediction with time series data and news headlines: A stacking ensemble approach,” *Journal of Intelligent Information Systems*, vol. 62, pp. 27–56, 2023.

[8] P. Lara-Benítez, M. Carranza-García and J. C. Riquelme, “An experimental review on deep learning architectures for time series forecasting,” *International Journal of Neural Systems*, vol. 31, no. 3, p. 2130001, 2021.

[9] B. R. Wu, L. Wang, and Y. R. Zeng, “Interpretable wind speed prediction with multivariate time series and temporal fusion transformers,” *Energy*, vol. 252, p. 123990, 2022.

[10] R. G. Yu, Y. Z. Sun, X. W. Li, J. Yu, J. Gao, Z. Q. Liu and M. Yu, “Time series cross-correlation network for wind power prediction,” *Applied Intelligence*, vol. 53, no. 10, pp. 11403–11419, 2023.

[11] Y. R. Chen, C. D. Cai, L. L. Cao, D. Zhang, L. M. Kuang, Y. Peng, H. Y. Pu, C. H. Wu, D. Zhou and Y. Cao, “WindFix: Harnessing the power of self-supervised learning for versatile imputation of offshore wind speed time series,” *Energy*, vol. 287, p. 128995, 2024.

[12] X. J. Chen and H. Y. Zhang, “Grey wolf optimization-based deep echo state network for time series prediction,” *Frontiers in Energy Research*, vol. 10, p. 858518, 2022.

[13] C. Kuranga, T. S. Muwani and N. Ranganai, “A multi-population particle swarm optimization-based time series predictive technique,” *Expert Systems with Applications,* vol. 233, p. 120935, 2023.

[14] X. Shen, Q. Dai, G. L. Zhu and J. Zhang, “Dynamic ensemble pruning algorithms fusing meta-learning with heuristic parameter optimization for time series prediction,” *Expert Systems with Applications*, vol. 225, p. 120148, 2023.

[15] L. Yang and Z. Zhang, “A deep attention convolutional recurrent network assisted by K-shape clustering and enhanced memory for short term wind speed predictions,” *IEEE Transactions on Sustainable Energy*, vol. 13, no. 2, pp. 856–867, 2022.

[16] C. Gu and H. Li, “Review on deep learning research and applications in wind and wave energy,” *Energies*, vol. 15, p. 1510, 2022.

[17] L. Zhang, S. He, J. Cheng, Z. Yuan and X. Yan, “Research on neural network wind speed prediction model based on improved sparrow algorithm optimization,” *Energy Reports*, vol. 8, pp. 739–747, 2022.

[18] F. Noman, G. Alkawsi, A. A. Alkahtani, A. Q. Al-Shetwi, S. Kiong Tiong, N. Alalwan, J. Ekanayake and A. I. Alzahrani, “Multistep short-term wind speed prediction using nonlinear auto-regressive neural network with exogenous variable selection,” *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 1221–1229, 2021.

[19] Y. Chen, Y. Wang, Z. Dong, J. Su, Z. Han and D. Zhou, “2-D regional short-term wind speed forecast based on CNN-LSTM deep learning model,” *Energy Conversion and Management*, vol. 244, no. 3, p. 114451, 2021.

[20] Y. Li, J. Ren, Y.Yan, A. Petrovski, “CBANet: An end-to-end cross band 2-D attention network for hyperspectral change detection in remote sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, p. 5513011, 2023.

[21] J. Ren, Z. Yu, G. Gao, G. Yu and J. Yu, “A CNN-LSTM-Light GBM based short-term wind power prediction method based on attention mechanism,” *Energy Reports*, vol. 8, pp. 437–443, 2022.

[22] M. Abdel-Basset, R. Mohamed, S. A. A. Azeem, M. Jameel and M. Abouhawwash, “Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion,” *Knowledge-Based Systems*, vol. 268, p.110454, 2023.

[23] S. Wang, H. Liu and G. Yu, “Short-term wind power combination forecasting method based on wind speed correction of numerical

weather prediction," *Frontiers in Energy Research*, vol. 12, p. 1391692, 2024.

[24] T. Sang-To, M. Hoang-Le and M. A. Wahab, "An efficient Planet Optimization Algorithm for solving engineering problems," *Scientific Reports*, vol. 12, p. 8362, 2022.

[25] A. Wan, Q. Chang, K. Al-Bukhaiti and J. He, "Short-term power load forecasting for combined heat and power using CNN-LSTM enhanced by attention mechanism," *Energy*, vol. 282, p. 128274, 2023.

[26] J. Xie, J. Zhang, J. Yu and L. Xu, "An adaptive scale sea surface temperature predicting method based on deep learning with attention mechanism," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 740–744, 2020.

[27] D. Geng, H. Zhang and H. Wu, "Short-term wind speed prediction based on principal component analysis and LSTM," *Applied Sciences*, vol. 10, no. 13, p. 4416, 2020.

[28] Z. Guo, L. Zhang, X. Hu and H. Chen, "Wind speed prediction modeling based on the wavelet neural network," *Intelligent Automation & Soft Computing*, vol. 26, no. 3, pp. 625–630, 2020.

[29] H. Shu, W. Song, Z. Song, H. Guo, C. Li and Y. Wang, "Multistep short-term wind speed prediction with rank pooling and fast Fourier transformation," *Wind Energy*, vol. 27, no. 7, pp. 667–694,2024.

[30] X. Zheng, D. Jia, Z. Lv, C. Luo, J. Zhao and Z. Ye, "Short-time wind speed prediction based on Legendre multi-wavelet neural network," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 3, pp. 946–962, 2023.

[31] S. Parri and K Teeparthi, "SVMD-TF-QS: An efficient and novel hybrid methodology for the wind speed prediction," *Expert Systems with Applications*, vol. 249, p. 123516, 2024.

[32] X. Jiao, D. Zhang, X. Wang, Y. Tian, W. Liu and L. Xin, "Wind speed prediction based on error compensation," *Sensors*, vol. 23, no. 10, p. 4905, 2023.

[33] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[34] L. Han, X. Y. Chen, H. J. Ye and D. C. Zhan, "SOFTS: Efficient multivariate time series forecasting with series-core fusion," *arXiv preprint arXiv:2404.14197*, 2024.

[35] J. Liu, C. Liu, G. Woo, Y. Wang, B. Hooi, C. Xiong and D Sahoo, "UniTST: Effectively modeling inter-series and intra-series dependencies for multivariate time series forecasting," *arXiv preprint arXiv:2406.04975*, 2024.

[36] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma and M. Long, "Itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[37] H. Long, Y. He, W. Xiang, Z. Guan, H. Tan and J. Yu, "Research on short-term wind speed prediction based on adaptive hybrid neural network with error correction," *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1290–1304, 2023.

[38] X. Wang, J. Wang, X. Niu and C. Wu, "Novel wind-speed prediction system based on dimensionality reduction and nonlinear weighting strategy for point-interval prediction," *Expert Systems with Applications*, vol. 241, p. 122477, 2024.

[39] C. Ma, Y. Shang, S. Jin, K. Yang and Z. Li, "Research on real-time energy management strategy of dual motor coupled PHEV based on model predictive control," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 1, pp. 76–85, 2023.