

Forecasting wind speed in Zabol city: a comparative study of CNN, LSTM, and CNN-LSTM models

Tohid Bagherpoor ^{*a}, Somayeh Kazemi Sormoli ^b

^a Department of Agronomy, Faculty of Agriculture, University of Zabol, Zabol, Iran

^b Department of Geo-information Engineering, School of Computer Science, China University of Geoscience (Wuhan), Wuhan, China

ARTICLE INFO

Article history:

Received: 4 October 2023

Accepted: 30 April 2024

Available online: 1 December 2024

Keywords:

Convolutional neural network

Hybrid models

Long short-term memory

Time series analysis

Wind speed prediction

ABSTRACT

One of the most essential concerns in renewable energy planning, weather forecasting, and environmental research is accurately predicting wind speed. This study evaluates and compares three machine learning methods: Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and a hybrid CNN-LSTM. The dataset includes daily-averaged wind speed (m/s), the day of the year, and additional meteorological variables recorded at Zabol station from 2010 to 2021. The dataset was preprocessed using Min-Max normalization and then split into four seasonal subsets: spring, summer, autumn, and winter. Each season was further divided into distinct subsets for training, validation, and testing in a 7:2:1 ratio. Python 3.8 was utilized for model development and data preparation using TensorFlow and Keras libraries. Performance evaluation metrics included Pearson correlation coefficient, root mean square error (RMSE), mean absolute error (MAE), and mean directional absolute percentage error (MDAPE). The results highlight that the CNN-LSTM hybrid model achieves superior accuracy compared to the standalone CNN and LSTM models. CNN effectively captures spatial patterns, while LSTM manages long-term temporal dependencies. The integration of these strengths in the CNN-LSTM model enhances forecasting accuracy across diverse conditions. Notably, the CNN-LSTM model achieved the highest accuracy in autumn, with an R^2 value of 0.980, showcasing its robustness in capturing wind speed variations across diverse seasonal conditions. This research highlights the strengths and limitations of each model. The main objective of this study is to precisely predict wind speed to optimize energy production, improve weather forecasting, and evaluate environmental impacts. Future research could explore additional meteorological variables, hybrid modeling, improved preprocessing techniques, and the use of larger datasets.

Highlights

- A novel integrated model was introduced for wind speed prediction.
- The proposed model demonstrated superior performance compared to three benchmark models across four seasons.
- The research highlighted the timeliness and significance by pointing out the absence of comprehensive studies comparing CNN, LSTM, and CNN-LSTM models in wind speed prediction.

1. Introduction

Wind speed prediction (WSP) is crucial for various applications, including renewable energy generation, weather prediction, and disaster management. Despite significant advancements, accurately predicting wind speed remains challenging due to its highly nonlinear and dynamic nature.

Accurate WSP plays a vital role in various sectors, including renewable energy management, aviation, and agriculture (Bali et al., 2019; Neshat et al., 2021a). Timely and reliable wind speed predictions enable efficient energy production planning, optimized flight routes, and improved crop management. Traditional numerical weather prediction models often face challenges in accurately

* Corresponding author.

E-mail address: t.bagherpour@uoz.ac.ir

<https://doi.org/10.22034/jelsa.2024.416818.1052>

capturing the complex and non-linear nature of wind patterns.

Recent advancements in WSP have incorporated mainstream deep learning techniques like CNN (Hinton and Salakhutdinov, 2006) and LSTM (Hochreiter and Schmidhuber, 1997). For instance, Jaseena and Kovoov explored decomposition methods in WSP, determining that an empirical wavelet transform combined with Bi-LSTM yielded accurate and reliable results (Jaseena and Kovoov, 2021). Yildiz et al. applied variational mode decomposition to extract time series features, reconstructing them into two-dimensional images for short-term wind power forecasting using a deep network model (Yildiz et al., 2021). Memarzadeh and Keynia utilized wavelet transforms for feature extraction in LSTM modeling (Memarzadeh and Keynia, 2020). Neshat et al. introduced a search-based CMA-ES model for hyperparameter tuning in LSTM training (Neshat et al., 2020). Bastos et al. integrated the U-Net semantic segmentation framework, widely utilized in medical image analysis, into WSP (Bastos et al., 2021). Additionally, hybrid models, which combine conventional time series models with machine learning, have gained traction in WSP, offering enhanced predictive power (Shobana Devi et al., 2021). However, these hybrid models cannot often capture spatiotemporal correlation data effectively.

As a result, machine learning models have gained significant attention for their ability to extract intricate patterns and make accurate forecasts. In this study, we focus on comparing three prominent machine learning models, namely CNN, Long LSTM networks, and the hybrid CNN-LSTM architecture, for WSP. By exploring the strengths and limitations of these models, we aim to contribute to the advancement of wind speed prediction (WSP) techniques and provide insights into the optimal choice of model for different forecasting scenarios. This research is motivated by the increasing need for reliable wind speed forecasts to support decision-making processes and facilitate the integration of renewable energy sources into the power grid. Moreover, understanding the comparative performance of these models can aid in designing more efficient operational strategies in sectors heavily reliant on wind speed information. To the best of our knowledge, no comprehensive study comparing CNN, LSTM, and CNN-LSTM models for WSP has been conducted in the literature, making this research timely and significant (Neshat et al., 2021b; Yang et al., 2022).

WSP is of significant importance in various sectors. In renewable energy management, accurate predictions are crucial for optimizing wind farms and maximizing energy production (Kusiak et al., 2013). These forecasts allow operators to schedule maintenance, plan turbine operations, and make informed decisions on energy storage and distribution (Ali, 2012). Grid operators can integrate wind energy efficiently, ensuring a stable electricity supply (Sun et al., 2015). In the agricultural sector, wind speed forecasts contribute to improved crop management practices. Farmers use this information for irrigation, pest control, and harvest timing (Vanderwende and Lundquist, 2015). Accurate predictions help mitigate the risk of wind damage,

optimize water usage, and enhance agricultural productivity.

CNNs have gained considerable attention in various domains, including computer vision and natural language processing. While initially designed for image analysis, CNNs have been successfully adapted for time series forecasting tasks. They excel at capturing local patterns and spatial dependencies through the use of convolutional filters. In time series forecasting, CNNs effectively extract temporal patterns and identify relevant features from the input data. By incorporating multiple convolutional layers, CNNs can learn hierarchical representations, enabling them to capture complex relationships within time series data (Dauphin et al., 2017).

LSTM networks, a type of recurrent neural network (RNN), are specifically designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. LSTMs are well-suited for time series forecasting due to their ability to retain and propagate information across different time steps. With memory cells that selectively store and update information, LSTMs excel at capturing temporal dynamics and complex patterns in time series data (Hochreiter and Schmidhuber, 1997).

The CNN-LSTM model combines the strengths of both CNNs and LSTMs, making it a powerful approach for time series forecasting. By integrating CNN layers to extract relevant features and LSTM layers to model temporal dependencies, the CNN-LSTM architecture captures both spatial and temporal information effectively. This hybrid model benefits from the CNN's ability to learn local patterns and the LSTM's capacity to capture long-term dependencies. By combining these two architectures, the CNN-LSTM model provides a comprehensive representation of the input data, enabling accurate and robust predictions in complex time series forecasting tasks (Qiao et al., 2022). Overall, the CNN-LSTM model offers a valuable solution for accurate and comprehensive time series forecasting.

Several studies have explored the application of machine learning techniques for the WSP, showcasing their potential for improving the accuracy of predictions. Sheng-Xiang Lv and Lin Wang conducted a comprehensive review of deep learning models for the WSP and highlighted the advantages of utilizing neural network (NN)-based approaches. They discussed the successful application of CNNs and LSTM networks in capturing temporal patterns and extracting relevant features from wind speed data. The review emphasized the importance of model architecture selection and data pre-processing techniques in achieving reliable wind speed forecasts (Lv and Wang, 2022).

Liu et al. proposed the use of LSTM NNs for WSP and compared their performance against traditional time series models. Their study demonstrated the superior accuracy of LSTM networks in capturing the complex dynamics of wind speed data. The ability of LSTM networks to capture long-term dependencies in the time series data was particularly advantageous in WSP tasks. The authors also highlighted the importance of data preprocessing, including

feature selection and normalization, for improving the performance of the LSTM model (Liu et al., 2020).

In addition to individual models, hybrid architectures combining CNN and LSTM networks have shown promising results in WSP. Qiao et al. (2022) proposed a hybrid CNN-LSTM model that leverages the strengths of both architectures. Their study demonstrated that the CNN-LSTM model effectively captured spatial and temporal information from wind speed data, leading to improved forecasting accuracy compared to standalone CNN or LSTM models. The hybrid model's ability to extract spatial features through the convolutional layers and model temporal dependencies with the LSTM layers contributed to its superior performance (Qiao et al., 2022).

These previous studies collectively highlight the potential of machine learning techniques, including CNNs, LSTMs, and hybrid architectures, in improving the WSP accuracy. However, despite the advancements in this field, there is still a need for comprehensive comparisons between different models to identify the optimal approach for specific forecasting scenarios. Moreover, considering the dynamic and non-linear nature of wind speed data, it is crucial to investigate the impact of various input features and model configurations on the performance of machine learning models. The present study aims to address these gaps by comparing the performance of CNN, LSTM, and CNN-LSTM models and providing insights into their suitability for the WSP tasks.

In summary, the existing literature on WSP using machine learning techniques exhibits gaps in terms of comparative analysis between different models, exploration of input feature combinations, and the focus on longer-term forecasting horizons. This study aims to address these limitations by conducting a comprehensive comparison of CNN, LSTM, and CNN-LSTM models, investigating the impact of various input features, and extending the prediction horizon to one year. By filling these gaps, this research will contribute to the advancement of WSP methodologies and provide valuable insights for decision-makers in various industries relying on accurate long-term WSPs.

The remainder of the paper is organized as follows: Section 2 provides the dataset from Zabol City, Iran, and the models are implemented. Section 3 presents the obtained results. Section 4 concludes the paper by summarizing the key findings and suggesting future research directions.

2. Material and Methods

2.1. Data Collection and Pre-processing

The data used in this study were collected from a meteorological station located in Zabol, with a latitude of 31.05, a longitude of 61.48, and an altitude of 489 meters. The meteorological station, equipped with various sensors and instruments, measures wind speed, temperature, humidity, atmospheric pressure, and other relevant meteorological variables. The wind speed data used in this study were recorded at a specific frequency on the tutiempo.net website (TuTiempo.Net, 2010-2021),

ensuring regular measurements and capturing the dynamic nature of wind patterns. The seven input variables (temperature, humidity, pressure, wind direction, solar radiation, etc.) were selected based on their demonstrated influence on wind speed, as supported by (e.g., Bali et al., 2019). Wind speed at 10 meters height was chosen as the target variable because it is a standard reference height in meteorological studies, ensuring consistency and comparability in results. The dataset comprises daily average wind speed measurements, spanning the period from 2010 to 2021. This time series dataset provides a valuable resource for the WSP research. With the availability of consistent and reliable data from the Zabol station, obtained from (TuTiempo.Net, 2010-2021), we aim to investigate the performance of various machine learning models in forecasting wind speed, contributing to the advancement of wind energy planning, weather prediction, and related fields.

The dataset comprises daily-averaged wind speed measurements spanning the period from 2010 to 2021 for 11 years. These wind speed measurements were recorded at the Zabol station, providing a consistent and reliable source of data obtained from (TuTiempo.Net, 2010-2021). Each entry includes the daily-averaged wind speed (m/s), day of the year, and additional relevant meteorological variables.

After pre-processing using Min-Max normalization, the dataset was split equally into four seasonal subsets: Spring, summer, autumn, and Winter. Each season was then divided into distinct subsets for training, validation, and testing in a 7:2:1 ratio (Qiao et al., 2022), corresponding to 720, 180, and 90 days, respectively. (See Figure 1). The training and testing sets were used to evaluate and train various machine learning models, while the validation set was used to fine-tune model hyperparameters. The statistical indicators for wind speed across the four seasons using the test set, including mean, maximum, minimum, standard deviation, skewness, and kurtosis, are presented in Table 1.

Figure 2 compares the wind speed distributions across the four seasons: spring, summer, autumn, and winter, with important statistical characteristics summarized in Table 1. The figure visually represents the frequency and density of wind speeds, allowing for a clear comparison of seasonal wind behavior.

Table 1 displays the key statistical characteristics of the wind speed data for each season of the test set. The dataset consists of 90 samples for each season, and the table provides the maximum and minimum wind speeds, mean wind speeds, standard deviation (STD), skewness, and kurtosis values. The summer test set has the highest mean wind speed (39.33 m/s) and a relatively low standard deviation (9.07 m/s), indicating less variability and stronger winds. In contrast, the autumn test set has the largest standard deviation (14.87 m/s) and the widest range of wind speeds, highlighting greater variability and more extreme wind conditions. The spring and winter test sets show moderate mean wind speeds, with the spring test set having the most symmetric distribution (skewness of 0.19) and the winter test set being more positively skewed

(skewness of 1.01), indicating more frequent lower wind speeds. The skewness and kurtosis values help describe the shape and behavior of the wind speed distributions in each season.

These important results from Table 1 help explain the patterns observed in Figure 2, where the summer test set

stands out with higher wind speeds and more concentrated distribution, while the autumn and winter test sets show more variability and extremes. The spring’s wind speeds exhibit moderate turbulence, indicating balanced but varied wind behavior.

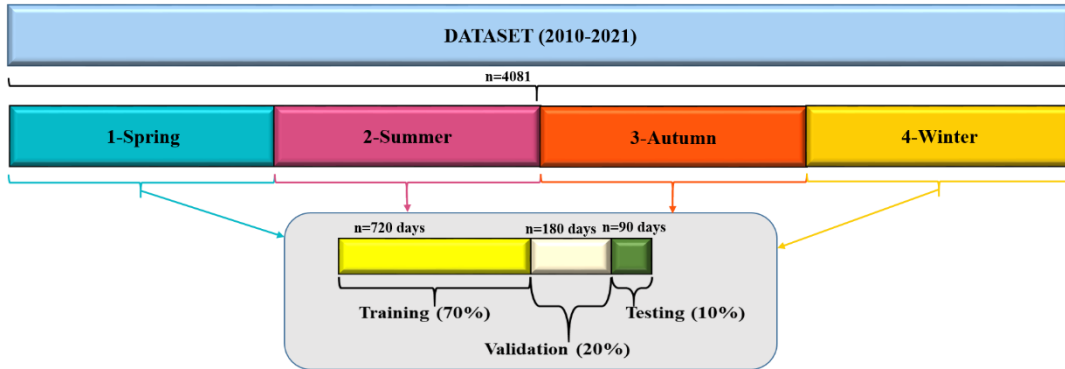


Figure 1. Dataset structure and division for the WSP (2010-2021)

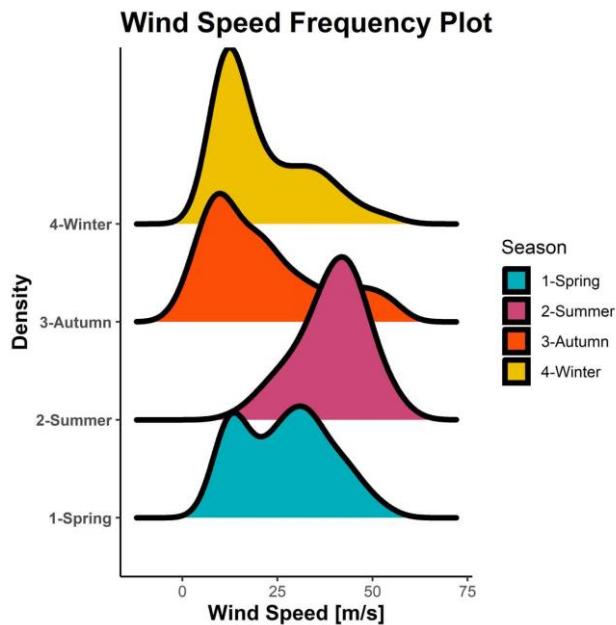


Figure 2. Wind speed frequency distribution across four seasons in 2021 (test set)

Table 1. Statistical characteristics of wind speed dataset for each season in 2021 (test set)

Season	Number of samples	Maximum (m/s)	Minimum (m/s)	Mean (m/s)	STD (m/s)	Skewness	Kurtosis
Spring	90	53.2	10	26.85	11.63	0.19	-0.96
Summer	90	59.1	14.3	39.33	9.07	-0.46	0.019
Autumn	90	55.2	0.9	20.88	14.87	0.83	-0.34
Winter	90	54.6	7	20.63	11.76	1.01	0.039

Before conducting any analysis or modeling, the collected wind speed data underwent a series of preprocessing steps. Missing values in the dataset were identified and treated using techniques such as linear interpolation or replacing them with the mean or median of neighboring values. Outliers, which could potentially affect the model's performance, were detected and either removed or adjusted. The wind speed data were then normalized using the Min-Max normalization to bring them to a common scale between 0 and 1, ensuring that all values fell

within a standardized range by using the following equation (Radhika and Shashi, 2009):

$$y_{norm} = \frac{y - \min(y)}{\max(y) - \min(y)} \tag{1}$$

where y_{norm} is the scaled version of the unscaled variable y .

In addition to wind speed, other meteorological variables such as temperature, humidity, and atmospheric pressure were collected from the same meteorological station. These variables underwent similar preprocessing steps, including handling missing values, treating outliers, and normalization. The dataset was divided into training, validation, and testing sets to facilitate the modeling process. The partitioning was done chronologically, preserving the temporal order of the data and avoiding any leakage of information from future observations into the training or validation sets. By ensuring data quality through preprocessing and an appropriate train-test split, we can generate reliable datasets for training and evaluating the CNN, LSTM, and CNN-LSTM models. These steps lay the foundation for accurate WSP and contribute to advancements in wind energy planning, weather prediction, and related fields.

2.2. Proposed CNN, LSTM, and CNN-LSTM Architectures

A CNN is a sophisticated feed-forward NN that employs convolutional operations to autonomously extract features. Its architecture aligns with the principles of local perception, weight sharing, and the integration of multiple convolution kernels found in traditional NN models (Lecun et al., 1998). The CNN model is a deep learning architecture commonly used for image processing tasks, but it can also be applied to time series forecasting (Qiao et al., 2022). The CNN model consists of convolutional layers, pooling layers, and fully connected layers. In the context of WSP, the input data is treated as a one-dimensional (1D) sequence, where each data point represents a time step. The convolutional layers perform feature extraction by applying filters to the input sequence, capturing local patterns and spatial dependencies. The pooling layers reduce the dimensionality of the extracted features, aggregating the information across neighboring time steps. The fully connected layers at the end of the network combine the extracted features and generate the final predictions. The CNN model's architecture and hyperparameters, such as the number of filters, filter size, pooling size, and activation functions, can be tuned to optimize the model's performance for the WSP tasks (Karthik et al., 2020).

The CNN's layer inputs are sequentially derived from the preceding layer's outputs, with the initial layer being the time series data of wind speed. The convolution computation process unfolds as follows (Chen et al., 2021):

$$X_j^{(l)} = \text{ReLU}\left(\sum_{i \in \mathfrak{N}_j} W_{i,j}^{(l)} * X_j^{(l-1)} + b_j^{(l)}\right) \quad (2)$$

where: $X_j^{(l)}$ is feature map j of layer l ; $X_j^{(l-1)}$ is the output of the previous layer; $*$ is the convolution operation; $W_{i,j}^{(l)}$ is the convolution kernel; and $b_j^{(l)}$ is the bias. All parameters are to be learned. The rectified linear unit (ReLU) stands as a widely employed non-linear activation

function in NNs, defined by $\text{ReLU}(X) = \max(0, X)$. It serves the purpose of establishing intricate functional connections between intermediate outcomes and the characteristics of wind speed.

The fundamental architecture of a CNN comprises convolutional layers, pooling layers, and fully connected layers. Convolutional layers generate feature maps by applying moving kernels and activation functions to gridded input data through dot product calculations. Pooling layers, on the other hand, reduce the input data's size by selecting representative values (such as maximum or mean) from defined windows. To prevent excessive reduction in image size during the convolutional and pooling processes, padding techniques involving the addition of extra dummy data are commonly employed. Lastly, fully connected layers transform the spatial features from the previous layer, which are in a gridded format, into 1-D vectors. These transformed data are then input into densely connected NNs to produce the final output.

Many hyperparameters define a NN, but increasing the number of optimization variables extends the duration of the Bayesian optimization process (Frazier and Wang, 2016). Therefore, in this study, several hyperparameters were assigned fixed and conservative values. These selected hyperparameters primarily affect the network's training time rather than its accuracy, so sub-optimal values are unlikely to impact the final results' accuracy. To cut down on computation time, the networks trained during hyperparameter optimization were given random segments of 10% of the training and validation data, evenly distributed across the mean wind speed time histories. The hyperparameter optimization process was conducted twice for each network: initially, a broad range of values was used for each hyperparameter; subsequently, the range was narrowed significantly around the optimal value identified in the first round. The specific hyperparameter values for each NN are detailed in the following sections. Bayesian optimization was provided by the Skopt package in Python 3.8.12 with 120 rounds of optimization for each network.

The CNN employs an iterative backpropagation training process to extract the most effective feature maps from input data through hyperparameter optimization. In this research, the optimal input feature size, structure, and hyperparameters of the proposed CNN model were determined with a focus on both performance and efficiency, as illustrated in Figure 3. The input data, measuring 31×31 in size, consisted of features and was fed into the CNN. The CNN model was trained using the MAE loss function and the Adaptive moment estimation (Adam) optimizer. Following numerous tests with various combinations of structure and hyperparameters, the model incorporated two convolutional layers with filter sizes of 3×3 and capacities of 64 and 128 filters, respectively. A subsequent max-pooling layer with a 2×2 window size was connected, followed by a fully connected layer housing 64 nodes. The final output layer corresponds to the target output value. Throughout all convolutional and fully connected layers, the ReLU activation function was consistently employed in this investigation.

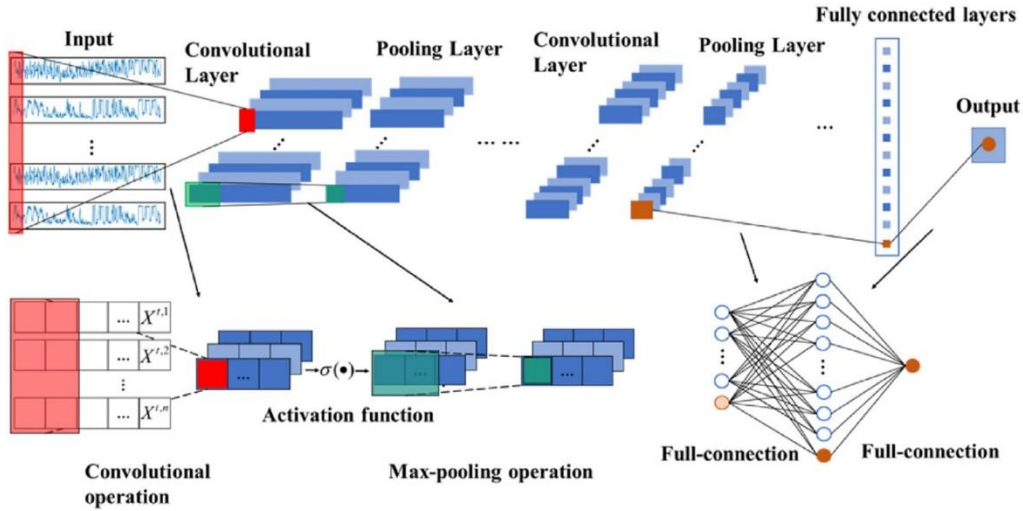


Figure 3. Architectures of CNN used in this study

The LSTM model is a type of RNN that is specifically designed to capture long-term dependencies and patterns in sequential data. Unlike traditional feed-forward NNs, LSTM models incorporate memory cells and gates, allowing them to retain and update information over multiple time steps. In the WSP, the LSTM model processes the input data sequentially, with each time step considered a sequence element. The model's architecture typically consists of LSTM layers, followed by fully connected layers. The LSTM layers use gates to control the flow of information and selectively remember or forget certain patterns over time. The fully connected layers combine the learned representations from the LSTM layers and produce the final WSPs. The LSTM model's architecture and parameters, including the number of LSTM cells, activation functions, and dropout rates, can be adjusted to optimize the model's performance for the WSP tasks.

The LSTM incorporates memory units within neurons to offer long-term memory capabilities and introduces a gating mechanism to regulate the flow of information to subsequent stages. This gating mechanism comprises an input gate $i(t)$, a forget gate $f(t)$, and an output gate $o(t)$. These three gates, along with the candidate hidden states $\tilde{c}(t)$, can be represented similarly as follows (Chen et al., 2021):

$$\begin{bmatrix} i(t) \\ f(t) \\ o(t) \\ \tilde{c}(t) \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \varphi \end{bmatrix} \left(W^{(*)} \begin{bmatrix} x(t) \\ s(t-1) \end{bmatrix} + b_{(*)} \right) \quad (3)$$

where $(*) \in \{i, f, o, \tilde{c}\}$ σ represents the sigmoid function yielding values within the range $[0, 1]$; φ denotes the hyperbolic tangent non-linear function, $\tanh()$; $x(t)$ stands for the input at time t ; $s(t-1)$ represents the external state from the previous time step; and b is an offset term. The

hidden state from the previous time, $c(t-1)$, and the candidate hidden state at the current time $\tilde{c}(t)$ are combined to form the complete state for the subsequent time step, which is expressed as:

$$c(t) = f(t) \cdot c(t-1) + i(t) \cdot \tilde{c}(t) \quad (4)$$

The calculation for obtaining the output of the current unit, which simultaneously serves as the hidden layer for the subsequent state $s(t)$, is as follows (Hou et al., 2022):

$$s(t) = o(t) \cdot \tanh[c(t)] \quad (5)$$

In detail, the LSTM model incorporates essential components, including a memory cell (c), an input gate (i), a forget gate (f), and an output gate (o). These components work in tandem to significantly enhance the NN's capacity to handle lengthy, sequential data. The interplay between these elements is illustrated in Figure 4. Specifically, the input gate regulates the influence of new input on the memory cell (c), the forget gate governs the influence of the memory unit in retaining the prior value, and the output gate regulates the impact of the output memory cell. By utilizing an LSTM cell, the inputs $x(t)$ and $s(t)$ can be used to compute $s(t)$, which is then transmitted to the subsequent layer and forwarded to the fully connected layer to derive the output value.

During the training of the LSTM model, the number of epochs was set to 150. Training sets were processed using mini-batch gradient descent. After forming the mini-batches, gradients were calculated for each batch, followed by parameter updates. The mean square error (MSE) was selected as the loss function. The optimization algorithm used was Adam, with a learning rate of 0.001, known for its effectiveness in practical applications. This algorithm adjusted the model's weights and bias parameters. Each LSTM layer was configured with 100 neurons, and the model included two LSTM layers with 50 units each, utilizing ReLU activation and dropout regularization.

The CNN-LSTM model combines the strengths of both CNN and LSTM architectures, aiming to capture both spatial and temporal patterns in the input data. This model is well-suited for time-series forecasting tasks where both local patterns and long-term dependencies are important.

The CNN-LSTM architecture typically starts with a few convolutional layers to extract spatial features from the input sequence. The output from the convolutional layers is then reshaped into a three-dimensional representation to be fed into the LSTM layers.

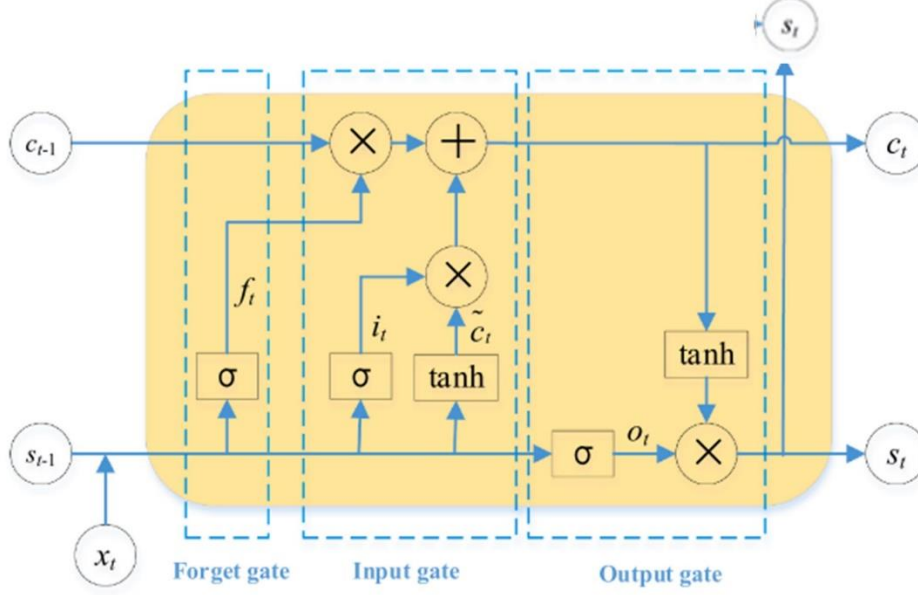


Figure 4. The basic LSTM cell

2.2. Evaluation metrics

Model performance was evaluated using appropriate evaluation metrics, such as root mean squared error (RMSE), mean directional absolute percentage error (MDAPE) or mean absolute error (MAE), Pearson correlation coefficient (PCC), and the coefficient of determination (R^2). These metrics provided quantitative measures of the models' accuracy in predicting wind speed. The models were also assessed based on their ability to capture temporal patterns and produce reliable long-term forecasts. The performance of each model was compared using statistical tests or other benchmark models to determine if the proposed CNN-LSTM model outperformed the individual CNN and LSTM models in the WSP. These metrics include:

- RMSE: It is the square root of the average squared difference between the predicted and actual wind speed values and provides a more interpretable measure of prediction error (Liu et al., 2019).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_{predicted} - X_{observed})^2} \quad (6)$$

- MAE: It calculates the average absolute difference between the predicted and actual wind speed values, providing a measure of the average magnitude of the prediction error (Liu et al., 2019).

$$MAE = \frac{1}{N} \sum_{i=1}^N |X_{predicted} - X_{observed}| \quad (7)$$

- MDAPE: It quantifies the average absolute percentage difference between the predicted and actual wind speed values. MDAPE provides a measure of the average percentage error in capturing the magnitude of wind speed changes (Lv and Wang, 2022).

$$MDAPE = \text{median} \left(\sum_{i=1}^N \frac{|X_{observe} - X_{predicted}|}{X_{observed}} \right) \times 100\% \quad (8)$$

- PCC: The PCC is a widely used statistical measure that quantifies the strength and direction of the linear relationship between two continuous variables. It ranges from -1 to 1, with negative values indicating a negative linear relationship, positive values indicating a positive linear relationship, and 0 suggesting no linear relationship. PCC is calculated by comparing the covariance of the two variables to the product of their standard deviations. It is a valuable tool for researchers and analysts in various fields to assess and quantify associations between variables, though it assumes linearity and may not capture more complex relationships (Yang et al., 2022).

$$PCC = \frac{\sum_{i=1}^N ((X_{observe} - \bar{X})(X_{predicted} - \bar{X}_{predicted}))}{\sqrt{(\sum_{i=1}^N (X_{observe} - \bar{X})^2) (\sum_{i=1}^N (X_{predicted} - \bar{X}_{predicted})^2)}} \quad (9)$$

- R^2 (R-squared): R-squared is a statistical measure that reflects how well the predicted wind speed values align with the actual measured values (Yang et al., 2022).

$$R^2 = 1 - \frac{\sum_{i=1}^N (X_{observe} - X_{predicted})^2}{\sum_{i=1}^N (X_{observe} - \bar{X})^2} \quad (10)$$

where $X_{predicted}$ is the prediction results, $X_{observed}$ is the real data, N is the number of real data, and $\bar{X}_{predicted}$, \bar{X} is the average of the prediction result and real data, respectively. These metrics provided measures of accuracy, prediction error magnitude, the proportion of variance explained by the models, and the average absolute percentage difference between predicted and actual values. It is important to consider that lower values of MAE and RMSE are indicative of better performance. The R^2 ranges from 0 to 1, and a value closer to 1 signifies a stronger alignment between the model's predictions and the actual measured values.

2.3. Experimental setup

In this study, the proposed network models were trained and evaluated using specific hardware and software

environments. The hardware consisted of an Intel Core i7-8750H processor and an Nvidia GeForce GTX 1070 GPU, providing powerful computing ability and memory bandwidth. Deep learning frameworks like TensorFlow or PyTorch were utilized for model implementation and training. The Nvidia GeForce GTX 1070 GPU accelerated the training process due to its parallel processing capabilities. Python, along with libraries like Pandas and NumPy, was used for data manipulation and preprocessing tasks. This hardware and software setup ensured efficient analysis and reduced computational time.

3. Results and discussions

To assess the performance of our proposed CNN-LSTM model, we conducted comparative experiments using three individual models: CNN, LSTM, and CNN-LSTM (Chen et al., 2021). These experiments aimed to assess the generalization ability of the model across different seasons (spring, summer, autumn, and winter).

As shown in Table 2, the performance of all four models was evaluated using key metrics such as MAE, RMSE, MDAPE, PCC, and R^2 . The results indicate that the proposed CNN-LSTM model consistently outperforms the other models in single-task prediction across all seasons. The CNN-LSTM model by Chen et al. (2021) displayed the second-best performance, followed by LSTM, with the CNN model showing the least favorable results.

Table 2. Assessing the precision of the four models in forecasting one step ahead.

Season	Model	Evaluation metrics				
		MAE	RMSE	PCC	MDAPE (%)	R^2
Spring	CNN	0.131	0.159	0.59	29.05	0.480
	LSTM	0.123	0.156	0.62	29.88	0.560
	CNN-LSTM (Chen et al.,2021)	0.408	0.525	-	-	0.934
	CNN-LSTM (proposed model)	0.047	0.056	0.95	10.36	0.966
Summer	CNN	0.107	0.135	0.47	16.78	0.472
	LSTM	0.096	0.116	0.53	14.91	0.554
	CNN-LSTM (Chen et al.,2021)	0.469	0.579	-	-	0.923
	CNN-LSTM (proposed model)	0.048	0.062	0.93	7.21	0.950
Autumn	CNN	0.150	0.210	0.51	40.06	0.492
	LSTM	0.144	0.201	0.58	37.06	0.579
	CNN-LSTM (Chen et al.,2021)	0.186	0.311	-	-	0.979
	CNN-LSTM (proposed model)	0.052	0.064	0.97	16.94	0.980
Winter	CNN	0.167	0.237	0.23	42.45	0.481
	LSTM	0.144	0.200	0.37	36.64	0.571
	CNN-LSTM (Chen et al.,2021)	0.393	0.495	-	-	0.964
	CNN-LSTM (proposed model)	0.048	0.062	0.95	14.71	0.975

These findings strongly validate the effectiveness and generalization capabilities of the model developed in this study. In the spring season, the proposed CNN-LSTM model demonstrated remarkable accuracy with the lowest MAE (0.047) and RMSE (0.056) values and a PCC of 0.95, indicating a strong linear relationship between predicted and actual wind speeds. The model also achieved the

lowest MDAPE (10.36%) and a high R^2 (0.966), reflecting its excellent predictive capabilities. These trends were observed consistently across all seasons, highlighting the robustness of the model in predicting wind-related variables under diverse conditions (Schepers and Snel, 2007; Yang et al., 2022).

Graphical representations of the forecast results for each models across seasons are presented in Figures 5-8. These visuals highlight the challenges posed by abrupt changes in wind speed, which significantly impact forecasting accuracy. Figure 5 demonstrates that the proposed CNN-LSTM model outperforms the individual CNN and LSTM models, especially in spring, by closely following the wind speed changes. The CNN model, typically used for image data processing, struggles with time series data, showing large deviations between predicted and actual wind speeds. The limitation is likely due to the CNN's relatively small receptive field, which hinders effective extraction of long-range temporal features. Similarly, the LSTM model shows slight improvements, but it also fails to accurately predict sharp wind speed changes (Yang et al., 2022).

Incorporating LSTM improves predictions slightly, especially in periods of stable wind speeds. However, combining LSTM with CNN (i.e., single-task CNN-LSTM) significantly enhances performance, as illustrated in Figures 6-8. Similar to the findings of Qiao et al. (2022), who demonstrated the effectiveness of their MS-CDL

model in handling sudden changes in wind speed over complex terrains, our proposed CNN-LSTM model also excels at managing sharp fluctuations without requiring multi-site data input. The model's ability to maintain high accuracy under such conditions underscores its robustness and efficiency in the WSP for diverse environments (Scheepers and Snel, 2007; Yang et al., 2022).

Table 2 demonstrates that during the summer season, the proposed CNN-LSTM model exhibits outstanding predictive performance, with an MAE of 0.048 and an RMSE of 0.062. The model also achieves a high R^2 value of 0.950, indicating a strong correlation between the predicted and actual wind speeds. Figure 6 further highlights the model's effectiveness in closely tracking wind speed fluctuations, particularly during the peaks and troughs. This superior performance underscores the model's ability to handle sudden changes in wind speed, making it a reliable option for summer WSP. This model captures the rapid variations in wind speed, especially during abrupt changes, far more accurately than other models.

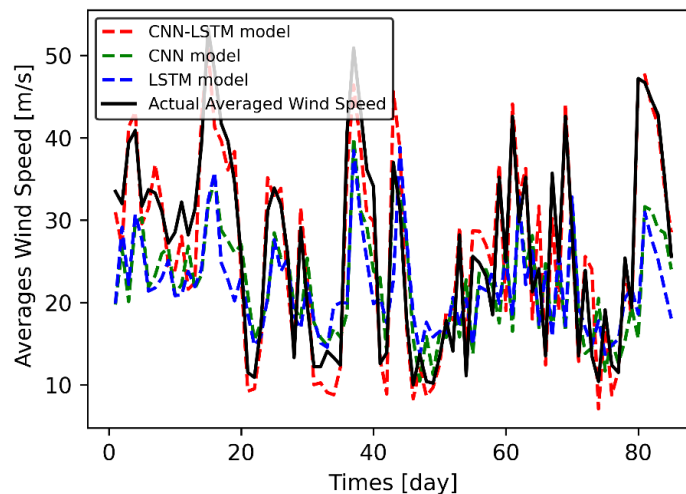


Figure 5. Comparison of CNN, LSTM, CNN-LSTM (proposed) model, and actual wind speed values on the spring test set

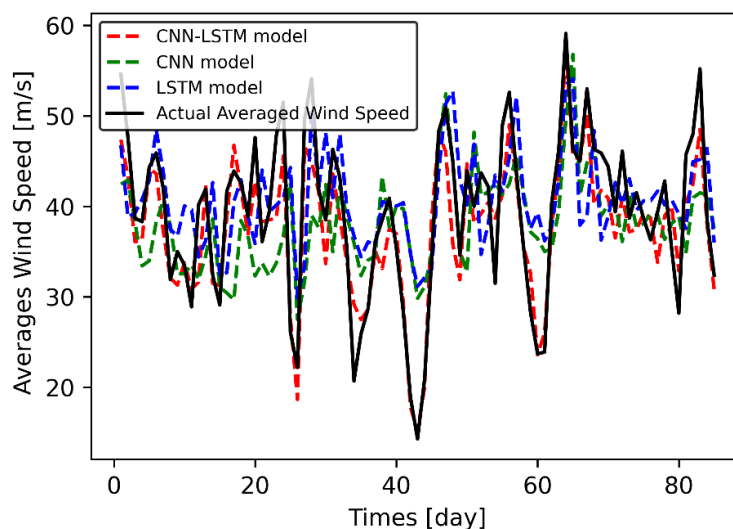


Figure 6. Comparison of CNN, LSTM, CNN-LSTM (proposed) model, and actual wind speed values on the summer test set

In the autumn season, as shown in Figure 7, the proposed CNN-LSTM model delivered its strongest performance, achieving an MAE of 0.052 and an RMSE of 0.064. The R^2 value of 0.980 demonstrates the model’s exceptional accuracy in predicting wind speed, even during the most challenging conditions when frequent and abrupt changes occur. This result is consistent with Liu et al. (2020), who found that hybrid models incorporating signal decomposition techniques and feature selection methods achieve superior forecasting performance for multi-step predictions, further validating the use of hybrid architectures like CNN-LSTM (Liu et al., 2020; Yang et al., 2022).

In the winter season, the proposed CNN-LSTM model continues to outperform other models with an MAE of 0.048 and an RMSE of 0.062. The high R^2 value of 0.975 indicates its robustness in capturing sudden wind speed changes and delivering highly accurate predictions. Figure 8 shows that, unlike the CNN and LSTM models, which exhibit notable discrepancies, the proposed CNN-LSTM model closely aligns with the actual wind speed, maintaining precision even during sudden fluctuations. This demonstrates that the proposed model is the most dependable option for winter forecasting, where wind speed changes tend to be more unpredictable and difficult to predict (Qiao et al., 2022).

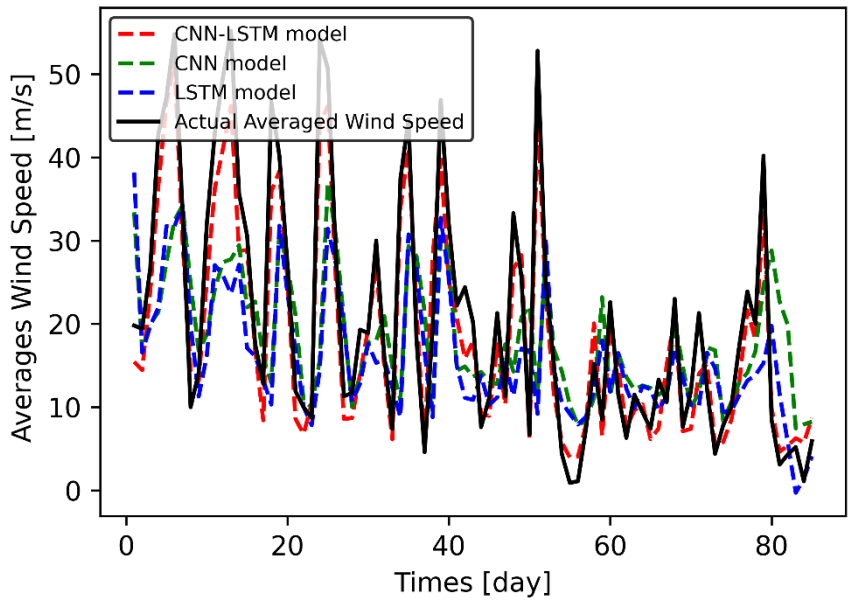


Figure 7. Comparison of CNN, LSTM, CNN-LSTM (proposed) model, and actual wind speed values on the autumn test set

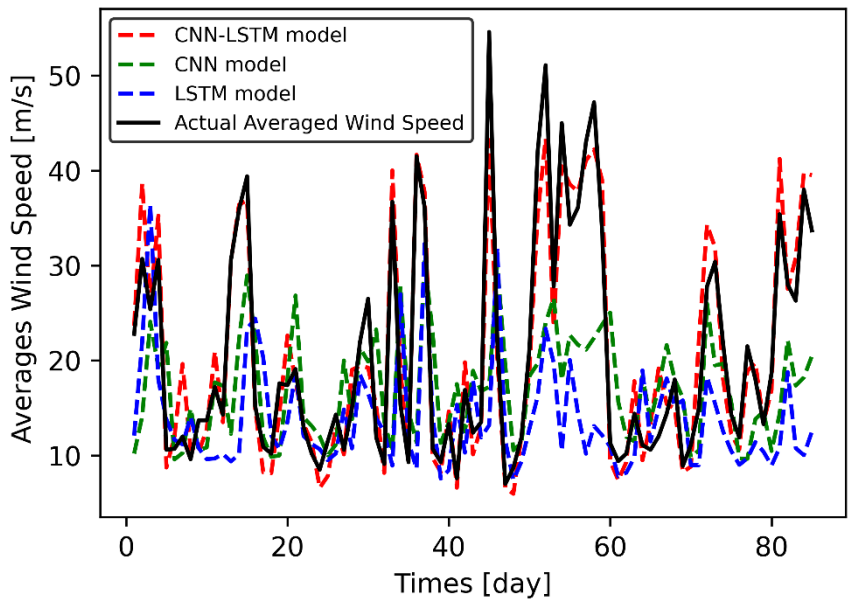


Figure 8. Comparison of CNN, LSTM, CNN-LSTM (proposed) model, and actual wind speed values on the winter test set

Figure 9 illustrates the R^2 values for each model in this study across the four distinct seasons: spring, summer, autumn, and winter. These figures provide a comprehensive overview of the model's performance in terms of achieving an exact fit between forecasted and real wind speed data. Notably, our proposed approach,

employing CNN with LSTM, consistently stands out as the top-performing model, attaining an exceptional R^2 value of 0.98 for the autumn. This remarkable R^2 value underscores the exceptional accuracy with which our proposed method aligns the forecasted and actual wind speed data, thereby demonstrating its superior predictive capabilities.

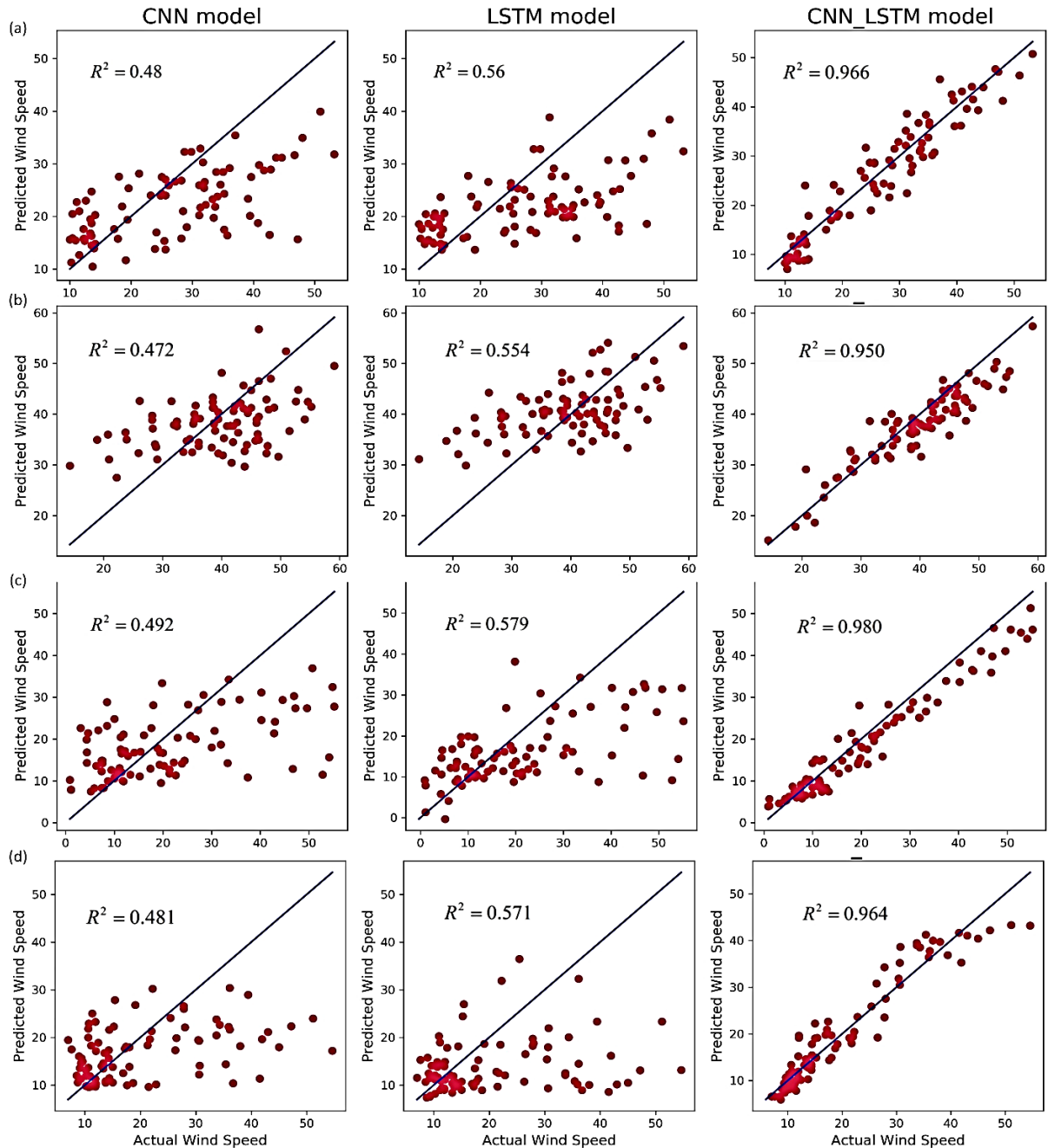


Figure 9. The comparison of a different forecasting model in (a) spring, (b) summer, (c) autumn and (d) winter seasons.

4. Conclusions

In this study, we conducted the interpretation and analysis of the results, which shed light on the performance of the CNN, LSTM, and CNN-LSTM models in the context of WSP. The superior performance of the CNN-LSTM

model can be attributed to its ability to capture both the spatial and temporal characteristics of the wind speed data.

4.1 Contribution of the CNN-LSTM Model in WSP

By incorporating convolutional layers, the CNN component of the model effectively captures spatial patterns and local features, such as wind direction and geographic influences. On the other hand, the LSTM

component captures the temporal dependencies and long-term trends in the data. The integration of these two components in the CNN-LSTM model results in a more comprehensive representation of the complex interactions influencing wind speed variations.

Furthermore, the CNN-LSTM model's ability to leverage the strengths of both CNN and LSTM architectures addresses some of the limitations of the individual models. While the CNN model focuses on spatial features, it may overlook temporal dependencies, and vice versa for the LSTM model. The CNN-LSTM model overcomes this limitation by combining both aspects, enabling a holistic understanding of wind speed patterns. This integration proves to be particularly valuable in the WSP, where both spatial and temporal factors play significant roles.

The results also highlight the importance of considering appropriate evaluation metrics in the WSP. The MAE, RMSE, MDAPE, PCC, and R^2 provide valuable insights into the accuracy, precision, and linear relationship between predicted and actual wind speed values. These metrics help in assessing the model's performance in capturing the overall wind speed patterns and in quantifying the errors associated with the predictions. By employing these metrics, we can better understand the strengths and limitations of the models and make informed decisions regarding their suitability for the WSP tasks.

Additionally, the findings suggest that the CNN-LSTM model has the potential to advance WSP in various applications. Accurate WSPs are crucial for industries such as renewable energy, aviation, and environmental monitoring. The ability to forecast wind speed accurately can optimize energy generation and distribution, improve flight safety, and enhance environmental impact assessments. The CNN-LSTM model's superior performance in the WSP offers practical benefits in these domains and provides an avenue for more precise and reliable decision-making.

However, despite the promising results, it is important to acknowledge the limitations of the study. The performance of the models may vary depending on the specific characteristics of the dataset, such as geographic location, period, time and data quality. Furthermore, the choice of hyperparameters and the architecture of the models can significantly impact their performance. Future research should focus on addressing these limitations by conducting comparative studies across different datasets, exploring different architectural variations, and considering additional factors that may influence the WSPs, such as topography and meteorological conditions.

In conclusion, the interpretation and analysis of the results demonstrate the effectiveness of the CNN-LSTM model in the WSP. Its ability to capture both spatial and temporal features, coupled with its superior performance in terms of accuracy and correlation, makes it a promising approach for the WSP tasks. These findings contribute to the advancement of WSP techniques and highlight the potential for improving decision-making in industries reliant on accurate WSPs. Further research and

experimentation are needed to refine and optimize the models, taking into account the specific characteristics and requirements of different WSP applications.

4.2 Limitations and Future Work

While the CNN-LSTM model shows superior performance, it also presents challenges such as higher computational complexity and longer training times. Scalability and generalizability remain concerns, especially with varied datasets. Future work will focus on addressing these limitations through optimization techniques and exploring ensemble models for enhanced robustness and sensitivity.

References

- Ali, M. H. (2012). *Wind energy systems: Solutions for power quality and stabilization*: Crc Press.
- Bali, V., Kumar, A., & Gangwar, S. (2019). Deep learning based wind speed forecasting- a review. Paper presented at the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India. doi: [10.1109/confluence.2019.8776923](https://doi.org/10.1109/confluence.2019.8776923)
- Bastos, B. Q., Oliveira, F. L. C., & Milidui, R. L. (2021). U-convolutional model for spatio-temporal wind speed forecasting. *International Journal of Forecasting*, 37(2), 949-970. doi: [10.1016/j.ijforecast.2020.10.007](https://doi.org/10.1016/j.ijforecast.2020.10.007)
- Chen, Y., Wang, Y., Dong, Z., Su, J., Han, Z., Zhou, D., Zhao, Y., & Bao, Y. (2021). 2-d regional short-term wind speed forecast based on cnn-lstm deep learning model. *Energy Conversion and Management*, 244, 114451. doi: [10.1016/j.enconman.2021.114451](https://doi.org/10.1016/j.enconman.2021.114451)
- Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). *Language modeling with gated convolutional networks*. Paper presented at the the 34th International Conference on Machine Learning, Sydney, Australia.
- Frazier, P. I., & Wang, J. (2016). Bayesian optimization for materials design. *Information science for materials discovery and design*, 45-75. doi: [10.1007/978-3-319-23871-5_3](https://doi.org/10.1007/978-3-319-23871-5_3)
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *SCIENCE*, 313(5786), 504-507. doi: [10.1126/science.1127647](https://doi.org/10.1126/science.1127647)
- Hochreiter, S., & Schmidhuber, J. u. (1997). Long short-term memory. *Neural Computation* 9(8), 1735-1780. doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735)
- Hou, J., Wang, Y., Zhou, J., & Tian, Q. (2022). Prediction of hourly air temperature based on cnn-lstm, geomatics. *Natural Hazards and Risk*, 13(1), 1962-1986. doi: [10.1080/19475705.2022.2102942](https://doi.org/10.1080/19475705.2022.2102942)
- Jaseena, K. U., & Kovoov, B. C. (2021). Decomposition-based hybrid wind speed forecasting model using deep bidirectional lstm networks. *Energy Conversion and Management*, 234. doi: [10.1016/j.enconman.2021.113944](https://doi.org/10.1016/j.enconman.2021.113944)
- Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., & Menaka, R. (2020). Attention embedded residual cnn for disease detection in tomato leaves. *Applied Soft Computing*, 86, 105933. doi: [10.1016/j.asoc.2019.105933](https://doi.org/10.1016/j.asoc.2019.105933)

- Kusiak, A., Zhang, Z., & Verma, A. (2013). Prediction, operations, and condition monitoring in wind energy. *Energy*, 60, 1-12. doi: **10.1016/j.energy.2013.07.051**
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-based learning applied to document recognition*. Paper presented at the in Proceedings of the IEEE. doi: **10.1109/5.726791**
- Liu, H., Duan, Z., Wu, H., Li, Y., & Dong, S. (2019). Wind speed forecasting models based on data decomposition, feature selection and group method of data handling network. *Measurement*, 148, 106971. doi: **10.1016/j.measurement.2019.106971**.
- Liu, H., Yu, C., Wu, H., Duan, Z., & Yan, G. (2020). A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting. *Energy*, 202, 117794. doi: **10.1016/j.energy.2020.117794**
- Lv, S.-X., & Wang, L. (2022). Deep learning combined wind speed forecasting with hybrid time series decomposition and multi-objective parameter optimization. *Applied Energy*, 311, 118674. doi: **10.1016/j.apenergy.2022.118674**
- Memarzadeh, G., & Keynia, F. (2020). A new short-term wind speed forecasting method based on fine-tuned lstm neural network and optimal input sets. *Energy Conversion and Management*, 213, 112824. doi: **10.1016/j.enconman.2020.112824**
- Neshat, M., Nezhad, M. M., Abbasnejad, E., Mirjalili, S., Groppi, D., Heydari, A., . . . Wagner, M. (2021a). Wind turbine power output prediction using a new hybrid neuro-evolutionary method. *Energy*, 229. doi: **10.1016/j.energy.2021.120617**
- Neshat, M., Nezhad, M. M., Abbasnejad, E., Mirjalili, S., Tjernberg, L. B., Astiaso Garcia, D., Alexander, B., & Wagner, M. (2021b). A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the lillgrund offshore wind farm. *Energy Conversion and Management*, 236, 114002. doi: **10.1016/j.enconman.2021.114002**
- Neshat, M., Nezhad, M. M., Abbasnejad, E., Tjernberg, L. B., Garcia, D. A., Alexander, B., & Wagner, M. (2020). An evolutionary deep learning method for short-term wind speed prediction: A case study of the lillgrund offshore wind farm. *ArXiv, abs/2002.09106*. doi: **10.48550/arxiv.2002.09106**
- Qiao, D., Wu, S., Li, G., You, J., Zhang, J., & Shen, B. (2022). Wind speed forecasting using multi-site collaborative deep learning for complex terrain application in valleys. *Renewable Energy*, 189, 231-244. doi: **10.1016/j.renene.2022.02.095**
- Radhika, Y., & Shashi, M. (2009). Atmospheric temperature prediction using support vector machines. *International Journal of Computer Theory and Engineering*, 55-58. doi: **10.7763/ijcte.2009.V1.9**
- Schepers, J. G., & Snel, H. (2007). Model experiments in controlled conditions. *ECN Report: ECN-E-07-042*, 484.
- Shobana Devi, A., Maragatham, G., Boopathi, K., Lavanya, M. C., & Saranya, R. (2021). *Long-term wind speed forecasting—a review*. Paper presented at the Artificial Intelligence Techniques for Advanced Computing Applications, Singapore. doi: **10.1007/978-981-15-5329-5_9**
- Sun, W., Liu, M., & Liang, Y. (2015). Wind speed forecasting based on feemd and lssvm optimized by the bat algorithm. *Energies*, 8(7), 6585-6607. doi: **10.3390/en8076585**
- TuTiempo.Net. (2010-2021). Global climate data, from <https://en.tutiempo.net/climate>
- Vanderwende, B., & Lundquist, J. K. (2015). Could crop height affect the wind resource at agriculturally productive wind farm sites? *Boundary-Layer Meteorology*, 158(3), 409-428. doi: **10.1007/s10546-015-0102-0**
- Yang, R., Liu, H., Nikitas, N., Duan, Z., Li, Y., & Li, Y. (2022). Short-term wind speed forecasting using deep reinforcement learning with improved multiple error correction approach. *Energy*, 239, 122128. doi: **10.1016/j.energy.2021.122128**
- Yildiz, C., Acikgoz, H., Korkmaz, D., & Budak, U. (2021). An improved residual-based convolutional neural network for very short-term wind power forecasting. *Energy Conversion and Management*, 228, 113731. doi: **10.1016/j.enconman.2020.113731**