



A New Convolutional Neural Networks and LSTM Based model for Realtime Wind Forecasting

Zahra T. Baghini^{1*}

1. Department of Computer Engineering, Kerman branch, Islamic Azad University, Kerman, Iran.

ARTICLE INFO	ABSTRACT
Article type: Research Paper	Nowadays, renewable energy resources like wind energy generation are the most important and critical issues for fossil fuel replacement and environmental rescue. Although recent wind power generation has developed rapidly, the planning of these generations is still difficult because the accurate forecasting of wind speed is challenging due to its volatile and uncertain nature, particularly in coastal areas. Therefore, this study proposes a real-time wind speed forecasting model based on a hybrid convolutional neural network architecture, where multiple meteorological variables are filtered through a cross-screening strategy for multivariate time series forecasting. To screen the parameters for training, an evaluation was conducted based on multi-feature wind speed datasets. The performances of the proposed and baseline models in terms of real-time wind speed forecasting accuracy and volatility were evaluated based on a comparative study of error metrics. To benchmark the results, the proposed model was compared with other state-of-the-art paradigms based on error metrics. The results indicate that the proposed CNN-LSTM model outperformed all baseline models in terms of training efficiency and delivered values with higher accuracy than the other models. Based on different test datasets, the CNN-LSTM model with the same hyperparameter configurations predicted more accurately than other baseline models under the collected datasets. For 10-meter datasets the RMSE index obtained as 0.25, 0.42 and 0.25. Also, for 50-meter datasets the RMSE index obtained as 0.26, 0.51 and 0.25. These results prove the proposed model accuracy.
Article History: Received 7 August 2026 Accepted 7 September 2026	
Keywords: Wind speed forecasting, Convolutional Neural Networks, Long Short-Term Memory, Feature Selection.	
Cite this article:	
Baghini, Z. (2026). A New Convolutional Neural Networks and LSTM Based model for Realtime Wind Forecasting. <i>Energy Systems: Technology and Management</i> , 2 (1), 79-95. DOI: http://doi.org/10.48306/estm.2026.595855.1020 .	



© Zahra T. Baghini / **Publisher:** Graduate University of Advanced Technology.
DOI: <http://doi.org/10.48306/estm.2026.595855.1020>.

* Corresponding author.

Email: koohkan.farhad2020@gmail.com

1. Introduction

As the global population continues to grow and industry rapidly develops, renewable energy sources are becoming increasingly important. Wind power has attracted significant attention globally as a sustainable and clean energy source. In contrast, conventional energy sources such as fossil fuels have led to serious environmental problems such as the greenhouse effect, water contamination, and air pollution. The safety and waste disposal of nuclear energy have become a constant focus of disputes. In many countries, wind power has been actively promoted and developed by local governments [1].

Wind speed is the main factor that determines the rotational speed of a wind turbine rotor. By accurately predicting the wind speed, wind power plants can better plan and optimize the operation of wind turbines [2-3]. However, the intermittent and nonstationary nature of the wind speed leads to difficulties in reliable wind speed forecasting processes [4]. The time horizon for wind-speed forecasting can be categorized into various types, with forecasts within 6 h belonging to real-time forecasting [5]. Wind-speed forecasting at different time horizons can assist wind power plants in scheduling wind power generation tasks [6].

Currently, the main methods for predicting wind speed are divided into four types: physical, statistical, artificial intelligence, and hybrid model-based predictions [7-8]. Physical models can predict long-term wind speeds using numerical weather prediction (NWP)-based meteorological physics simulations. However, these methods require a large amount of data and time [9-10]. Statistical models can make predictions utilizing only historical wind speed data. However, the accuracy of these models decreases as the forecasting range increases [11]. Artificial intelligence models have evolved from traditional machine learning models, such as support vector machines [12], extreme learning machines [13], and widely accepted deep learning models in recent years, such as recurrent neural networks (RNNs), gated

regulated unit networks (GRUs) [14-15], long short-term memory networks (LSTM) [16] and convolutional neural network (CNN) [17-19]. Notable progress and breakthroughs have been made in the field of wind speed forecasting owing to these deep neural network models.

However, it is challenging to apply a single neural network model across various scenarios for wind-speed forecasting. For instance, when dealing with complex wind-speed datasets, the predictive accuracy of models based on a single network structure often experiences a significant decline [20-21]. Therefore, researchers have shown a strong interest in improving the performance of wind speed forecasting models using hybrid models [22]. For example, [23] proposed a hybrid model, SSA-MEMD-ACNN-BiLSTM, that considers both historical and future information of time-series data and forecasts wind speed accurately. Li et al. [24] utilized the MODWT to decrease volatility the original wind speed dataset and combined it with the LSTM algorithm to build a hybrid model to forecast wind speed. Jiang et al. [25] proposed a multistep wind speed forecasting model for a convolutional gated recurrent unit network based on quadratic decomposition (SD) and a multilabel-specific XGBoost feature selection method. Hybrid models generally combine deep learning algorithms with data processing techniques, such as noise reduction [26] and feature extraction [27] to improve forecasting performance. The performance of the hybrid model is highly correlated with the structure of its sub-models. Most of the aforementioned hybrid models employ recurrent networks, such as GRU or LSTM. These types of network structures require computations from previous time steps, leading to a significant increase in the computation time as the sequence length increases. Moreover, the problem of vanishing gradients makes it more challenging for models to learn long-term dependencies [28-29].

Recently, the CNN-LSTM architecture [30-31] has led to significant advancements in natural language processing (NLP) and computer vision. It has even replaced RNN models in the NLP domain, as seen in popular

models such as GPT and BERT. To deal with feature extraction and time-series data forecasting in coastal wind speed forecasting, the present study proposed a hybrid model based on the CNN-LSTM architecture for a high degree of forecasting accuracy. The forecasting results of the proposed and baseline models (GRU, LSTM, CNN, CNN-GRU, and CNN-LSTM) were compared and analyzed using high-quality wind speed datasets from three cities (Dataset1, Dataset2, and Dataset3) in the southeast coastal region of China. Moreover, we tested and evaluated the volatility performance of the proposed model against baseline models by introducing artificial volatility into the datasets .

The main innovations of this work are as follows:

This study combined the feature selection with cross-validation to determine the feature vectors that have the highest coefficient and are selected as inputs of the forecasting model. This strategy reduced the computational cost of model training and testing and ensured the accuracy of the final predictions.

This study proposed a new real-time wind speed forecasting model based on a hybrid CNN-LSTM architecture. The forecasting performance of the proposed model was analyzed and compared with other models for wind speeds.

The obtained results outperform other recent models .

The rest of the paper is organized as follows. Section 2 describes the methodology adopted to carry out the above research process. Section 3 describes the experiments and analyzes the results. Finally, Section 4 summarizes the conclusions of the study.

2. Methodology

The deep learning methods and coupling process of the proposed hybrid model are described in this section. Specifically, the CNN is described in Section 2.1, the CNN-LSTM is described in Section 2.2, the modeling and coupling process of the CNN-LSTM hybrid model are described in Section 2.3, and finally, an overview of the

implementation steps is provided in Section 2.4.

2.1. CNN

CNNs are a special type of feed-forward neural network and deep-learning models constructed using this method require significantly fewer parameters for training because of weight sharing and local connectivity properties [32]. CNNs extract image features and perform classification or regression using components, such as convolutional, pooling, and fully connected layers. CNNs are commonly adopted to process image data but can also be applied to time-series forecasting tasks. With appropriate data preparation and model construction, CNNs can be used to process time-series data and extract key features. CNNs use components such as convolutional and pooling layers to capture local patterns and real-time dependencies in time-series data through local awareness and weight-sharing mechanisms. This enables the CNN to perform feature extraction and efficiently predict the output of the time-series. Each neuron in the convolutional layer extracts the local features by performing a convolution operation on the input image. The formula is as follows:

$$y_i = \sigma \left(\sum_j W_{ij} * x_j + b_i \right) \quad (1)$$

where y_i is the output of the i th neuron in the convolutional layer, σ is the activation function, W_{ij} is the weight matrix, $*$ is the convolution operation, x_j is the input feature image, and b_i is the bias term.

The pooling layer was designed to reduce the size of the feature map and retain key features. Maximum and average pooling operations are commonly applied. The formula is as follows:

$$y_i = \max(\text{Pool}(x_i)) \quad (2)$$

where y_i is the output of the i th neuron in the pooling layer, pool is the pooling operation, and x_i is the input feature map.

A fully connected layer was used to map the output of the pooling layer to the final output category or regression results. The formula used is as follows:

$$y = \sigma(W \cdot x + b) \quad (3)$$

where y is the output of the fully connected layer, σ denotes the activation function, and W is the activation function, and W is a weight matrix. x is the input vector, and b is the bias term.

With components including convolutional and pooling layers, CNN can automatically extract features with local patterns and temporal coefficients from wind speed time-series data, rather than manually designing the features. A CNN can effectively capture local structures and important patterns in wind-speed time-series data by utilizing the mechanisms of weight sharing and local perceptibility. In addition, CNNs have fewer parameters and lower computational complexity, making them suitable for processing large-scale time-series data. With the introduction of multilayer stacking and deep learning techniques, CNNs can adaptively extract multi-horizon time-series features and construct complex wind speed time-series feature extraction models. The internal structure of the CNN is shown in Fig. 1.

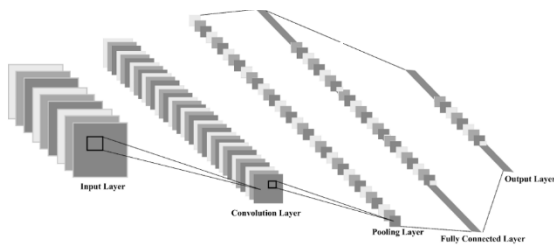


Fig. 1. CNN internal structure.

2.2. CNN-LSTM structure

A CNN-LSTM is a sequence processing model Fig 3. It calculates the weights of each feature in the input by using MI-IG methods. The core of the CNN-LSTM is an encoder-decoder layer structure, where both the encoder and decoder consist of multiple stacked layers. Each layer is composed of a self-attention sublayer and a feedforward

neural network sublayer [33]. The self-attention sublayer computes the attention weights via linear transformations of queries, keys, and values, and multiplies the weights by the values to obtain the self-attention representation. The feedforward neural network sublayer performs a non-linear transformation of the self-attention representation using a multilayer perceptron. By stacking multiple encoder layers, the CNN-LSTM model can capture long-range dependencies within the input sequence, thereby achieving significant results in time-series processing tasks Fig 2 .

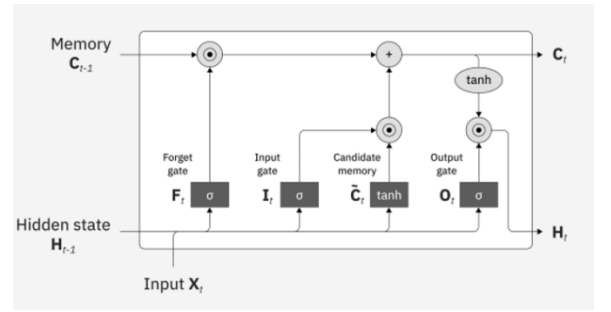


Fig. 2. LSTM internal structure.

2.3. Hybrid model based on CNN-LSTM

An innovative hybrid model for wind speed time-series forecasting is proposed by coupling a CNN and LSTM in tandem. The hybrid model takes full advantage of the strength of the CNN in capturing local features and time-series patterns and the benefits of the LSTM in modeling global dependencies and long-term temporal coefficients. The proposed model places a CNN at the head end of the LSTM to acquire input samples with the multiple matrices obtained after preprocessing. A CNN learns the original features of the samples and outputs a sequence of high-level features. This sequence can be converted into an input format that is acceptable to the CNN-LSTM. This coupling allows the model to combine local and global information to capture the features and dependencies of temporal data more comprehensively.

The structure of the CNN-LSTM hybrid model is schematically shown in Fig. 3.

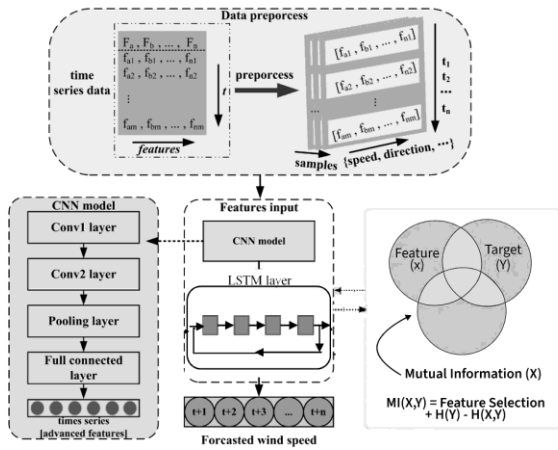


Fig. 3. Proposed Model.

2.4. Implementing procedure

The methodological steps of this study to facilitate wind speed forecasting are as follows:

1 (Data Collection: Wind speed datasets were collected from the NASA website for three cities (Dataset1, Dataset2, and Dataset3) from 01/27/2023 to 06/01/2023 [34]. Each original dataset contained 10 feature classes: wind speed and wind direction at 10/50 m, temperature, dew point temperature, precipitation, specific humidity, relative humidity, surface pressure, and temperature at 2 meters.

2 (Feature Selection: The datasets were analyzed for feature selection by MI-IG algorithm. The features with the highest coefficients were selected as input parameters. The selected input features for the model include wind speed and wind direction at 10/50 m, temperature, dew point temperature, precipitation, and relative humidity, all sampled at the same time intervals.

3 (Data Preparation: The dataset was divided into training, validation, and test sets at a ratio of 8:1:1. Hourly features were extracted from the divided dataset, normalized, and overlapping sliding windows were applied to obtain the data samples for prediction. This ensured that each input feature vector corresponds to the same time interval, aligning the time-series data for accurate forecasting.

4 (Handling Input Features: To address the issue of using actual future values, our model relies on past observed values for input features:

Step 1: For wind speed forecasting at different heights in different cities, six different feature vector combinations shown in Tables 3 to 5 are used as input vectors, and the sliding time window size is set to 24 hours.

Step 2: These past observed values are used to predict the wind speed 1 hour ahead.

This ensures that the model does not depend on future actual values, which are not available at the forecasting time.

5 (Model Construction: A hybrid CNN-LSTM model was built using Pytorch. The CNN was used to extract local features and real-time dependencies from the time-series data, while the CNN-LSTM was employed to capture global dependencies and long-term temporal coefficients. The model structure is shown in Fig. 3.

6 (Training and Hyperparameter Tuning: The model was trained using the training set data, and hyperparameter tuning was performed on the validation set. An early stopping strategy was adopted to prevent overfitting, stopping training when the validation performance did not improve for 30 consecutive epochs.

7 (Forecasting and Evaluation: The trained model was used to predict the wind speed 1 hour ahead. The performance of the proposed model was compared with other benchmark models using metrics such as RMSE, MAE, and R^2 . The output of the model is the predicted wind speed 1 hour ahead.

8 (Volatility Test: To evaluate the robustness of the model, Gaussian noise was added to the dataset, and a volatility immunity test was performed. The model's performance on the noisy data was compared with its performance on the original data.

3. Experiments

The performance of the proposed model was compared with that of the other model using wind-speed measurements from three cities in southeastern China. The experimental procedure and analysis of the results are presented in Section 3.1.

3.1. Data acquisition and preprocessing

Selecting the most effective input features is necessary to control the computational cost of model training and testing and to ensure the accuracy of the final forecasting results. This study applied the feature selection method [35], which had the highest coefficient with the predicted output, using Pearson and Spearman matrices to cross-validate the selected features for MI-IG analysis. The two matrices measure the strength of the relationship between each pair of variables using the MI-IG coefficients.

Table 1. Dataset's feature description.

Symbol	Description	Unit
WS10M	Wind Speed at 10 Meters	<i>m/s</i>
WD10M	Wind Direction at 10 Meters	degree
WS50M	Wind Speed at 50 Meters	<i>m/s</i>
WD50M	Wind Direction at 50 Meters	degree
PRECTOTCORR	Precipitation Corrected	<i>mm/hour</i>
QV2M	Specific Humidity at 2 Meters	<i>g/kg</i>
RH2M	Relative Humidity at 2 Meters	%
PS	Surface Pressure	<i>kPa</i>
T2M	Temperature at 2 Meters	°C
T2MDEW	Dew/Frost Point at 2 Meters	°C

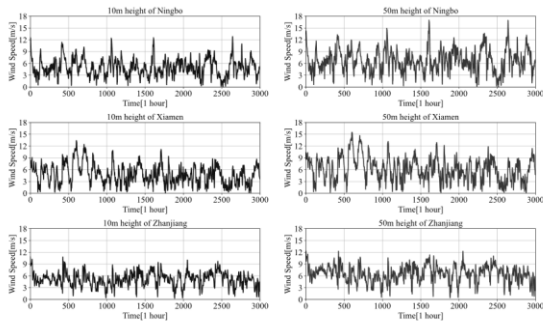


Fig. 4. Wind speed recordings from 08:00 on 01/27/2023 to 07:00 on 01/06/2023 (UTC).

Their values are in the interval $[-1,1]$, with values closer to 0 representing a lower coefficient.

The degree of coefficient of the features among the wind speed datasets of the three selected cities is shown in Table 2. This indicates that for wind speeds at different heights in each city, the features with a higher degree of coefficient differed. With comprehensive consideration, six features with the highest coefficient for each type of wind-speed data were selected as input features for the forecasting model. The subsets of features selected for the three datasets are shown in Table 2. Therefore, all the models utilized in this study were considered multivariate input time-series forecasting models. Owing to the different measurement ranges of each feature, normalization was required to unify the value scales between the features during training. The maximum-minimum normalization method was selected, where the values corresponding to all features are scaled to the interval $[0, 1]$. This can provide a certain degree of denoising for the input data, improve the convergence speed, and enhance the generalization ability of the model. The formula for the maximum normalization method is as follows [36]:

$$x_{\text{scaled}} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4)$$

Table 2. Selected features dataset.

Predicted object	Selected features	Pearson coefficient	Spearman coefficient
WS10M	WS10M	1	1
	WS50M	0.98	0.98
	WD10M	-0.47	-0.48
	WD50M	-0.47	-0.47
	T2MDEW	-0.45	-0.47
	QV2M	-0.44	-0.47
WS50M	WS10M	0.98	0.98
	WS50M	1	1
	WD10M	-0.46	-0.48
	WD50M	-0.45	-0.47
	T2MDEW	-0.44	-0.46
	QV2M	-0.43	-0.46

The following error metrics were used to evaluate the performance of all the predictive models:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (5)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (6)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (\hat{y}_i)^2} \quad (8)$$

where y_i is the actual observed value, $\hat{y}_i$ is the predicted value of the model, and n is the number of samples. MSE measures the mean squared error between the observed and predicted values, with smaller values indicating more accurate predictions of the model. The MSE is sensitive to outliers and is the squared unit of the target variable, which is used in the validation set to assess the training performance of the model. The RMSE is the square root of the MSE, which eliminates the MSE unit problem and treats the unit of the target variable as the unit of the indicator. The lower the RMSE, the more accurate the forecasting of the model. MAE measures the mean absolute error between the observed and predicted values, which is the average of the absolute values of the errors. MAE is insensitive to outliers that are in the same unit as the target variables. R^2 represents the

explanatory ability of the model for the observed values. R^2 ranges from 0 to 1; the closer it is to 1, the better the explanatory ability of the model to the observed values.

3.1.1. Experimental condition and preparation

This study divided the preprocessed dataset into training, validation, and test sets at a ratio of 8:1:1. The training set was used to train the model and the validation set was used to monitor the degree of improvement in the model performance and convergence of the loss decline. An early stop strategy was adopted to stop the training of the model and prevent overfitting. Specifically, we set a relatively reasonable threshold parameter of 30 iterations for the model on the validation set and stopped training when the number of consecutive training iterations exceeded 30 without any improvement in the model performance. The test set was a unique dataset from the training and validation sets used to test the forecasting performance and generalization ability of the model. Before the formal start of the experiment, some pre-tests were performed to adjust the relatively optimal hyperparameter combinations of each model based on the current computer. The descriptions and configurations of the hyperparameters of each model are shown in Table 3.

Table 3. Model architecture and hyper-parameter configurations.

Hyperparameters	LSTM	GRU	CNN	CNN-GRU	LSTM-GRU	CNN-LSTM
Hidden layer size	64	32	N/A	32	32	64
Hidden layers	2	2	N/A	2	2	2
Multi-heads of self-attention	N/A	N/A	N/A	4	N/A	N/A
Self-attention layer size	N/A	N/A	N/A	32	N/A	N/A
Output channels	N/A	N/A	8	8	8	8
Kernel size	N/A	N/A	6	6	6	6
Stride	N/A	N/A	2	2	2	2
Padding	N/A	N/A	1	1	1	1
Dropout	0.1	0.1	0.1	0.1	0.1	0.1
Learning rate	0.0001	0.0001	0.0003	0.0003	0.0001	0.0003

All the optimizers applied for the models were AdamW, the number of iterations per training epoch was 200, the batch size for

processing the sample data was 256, the amount of actual data used for forecasting each time, namely, the sliding window size was 24.

The segmentation of the sample data and forecasting of the time-series data are shown in Fig. 5.

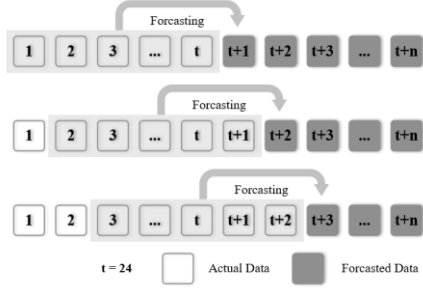


Fig. 5. Sliding window method for direct forecasting.

All forecasting models used for the experiments in this study were built based on the Pytorch [37] framework with the following computer configurations: the CPU was an AMD Ryzen 9 5900HX, the RAM was 32 GB, and the graphics card was an NVIDIA GeForce RTX 3070 with 8 GB of video memory.

3.1.2. Experiment 1: Wind speed forecasting based on normal dataset of 10 m height

Based on the previous preparations, the proposed model and the baseline model were iteratively trained for each dataset. When the learning process was completed, the training and validation losses dropped to close to 0. It

Table was used and the forecasting accuracies were similar. This proves that the generalization of the proposed model is superior.

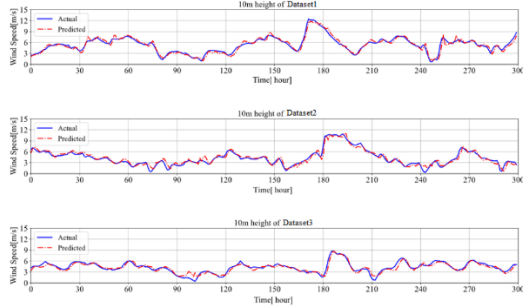


Fig. 7. Predicted wind speed on 10 m height by CNN-LSTM.

The regression plots of the training and test sets shown in were used to further validate the

was showed in Fig. 6. This indicates that the model had no overfitting or underfitting phenomena in either the training or validation sets and that the training effect was adequate. It was showed in Fig. 6. This indicates that the model had no overfitting or underfitting phenomena in either the training or validation sets and that the training effect was adequate.

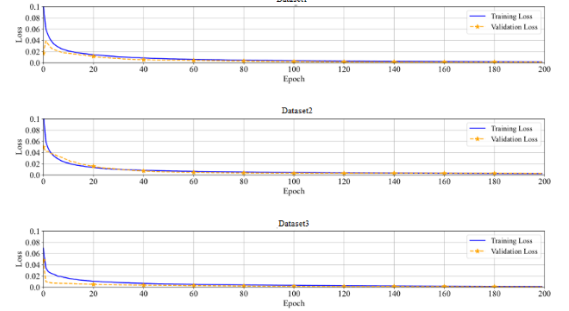


Fig. 6. The variation of the loss function versus epochs of wind speed on 10m height.

The results of the 1 h wind speed forecasting for the three cities at 10 m height utilizing the well-trained CNN-LSTM model are illustrated in Fig. 77. The predicted trends exhibit relatively large deviations from the actual values for only 10 h of each 300h test set, while the rest of the values and trends are largely fitted to those in the original data. For the different test sets, the proposed model with the same structure as that shown in

training and testing effects of the proposed model in Fig. 8. If the predicted data points in the plots are centrally distributed around the regression line $y=x$, the data consistency between the actual and predicted wind speeds can be proved. The figure shows that for the training set in each dataset, the degree of regression of the predicted points was not equal to that of the predicted points on the test set. This is due to the introduction of a validation-set-based early stopping strategy that prevents the model from overtraining on the training set and overfitting on the test set. The results demonstrated that the strategy used was successful and the trained model exhibited adequate accuracy and generalization on the test set and showed no overfitting.

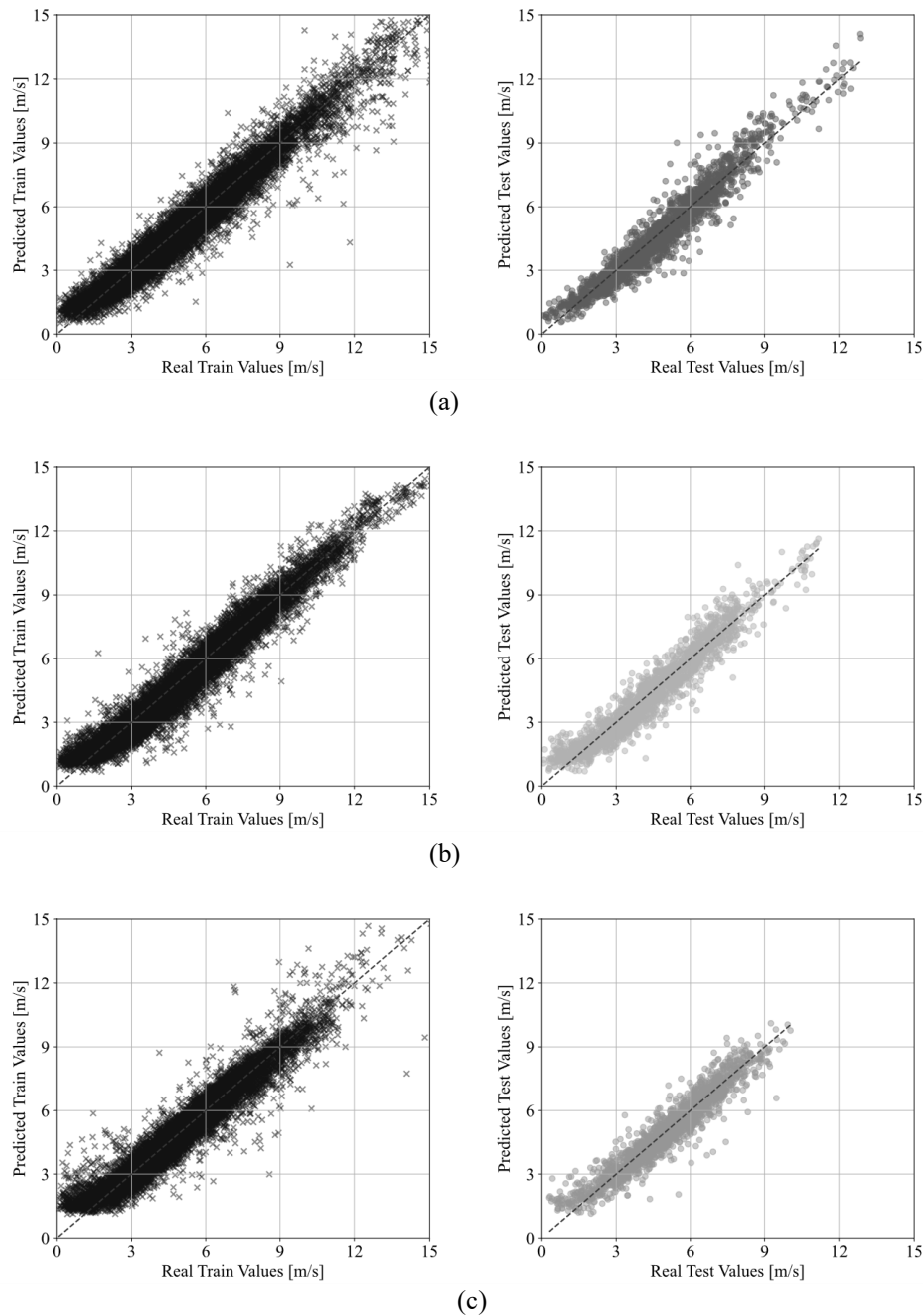


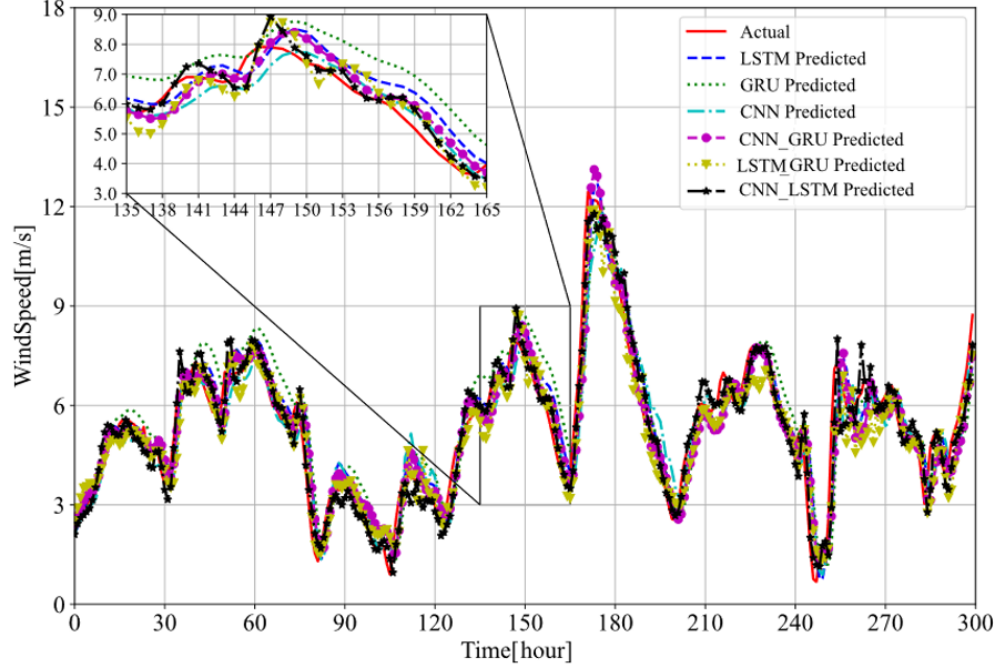
Fig. 8. Regression plot between real and predicted values for the training and test datasets of 10 m height wind speed for (a) Dataset1, (b) Dataset2, (c) Dataset3.

Fig 9 shows the forecasting results of the proposed hybrid model and each baseline model for the 10 m height wind speed of the three cities. From the figure, it is evident that the forecasting values of the models for the wind speeds in the test set were fitted to the actual values in the forecasting range of 300 h. However, the forecasting results for 30 h scaled up from the median interval show that the forecasting accuracies of the models are different. In this range, the GRU model

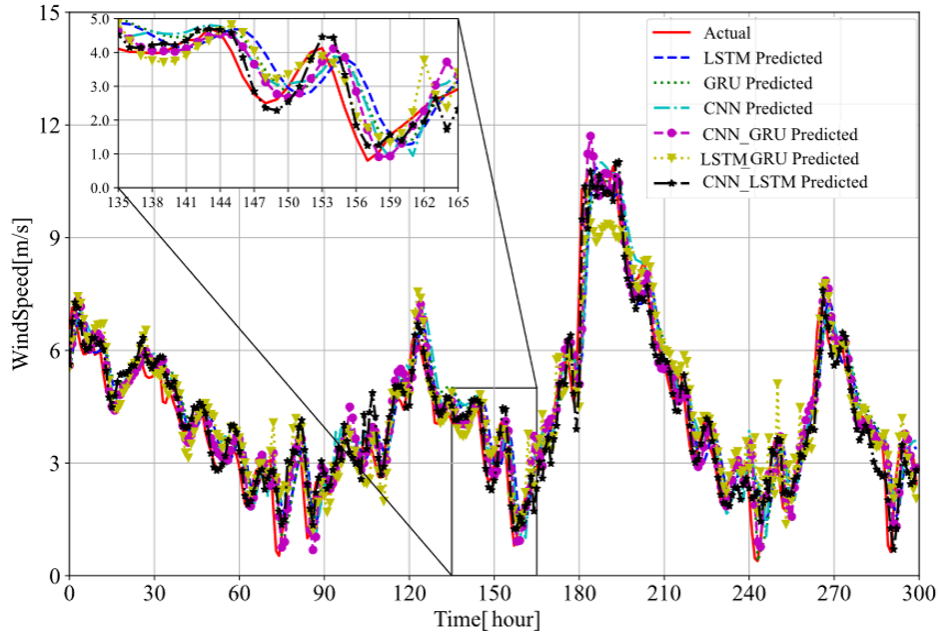
performed poorly in the Dataset1 dataset; although the predicted trends were similar to the real data, the values deviated more than those of the other models. In the Dataset2, the LSTM model presented poor predictions overall and the CNN-GRU model showed the largest bias in the 160–163 h interval. Furthermore, the forecasting trends of all models in the Dataset3 for 135–153 h are not as efficient as the actual values in the previous two datasets. However, the predictions of the

proposed model were closer to the actual values than those of the other models. This proves that the self-attention mechanism within the CNN-LSTM sub-model works,

gradually learning and adjusting the correct forecasting trend based on the long-term dependency of previous data.



(a)



(b)

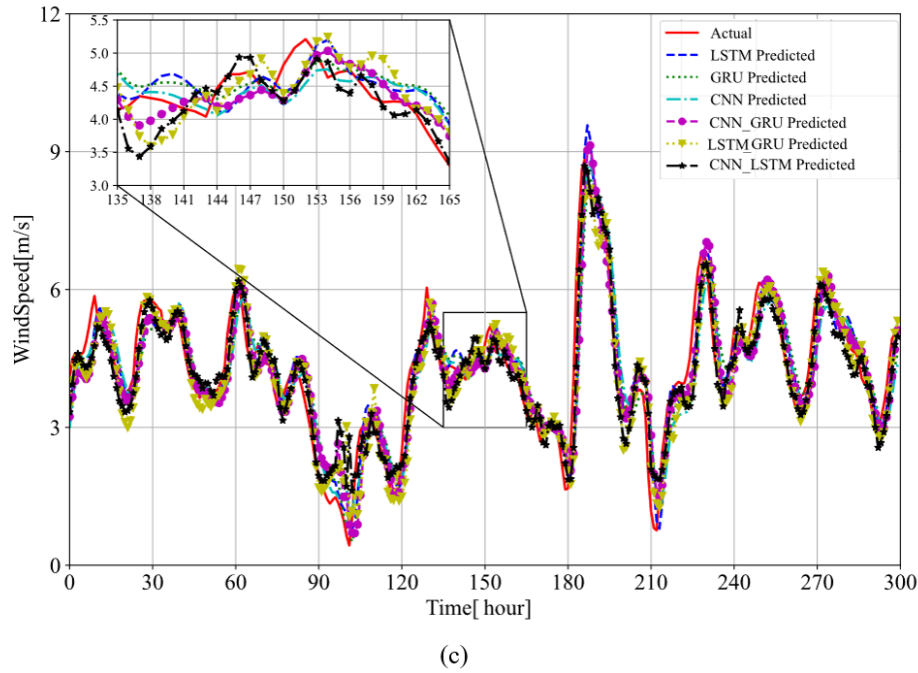


Fig. 9. Forecasting results comparison of the proposed model and baseline models for 10 m height wind speed for (a) Dataset1, (b) Dataset2, (c) Dataset3.

The differences in forecasting performance are listed in Table 5. The proposed CNN-LSTM hybrid model in this study has the best results for all three types of error evaluation metrics (RMSE, MAE, and R^2) for the three datasets compared with other baseline models. The R^2 values of the CNN-LSTM model on the three test sets were 0.9218, 0.9180, and 0.9027, respectively, whereas the R^2 values of

the other baseline models were less than those of the other models. This indicates that the proposed model had the best ability to explain the forecasting of the time-series data. In terms of numerical quantification, the superior predictive accuracy and generalization ability of the proposed model for the three datasets were proven.

Table 4. Results of benchmark probabilistic forecasting models to 10 m wind speed.

CASE	Evaluation metrics	LSTM	GRU	CNN	CNN-GRU	LSTM-GRU	CNN-LSTM
Dataset1	RMSE	0.4174	0.4664	0.4577	0.2876	0.2770	0.2547
	MAE	0.2450	0.2842	0.2704	0.2185	0.2267	0.1714
	R^2	0.8750	0.8451	0.8415	0.8771	0.8728	0.7218
Dataset2	RMSE	0.4620	0.4222	0.4455	0.4546	0.4810	0.4286
	MAE	0.4577	0.4456	0.4442	0.2705	0.2860	0.2567
	R^2	0.8400	0.8652	0.8508	0.7048	0.8887	0.7180
Dataset3	RMSE	0.2724	0.4078	0.4021	0.2704	0.2771	0.2514
	MAE	0.2006	0.2427	0.2268	0.2154	0.2041	0.1847
	R^2	0.8861	0.8540	0.8572	0.8707	0.8821	0.7027

3.1.3. Experiment 2: Wind speed forecasting based on volatility of 50 m height

The primary difference between Experiments 1 and 2 lies in the data used and the objective of each experiment. Experiment 1 focuses on evaluating the forecasting

performance of the CNN-LSTM model using clean, unaltered wind speed data at a height of 10 meters from three cities. In contrast, Experiment 2 aims to assess the robustness and volatility of the model by introducing Gaussian noise into the wind speed data at a height of 50 meters, simulating real-world sensor volatility.

This comparison highlights the model's ability to maintain forecasting accuracy under different conditions, with Experiment 1 serving as the baseline and Experiment 2 testing the model's performance in noisy environments.

Wind speed data measured at ground stations generally contain Gaussian or uniform white noise [37]. Gaussian noise was added to the original dataset as a disturbance. Gaussian Noise is a type of random noise that obeys a Gaussian (normal) distribution and can closely simulate random errors and uncertainties generated by sensors in the real world. This is represented by the following mathematical formula:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (9)$$

where x is a random variable (volatility value), μ is the mean (expected value) of volatility, and σ is its standard deviation. The relative strength of the signal-to-noise ratio (SNR) in a noisy dataset can be expressed as the SNR, which is generally measured in decibels (dB). SNR was calculated using the following formula:

$$SNR = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (10)$$

where P_{signal} is the power of the signal and P_{noise} is the power of the volatility. A logarithmic operation of 10 bases was used. A noisy dataset was tested for volatility using each model. A suitable SNR for a noisy dataset containing Gaussian noise is 40 dB[39].

Fig. 1 shows a comparison between the original and volatility of the 50 m height wind

speed in the three cities after introducing Gaussian volatility. The noisy dataset appears to have more complex fluctuation trends and numerical distributions than the original dataset. In this section, the volatility dataset is used to perform a new round of training for the proposed hybrid model and other baseline models according to the hyperparameter configurations and experimental steps previously implemented.

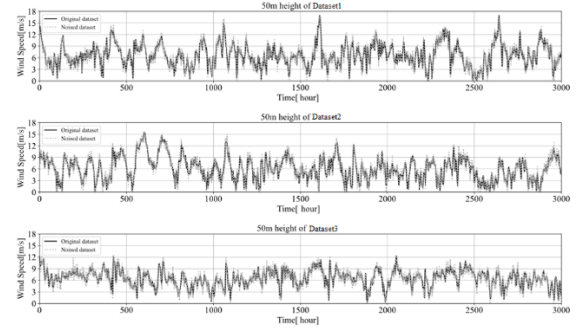
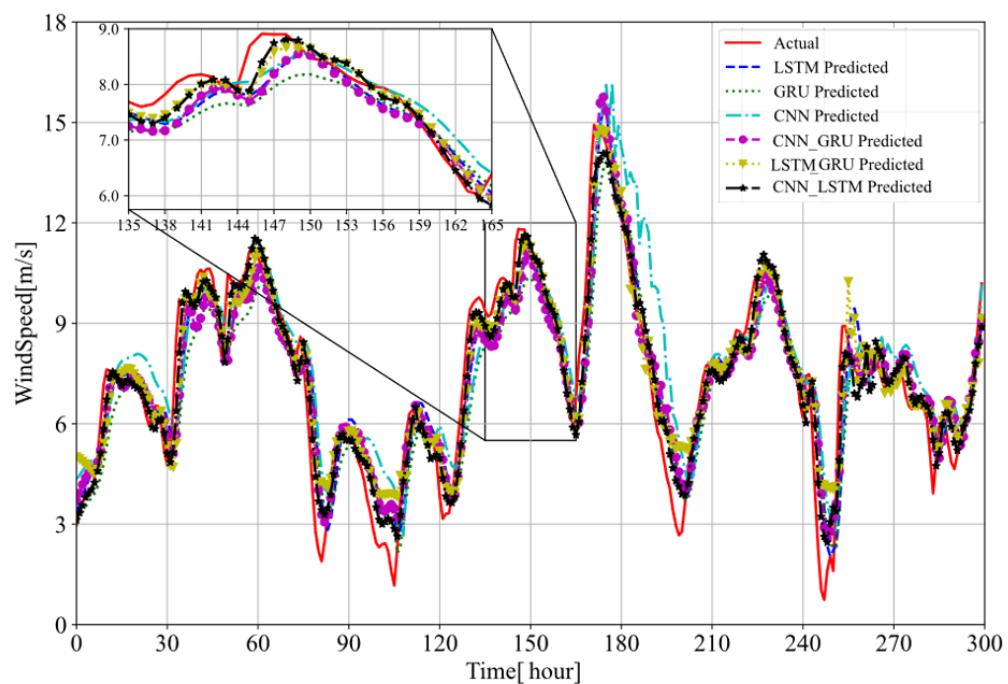
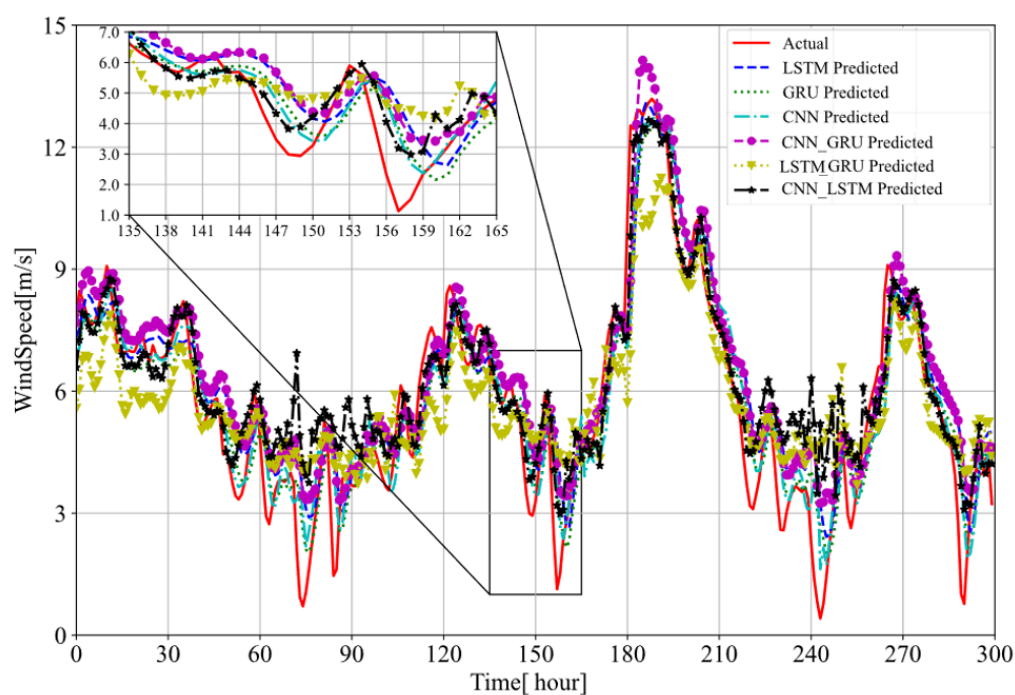


Fig. 1. Comparison between origin dataset and volatility dataset for 50 m height wind speed.

Next, we assessed the forecasting performance on the original data from the three cities using a forecasting model trained on noisy data. The final forecasting results are shown in Fig. 2. As shown, the overall fit of the forecasting results of the models was significantly worse than that of the models previously trained using the original dataset. The predicted trends of the models were similar to those of the real data, but the predicted values had relatively large deviations. This proves that the Gaussian volatility introduced in this volatility immunity test is reasonable and neither overly changes the trend of the original data nor disturbs the actual forecasting results to a certain extent.



(a)



(b)

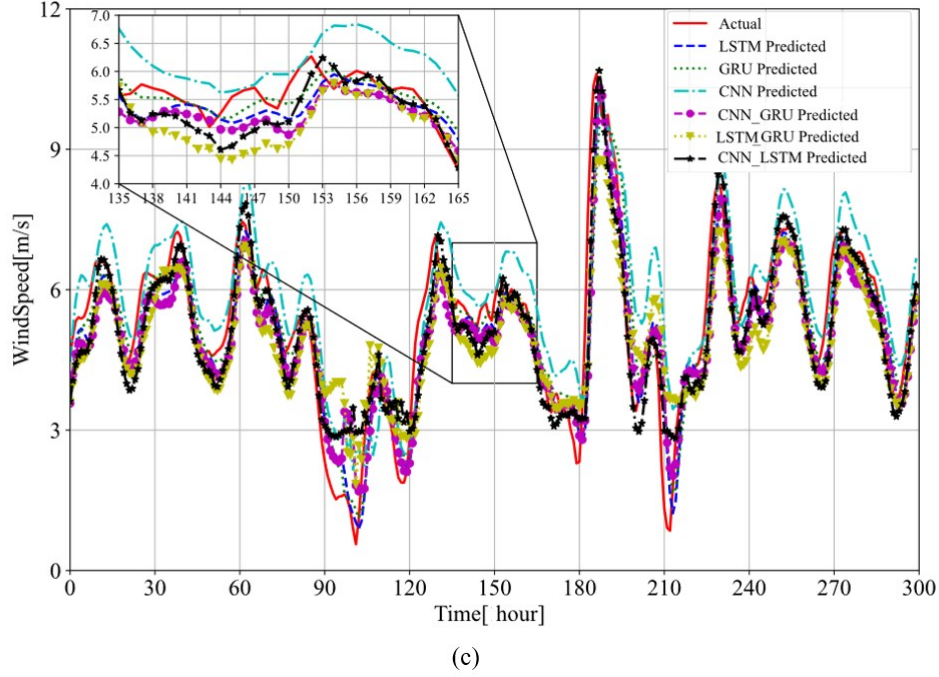


Fig. 2. Forecasting results comparison of the proposed model and baseline models for 50 m height wind speed in volatility dataset of (a) Dataset1, (b) Dataset2, (c) Dataset3.

The results show that the different models have different resistances to the same volatility; the specific differences are shown in Tables 5-7. As can be seen, the forecasting performance of both the proposed model and the baseline model decrease; in particular, the forecasting accuracy of CNN for the Dataset1 dataset and LSTM, CNN-LSTM, and CNN-GRU for the Dataset2 dataset dropped below 80%. The CNN-LSTM hybrid model showed the least degradation in performance, with

accuracy losses of 1.38%, 1.11%, and 0.47% for R^2 on the three datasets. The proposed model demonstrated superior accuracy in wind speed forecasting at a height of 50 m and minimal sensitivity to volatility compared to other baseline models. As a result, it exhibited exceptional predictive capability. Moreover, the experimental findings unequivocally establish that the hybrid CNN-LSTM model presented in this study possesses unparalleled robustness compared with all other models.

Table 5. Comparison of benchmark probabilistic forecasting models to 50 m height wind speed of noisy dataset.

CASE	Evaluation metrics	LSTM	GRU	CNN	CNN-GRU	LSTM-GRU	CNN-LSTM
Dataset1	RMSE	0.5035	0.5306	0.5961	0.3755	0.3858	0.3930
	MAE	0.3132	0.3352	0.3885	0.3893	0.3857	0.3235
	R^2	0.8553	0.8331	0.7657	0.8656	0.8567	0.9185
Dataset2	RMSE	0.5336	0.5657	0.5663	0.5363	0.5938	0.5315
	MAE	0.5089	0.3555	0.3606	0.5150	0.5591	0.3160
	R^2	0.7891	0.8398	0.8385	0.7871	0.7500	0.8630
Dataset3	RMSE	0.3037	0.3119	0.3593	0.3003	0.3335	0.2593
	MAE	0.3315	0.3261	0.3717	0.2395	0.2579	0.1899
	R^2	0.8531	0.8539	0.8037	0.8557	0.8335	0.8905

Table 6. Comparison of benchmark probabilistic forecasting models to 50 m height wind speed of original dataset.

CASE	Evaluation metrics	LSTM	GRU	CNN	CNN-GRU	LSTM-GRU	CNN-LSTM
Dataset1	RMSE	0.3553	0.3660	0.3537	0.3990	0.3369	0.2663
	MAE	0.3688	0.3783	0.2656	0.2303	0.2578	0.1985
	R ²	0.8796	0.8735	0.8871	0.9157	0.8910	0.9333
Dataset2	RMSE	0.5593	0.5298	0.5513	0.5275	0.5226	0.5101
	MAE	0.3559	0.3255	0.3357	0.3213	0.3166	0.3121
	R ²	0.8535	0.8626	0.8556	0.8655	0.8662	0.8731
Dataset3	RMSE	0.2801	0.2920	0.2676	0.2870	0.2817	0.2550
	MAE	0.2029	0.2122	0.1950	0.2093	0.2037	0.1870
	R ²	0.8755	0.8623	0.8855	0.8681	0.8730	0.8951

Table 7. Forecasting accuracy loss of 50 m height wind speed between the noisy dataset and the original dataset.

CASE	Evaluation metrics	LSTM	GRU	CNN	CNN-GRU	LSTM-GRU	CNN-LSTM
Dataset1	RMSE	5.71%	6.56%	7.55%	7.55%	5.89%	2.57%
	MAE	5.35%	5.69%	12.38%	5.91%	2.79%	2.50%
	R ²	-3.53%	-5.95%	-12.25%	-5.91%	-3.53%	-1.38%
Dataset2	RMSE	7.53%	3.59%	2.50%	10.88%	17.02%	2.15%
	MAE	6.50%	2.90%	2.59%	9.27%	13.25%	0.39%
	R ²	-5.53%	-2.28%	-1.71%	-7.73%	-12.62%	-1.11%
Dataset3	RMSE	2.26%	1.99%	8.16%	1.33%	5.18%	0.52%
	MAE	1.85%	1.39%	7.67%	2.01%	5.52%	0.29%
	R ²	-2.13%	-1.95%	-8.17%	-1.25%	-5.05%	-0.57%

4. Conclusion

This study proposed a new forecasting model based on a hybrid CNN-LSTM architecture for wind speed for 1 h ahead and the results are compared with those of five baseline models (LSTM, GRU, CNN, LSTM-GRU, and CNN-GRU). All models used the six most suitable meteorological variables as input parameters, which were selected by cross-screening the Pearson and Spearman matrices.

In this study, the proposed model and other baseline models were applied to a case study using wind speed datasets collected from NASA at heights of 10 and 50 m in three different coastal cities in southeastern China: Dataset1, Dataset2, and Dataset3. The study performed qualitative evaluations of the forecasts and volatility immunity using a graphical analysis of the wind speed predictions for each model. The performances of the models were assessed using the

statistical metrics RMSE, MAE, and R², along with a quantitative evaluation of the accuracy loss due to volatility. Based on the proposed methodology and present results, the proposed hybrid CNN-LSTM structure was effective for real-time wind speed forecasting with multiple meteorological variables as inputs. Based on different test datasets, the CNN-LSTM model with the same hyperparameter configurations predicted more accurately than other baseline models under the collected datasets. For 10-meter datasets the RMSE index obtained as 0.25, 0.42 and 0.25. Also, for 50-meter datasets the RMSE index obtained as 0.26, 0.51 and 0.25. These results prove the proposed model accuracy.

Reference

- [1] J. Wang, Y. Wang, P. Jiang, The study and application of a novel hybrid forecasting model - a case study of wind speed forecasting in China. *Appl Energy* 2015; 143: 472-88.

- <http://dx.doi.org/10.1016/j.apenergy.2015.01.038>.
- [2] U. Focken, M. Lange, K. Mönnich, HP. Waldl, HG. Beyer, A. Luig, Short-term prediction of the aggregated power output of wind farms - A statistical analysis of the reduction of the prediction error by spatial smoothing effects. *J Wind Eng Ind Aerodyn* 2002; 90: 231-46. [http://dx.doi.org/10.1016/S0167-6105\(01\)00222-7](http://dx.doi.org/10.1016/S0167-6105(01)00222-7).
 - [3] Ahmed SD, Al-Ismail FSM, Shafiullah M et al (2020) Grid integration challenges of wind energy: a review. *IEEE Access* 8:10857–10878. <https://doi.org/10.1109/ACCESS.2020.2964896>.
 - [4] A. Meng, J. Ge J, H. Yi, S. Chen, Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. *Energy Convers Manage* 2016; 114: 75-88. <http://dx.doi.org/10.1016/j.enconman.2016.02.013>.
 - [5] Dokur, E., Erdogan, N., Salari, M. E., Karakuzu, C., & Murphy, J. (2022). Offshore wind speed real-time forecasting based on a hybrid method: Swarm decomposition and meta-extreme learning machine. *Energy*, 248, 123595.
 - [6] Wang, H., Zhang, Y. M., & Mao, J. X. (2022). Sparse Gaussian process regression for multi-step ahead forecasting of wind gusts combining numerical weather predictions and on-site measurements. *Journal of Wind Engineering and Industrial Aerodynamics*, 220, 104873.
 - [7] Yu, C., Li, Y., Chen, Q., Lai, X., & Zhao, L. (2022). Matrix-based wavelet transformation embedded in recurrent neural networks for wind speed prediction. *Applied Energy*, 324, 119692.
 - [8] Cassola, F., & Burlando, M. (2012). Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. *Applied energy*, 99, 154-166.
 - [9] Zhang, Y., Li, Y., & Zhang, G. (2020). Short-term wind power forecasting approach based on Seq2Seq model using NWP data. *Energy*, 213, 118371.
 - [10] Al-Yahyai, S., Charabi, Y., & Gastli, A. (2010). Review of the use of numerical weather prediction (NWP) models for wind energy assessment. *Renewable and Sustainable Energy Reviews*, 14(9), 3192-3198.
 - [11] Hoolohan, V., Tomlin, A. S., & Cockerill, T. (2018). Improved near surface wind speed predictions using Gaussian process regression combined with numerical weather predictions and observed meteorological data. *Renewable Energy*, 126, 1043-1054.
 - [12] Karasu, S., Altan, A., Bekiros, S., & Ahmad, W. (2020). A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. *Energy*, 212, 118750.
 - [13] Qolipour, M., Mostafaeipour, A., Saidi-Mehrabad, M., & Arabnia, H. R. (2019). Prediction of wind speed using a new Grey-extreme learning machine hybrid algorithm: A case study. *Energy & Environment*, 30(1), 44-62.
 - [14] Araya, I. A., Valle, C., & Allende, H. (2020). A Multi-Scale Model based on the Long Short-Term Memory for day ahead hourly wind speed forecasting. *Pattern Recognition Letters*, 136, 333-340.
 - [15] Li, J., & Armandpour, M. (2022, May). Deep spatio-temporal wind power forecasting. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4138-4142). IEEE.
 - [16] Karasu, S., & Altan, A. (2022). Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. *Energy*, 242, 122964.
 - [17] Ding, M., Zhou, H., Xie, H., Wu, M., Nakanishi, Y., & Yokoyama, R. (2019). A gated recurrent unit neural networks-based wind speed error correction model for short-term wind power forecasting. *Neurocomputing*, 365, 54-61.
 - [18] Araya, I. A., Valle, C., & Allende, H. (2020). A Multi-Scale Model based on the Long Short-Term Memory for day ahead hourly wind speed forecasting. *Pattern Recognition Letters*, 136, 333-340.
 - [19] Harbola, S., & Coors, V. (2019). One-dimensional convolutional neural network architectures for wind prediction. *Energy Conversion and Management*, 195, 70-75.
 - [20] Yu, R., Liu, Z., Li, X., Lu, W., Ma, D., Yu, M., ... & Li, B. (2019). Scene learning: Deep convolutional networks for wind power prediction by embedding turbines into grid space. *Applied energy*, 238, 249-257.
 - [21] Zhang, J., Liu, D., Li, Z., Han, X., Liu, H., Dong, C., ... & Xia, Y. (2021). Power prediction of a wind farm cluster based on spatiotemporal correlations. *Applied Energy*, 302, 117568.
 - [22] Ahmadi, M., & Khashei, M. (2021). Current status of hybrid structures in wind forecasting. *Engineering applications of artificial intelligence*, 99, 104133.
 - [23] Zhang, S., Chen, Y., Xiao, J., Zhang, W., & Feng, R. (2021). Hybrid wind speed forecasting model based on multivariate data secondary decomposition approach and deep learning algorithm with attention mechanism. *Renewable Energy*, 174, 688-704.
 - [24] Li, Y., Peng, T., Zhang, C., Sun, W., Hua, L., Ji, C., & Shahzad, N. M. (2022). Multi-step ahead wind speed forecasting approach coupling maximal overlap discrete wavelet transform, improved grey wolf optimization algorithm and long short-term memory. *Renewable Energy*, 196, 1115-1126.
 - [25] Jiang, Z., Che, J., He, M., & Yuan, F. (2023). A CGRU multi-step wind speed forecasting model based on multi-label specific XGBoost feature

- selection and secondary decomposition. *Renewable Energy*, 203, 802-827.
- [26] Tian, Z., & Wang, J. (2023). A wind speed prediction system based on new data preprocessing strategy and improved multi-objective optimizer. *Renewable Energy*, 118932.
- [27] Liu, X., Zhang, L., Wang, J., Zhou, Y., & Gan, W. (2023). A unified multi-step wind speed forecasting framework based on numerical weather prediction grids and wind farm monitoring data. *Renewable Energy*, 211, 948-963.
- [28] Abbasimehr, Hossein, Mostafa Shabani, and Mohsen Yousefi. "An optimized model using LSTM network for demand forecasting." *Computers & industrial engineering* 143 (2020): 106435.
- [29] Wu, Q., Zheng, H., Guo, X., & Liu, G. (2022). Promoting wind energy for sustainable development by precise wind speed prediction based on graph neural networks. *Renewable Energy*, 199, 977-992.
- [30] Qu, K., Si, G., Shan, Z., Kong, X., & Yang, X. (2022). Short-term forecasting for multiple wind farms based on CNN-LSTM model. *Energy Reports*, 8, 483-490.
- [31] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- [32] Shen, Z., Fan, X., Zhang, L., & Yu, H. (2022). Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network. *Ocean Engineering*, 254, 111352.
- [33] Nascimento, E. G. S., de Melo, T. A., & Moreira, D. M. (2023). A wavelet-based deep neural network with wavelet transforms for forecasting wind speed and wind energy. *Energy*, 278, 127678.
- [34] NASA, <https://power.larc.nasa.gov/data-access-viewer/>.
- [35] Moustafa, E. B., Hammad, A. H., & Elsheikh, A. H. (2022). A new optimized artificial neural network model to predict thermal efficiency and water yield of tubular solar still. *Case Studies in Thermal Engineering*, 30, 101750.
- [36] Zhao, Y., & Cen, Y. (2013). *Data mining applications with R*. Academic Press.
- [37] PyTorch, <https://zh.wikipedia.org/wiki/PyTorch>, 06.01.2024/07.09.2024.
- [38] Shu, Z. R., Chan, P. W., Li, Q. S., He, Y. C., & Yan, B. W. (2021). Investigation of chaotic features of surface wind speeds using recurrence analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 210, 104550.
- [39] Qu, F., Liu, J., Zhu, H., & Zhou, B. (2020). Wind turbine fault detection based on expanded linguistic terms and rules using non-singleton fuzzy logic. *Applied Energy*, 262, 114469.