Multi-step wind power forecasting over 24 hours using a CNN-LSTM hybrid model

Long Duong Hoang Vu[1], Quan Phung Mac[1], Kien Tran Ngoc[2], Van Nguyen Dinh[1], Ninh Nguyen Tuan[1*]

[1]*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology*
[2]*Queensland University of Technology*
[*]*Corresponding author E-mail: ninh.nguyentuan@hust.edu.vn*
DOI: https://doi.org/10.64032/mca.v29i4.345

Abstract

This study evaluates and compares three deep learning architectures for short-term wind power forecasting: Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and a hybrid CNN-LSTM model. While traditional statistical models often struggle with the nonlinear and temporal complexity of wind power data, neural-network-based approaches, especially hybrid architectures, have demonstrated superior performance. Using real-world data collected from a multi-turbine wind power plant in Soc Trang, Vietnam, we conducted a systematic evaluation across multiple forecasting horizons (1-hour, 12-hour, and 24-hour). The hybrid CNN-LSTM model consistently outperformed the standalone CNN and LSTM models, achieving up to 18% lower MAE and R^2 scores exceeding 0.87 for 1-hour forecasts. A robust preprocessing pipeline incorporating cyclical time encoding and seasonal feature annotation was implemented to enhance model performance. Although hybrid models have been explored in previous literature, this work contributes a localized and practical assessment of their effectiveness in a Southeast Asian context, with implications for operational wind energy forecasting.

***Keywords**: Convolutional Neural Network, Long Short-Term Memory, Multi-Step Forecasting, Wind Power Prediction*

Symbols

Symbols	Units	Description
N		Total forecasted sample
p_i	Watt	Actual Power production at time i
p_i'	Watt	Predicted Power production at time i
p_{mean}	Watt	The mean of p_i
θ	Degree	Angle variable
h	Hour	Hour factor
d	Day	Day factor
m	Month	Month factor
R^2		R square score

Abbreviations

ML	Machine learning
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory networks
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
ReLU	Rectified Linear Unit

1. Introduction

Recently, wind energy has been widely regarded as a promising renewable resource; however, its efficacy is hindered by the variability in weather conditions [1]. Consequently, accurate wind power generation forecasting is essential for optimizing the power distribution and ensuring the reliability of the electricity supply [2]. Various studies have utilized either statistical or machine learning (ML) methods for wind power prediction [3]. Traditional statistical models, such as AutoRegressive Integrated Moving Average and physical-based models, exhibit limitations in capturing the complex nonlinear dependencies of wind power. In comparison, Convolutional Neural Networks (CNN) demonstrate superior predictive performance and solve complex nonlinear relationship problems between datasets [4]. Despite these advancements, the current study had several limitations. Most ML-based wind power forecasting models are designed for specific locations and are evaluated using datasets that are closely related to their training data.

This study compares three machine learning architectures, LSTM, CNN, and hybrid CNN-LSTM, for wind power forecasting, with the aim of identifying models that provide better insight into temporal dynamics. The pipeline consists of data preprocessing, feature selection, and model training, with consideration of missing values and normalization. Experiments conducted on wind power data from South Vietnam demonstrated that the CNN-LSTM model consistently outperformed the standalone CNN and LSTM models. By evaluating these architectures under the same conditions, this study highlights the advantages of hybrid deep learning for improving short-term wind power prediction.

2. Dataset

The dataset comprises measurements from seven wind turbines at a wind power plant in Soc Trang, collected at 10-minute intervals from January 1, 2022, to May 1, 2024. Five variables were recorded for each turbine: the wind speed (m/s), wind direction (°), turbine direction (°), ambient temperature (°C), and actual power output (kW). The wind speed ranges from approximately 0 to 24.3 m/s, the wind and turbine directions span from 0° to 359°, the ambient temperature varies from 21°C to 33°C, and the power output of these turbines ranges from -50 kW to 4200 kW. Notably, a

Received: 30 May 2025; Revised: 17 October 2025; Accepted: 30 October 2025.

small portion of the measured power output may be negative because of the power consumed by the turbine itself during low-wind conditions or in startups. Owing to the distinct seasonal climate in Soc Trang, two Boolean attributes, dry and wet seasons, were added based on the timestamp of the year.

Fig. 1 presents a correlation heatmap of key environmental variables and power production, revealing important relationships within the dataset. A strong positive correlation (0.92) was observed between the wind speed and power, reinforcing the critical role of wind speed in energy generation. Similarly, the wind and turbine directions show a high correlation (0.95), indicating that the turbine alignment closely follows the wind patterns. Temperature appears to have a moderate negative correlation with both wind speed (-0.24) and power (-0.21), suggesting that cooler conditions may favor wind generation at the site. Seasonal variables exhibited contrasting relationships: the dry season correlated negatively with wind speed (-0.42) and power (-0.41), while the wet season showed positive correlations (0.42 and 0.41, respectively), implying that the wet season may be associated with more favorable wind conditions.

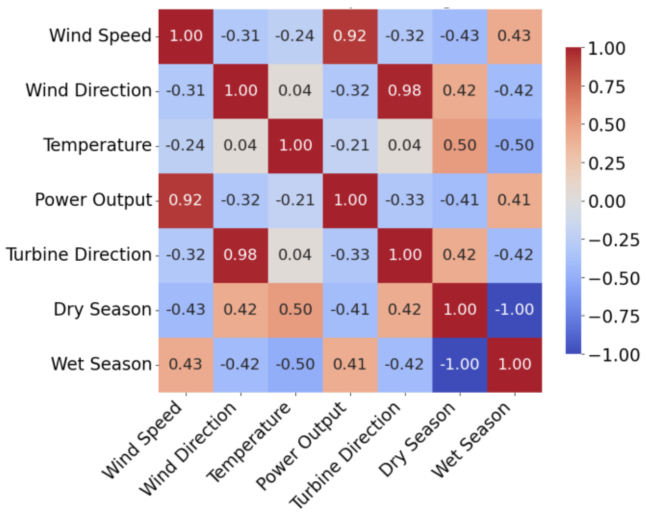

Figure 1: Correlation Heatmap between features of the dataset

3. Pre-processing

Data preprocessing plays a vital role in machine learning pipelines by transforming raw data into a clean, consistent, and structured format suitable for model training. This step not only helps ensure data quality but also enhances model performance by removing noise, handling missing values, standardizing input formats, and preserving meaningful patterns. Effective preprocessing enables machine learning algorithms to learn more efficiently from data, ultimately leading to more accurate and reliable predictions.

3.1 Data cleaning

Missing values were handled using appropriate imputation methods to ensure data quality and improve model performance. Most columns were imputed using linear interpolation, as the data were recorded at 10-minute intervals and weather and power measurements typically changed gradually over short periods. Therefore, linear interpolation is suitable for estimating the missing values in such continuous variables. In contrast, the wind and turbine directions, which are circular in nature (ranging from 0° to 360°), were imputed using a forward fill. This method preserves the continuity of directional data and avoids issues in which 0° and 360° represent similar directions, which could be misrepresented by interpolation.

3.2 Resampling

The original dataset, recorded at 10-minute intervals, was resampled to an hourly frequency using the mean aggregation method. This resampling process aims to reduce the high-frequency noise present in the raw measurements, which can negatively affect the model performance by introducing unnecessary variance. By averaging the values over each hour, the data became smoother and emphasized the overall trends rather than the short-term fluctuations. This transformation enhances the ability of the model to learn meaningful patterns, particularly for forecasting tasks in which temporal consistency and trend detection are more valuable than capturing rapid changes.

3.3 Data encoding

The time data was first converted to the DateTime format to enable time-series analysis, including autocorrelation and partial autocorrelation plots, for detecting temporal dependencies. Feature engineering was then applied, with particular attention paid to cyclical variables, such as wind direction and time-based components. To preserve the continuity of the angular variables, such as the wind direction (denoted as), sine and cosine transformations were used. Each angle $\theta \in [0, 360)$ was encoded as:

$$\theta_{\sin} = \sin\left(\frac{2\pi\,\theta}{360}\right), \theta_{cos} = \cos\left(\frac{2\pi\theta}{360}\right) \quad (1)$$

This encoding ensures that values such as 0° and 359° are numerically close, thereby maintaining the true circular nature of the data.

Similarly, temporal variables, such as hour of day (h), day of month (d), and month of year (m), were encoded using sine and cosine transformations to reflect their periodic or seasonal structures:

$$h_{sin} = \sin\left(\frac{2\pi\,h}{24}\right), \quad h_{cos} = \cos\left(\frac{2\pi\,h}{24}\right) \quad (2)$$

$$d_{sin} = \sin\left(\frac{2\pi d}{31}\right), \quad m_{cos} = \cos\left(\frac{2\pi d}{31}\right) \quad (3)$$

$$m_{sin} = \sin\left(\frac{2\pi m}{12}\right), \quad m_{cos} = \cos\left(\frac{2\pi m}{12}\right) \quad (4)$$

These transformations help the model capture regular time-based patterns more effectively. Broader seasonal effects were encoded using one-hot vectors to represent the four seasons.

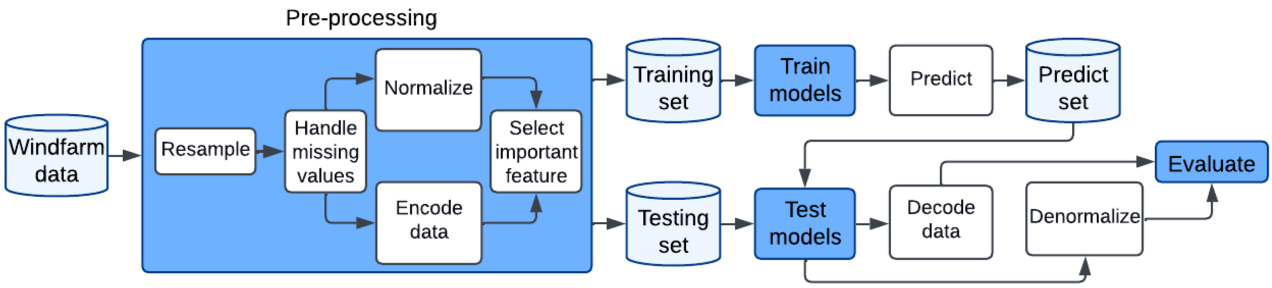

Figure 2: Data Pipeline

3.4 Normalization

Before training, the numerical input features were standardized using StandardScaler to ensure zero mean and unit variance. This step improves the model convergence by aligning feature distributions and preventing bias toward attributes at larger scales. Features that had undergone prior transformation, such as sine-cosine encoded cyclical variables (e.g., wind direction and time-based features) and one-hot encoded seasonal indicators, were excluded from standardization to preserve their structural properties. The target variable (wind power) was also standardized separately to maintain consistency during the regression.

3.5 Data pipeline

After encoding, the dataset was structured into input features X and target outputs Y, where the model learns to predict Y based on X. At each time step t, the input X_t includes all available features at time t, excluding data from the final day to avoid look-ahead bias. The target Y_t consists of wind power values for the next 24 hours, defined as:

$$Y_t = [p_{t+1}, p_{t+2}, \dots, p_{t+24}] \quad (5)$$

where P_t denotes the wind power at hour t. This formulation enables multistep forecasting, allowing the model to learn temporal dependencies between current conditions and short-term future outputs.

Fig. 2 shows the wind power forecasting pipeline. Initially, the raw wind farm data underwent preprocessing, including missing value imputation, wind direction encoding, data normalization, and feature selection. Subsequently, the processed data were divided into training and testing sets. The training set was utilized to develop machine learning models, while the testing set was reserved for evaluation. Following this, the predictions were decoded and denormalized before being assessed for accuracy and performance. This structured approach ensures robust and interpretable forecasting.

The dataset was partitioned chronologically to ensure a realistic and leakage-free evaluation setting. Specifically, data collected before Febuary 2, 2024 (constituting approximately 90% of the entire dataset) were used exclusively for training, while data from Febuary 2, 2024 onward (the remaining 10%) were reserved for testing. No data shuffling was performed during the split, in contrast to typical machine learning pipelines. This decision was made to prevent information from future time steps contaminating the training process, which could otherwise artificially inflate model performance and undermine its ability to generalize to unseen future data- a well-documented issue in time series forecasting tasks. Various machine learning models have been implemented and assessed to determine the most effective approach for wind power prediction. These models range from conventional statistical and ensemble methods to advanced deep learning architectures.

4. Models

4.1 Convolutional neural network

In time-series data, Convolutional Neural Networks (CNNs) work by applying convolutional layers to capture local patterns or trends over time. The convolutional filters slide across the data sequence, identifying important features, such as trends, seasonality, and other temporal dependencies. Pooling layers were then used to downsample the data, retaining only the most important features. This allows CNNs to efficiently extract temporal patterns and make predictions based on sequences of past data points. Despite being commonly used in image processing, CNNs are also effective in time-series forecasting by detecting complex temporal dependencies that may be difficult to model with traditional methods [5].

The CNN architecture implemented in this study was configured to balance forecasting accuracy with computational efficiency, making it suitable for time-sensitive applications. The model consists of a one-dimensional convolutional layer (Conv1D) with 32 filters and a kernel size of 3 to capture short-term temporal patterns in the input sequence. This is followed by a flattening layer, a dense layer with 16 ReLU-activated units, a dropout layer with a rate of 0.1, and a final output layer that predicts 24 future hourly wind power values. This lightweight structure is motivated by findings in [6], where compact CNN-based models demonstrated fast inference and training while achieving competitive short-term forecasting performance.

However, the limited receptive field of Conv1D layers hinders the model's ability to capture long-range dependencies. As shown in [7], standalone CNNs underperform in multistep wind power forecasting tasks where temporal memory is critical. To mitigate this, recent studies have adopted hybrid models that integrate CNNs for feature extraction with recurrent layers like LSTM or GRU to capture temporal dynamics [8].

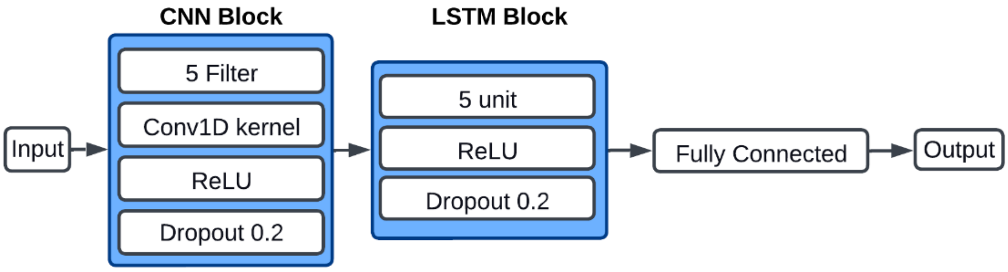

Figure 3: CNN-LSTM architecture

In our experiments, the standalone CNN served as a strong baseline for short-horizon predictions, but its performance declined when forecasting extended time ranges, confirming the necessity of sequence-aware modeling in wind power prediction.

4.2 Long short term memory

Long Short-Term Memory (LSTM) networks are designed to handle sequential data, such as time-series. They operate by maintaining a memory cell that stores information over long periods. This memory cell is updated through three gates: the input gate (which decides what new information to add), forget gate (which decides what information to discard), and output gate (which determines what to output) [9]. This structure allows LSTMs to capture long-term dependencies in time-series data, making them particularly effective for forecasting, anomaly detection, and modeling trends over time. Unlike traditional Recurrent Neural Networks, LSTMs can remember information over longer sequences, mitigating the problem of vanishing gradients that hinders other models in time-series tasks [10].

The optimized LSTM model developed in this study adopts an encoder-decoder architecture tailored for short-term wind power forecasting. The encoder comprises an LSTM layer with 24 units, activated by ReLU for the input and sigmoid for the recurrent state, and includes dropout and L2 regularization to enhance generalization. A RepeatVector layer bridges the encoder and decoder, allowing the decoder to generate multistep forecasts based on the encoded context. The decoder is another LSTM layer with 24 units and similar configurations, returning sequences for each future time step. The output layer is a time-distributed dense layer that produces one value per time step, allowing for forecasting over a defined horizon. This structure is further supported by gradient clipping, learning rate scheduling, and early stopping mechanisms to ensure training stability and performance. Hyperparameters, including a 10 steps lookback window, batch size of 256, and dropout rate of 0.2, were fine-tuned for faster training without compromising accuracy. The model was trained using the Adam optimizer with a learning rate of 0.001 and mean squared error as the loss function.

4.3 Hybrid CNN-LSTM model

The CNN-LSTM model combines the strengths of two powerful neural network architectures: Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). This model is widely used in time-series forecasting and spatiotemporal data tasks, such as weather prediction, sound recognition, or signal analysis. CNN is responsible for extracting spatial features from the input data, whereas LSTM captures long-term temporal dependencies, making this model robust for understanding and predicting time-dependent data.

Fig. 3 illustrates the hybrid CNN-LSTM model. The model begins with a CNN layer, which is a 1D convolutional layer that extracts local features from the input time-series data. It uses five CNN filters of size three to learn temporal patterns and capture local dependencies, applying ReLU activation and padding to maintain the input size. The dropout rate was set to 0.2. Following the CNN block, an LSTM layer with five units captures long-term dependencies in the data, processing the output of the convolutional layer and learning sequential patterns over time. A dropout rate of 0.2 is applied after the LSTM layer to prevent overfitting. The model processes input sequences of 10 time steps and uses a batch size of 256. Finally, the model ends with a dense layer consisting of five units, performing the final mapping to the output space. The model was optimized using the Adam optimizer with a learning rate of 0.001 and trained using the Mean Squared Error loss function. Despite the sophisticated structure of the model, the total number of units was relatively low, allowing for faster training compared to previous models.

5. Result and Evaluation

5.1 Evaluate methodology

Each model, including CNN, LSTM, and CNN-LSTM, was tested using the optimal hyperparameters and evaluated using three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2. These metrics ensure a comprehensive assessment of each model's accuracy, robustness, and generalization ability in predicting wind turbine power output [11].

5.1.1 *Mean Absolute Error*

MAE is the average of the absolute errors between the predicted and actual values in the test set, providing a straightforward measure of prediction accuracy.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|p_i - p_i'\right| \quad (6)$$

where p_i' represents the predicted power output, p_i is the actual power output at time i, and N denotes the total number of forecast samples.

5.1.2 *Root Mean Squared Error*

The RMSE is the square root of the mean squared error between the predicted and actual values. It penalizes larger errors more heavily, making it useful for assessing model accuracy and detecting significant deviations.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - p_i')^2} \quad (7)$$

where p_i' is the predicted power output, p_i is the actual power output at time i , and N is the total number of forecast samples.

5.1.3 *R Square*

The correlation coefficient, often denoted as 'R, ranges from -1 to 1. A value of +1 signifies that an increase in input x leads to a proportional increase in output y, whereas -1 indicates an inverse relationship, where an increase in x causes y to decrease. A value of 0 indicates that no correlation exists between the two variables. Similar to correlation, R-squared (R^2) measures the strength of the relationship between variables but is more intuitive to interpret. This represents the proportion of variance in the dependent variable that can be explained by the independent variable.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(p_i - p_i')^2}{\sum_{i=1}^{N}(p_i - p_{mean})^2} \quad (8)$$

where p_i' is the predicted power output, p_i is the actual power output at time i, p_{mean} is the mean of p_i and N is the total number of forecast samples. The R^2 value ranges from 0 to 1 (or 0% to 100%), with higher values indicating a better-fitting predictive model.

5.2 Result

Fig. 4 depicts the training and validation loss curves of our proposed CNN-LSTM hybrid model over 80 epochs of training for the best performance. The graph demonstrates exceptional convergence characteristics, with both training and validation losses rapidly decreasing from initial values of approximately 1.2 to stabilize around 0.4 by epoch 20. Notably, the minimal gap between the training and validation curves throughout the training process indicates excellent generalization capabilities without signs of overfitting. The close allignment of both curves after epoch 30 suggests that the model successfully reached an optimal state, capturing the underlying patterns in the data while maintaining robustness on the unseen samples. This balanced performance validates our architectural choices and hyperparameter settings, thereby confirming the effectiveness of combining convolutional layers for spatial feature extraction with LSTM units for temporal dependencies in the target domain.

Fig. 5 demonstrates the CNN-LSTM model's prediction capabilities at different time horizons, with excellent 1-hour forecasting accuracy that closely tracks actual values including extreme peaks and rapid transitions. Then, the performance degraded for 12-hour predictions and further degraded for 24-hour predictions, particularly struggling with extreme value magnitudes and transition timing, highlighting the inherent trade-off between prediction horizon length and forecast accuracy.

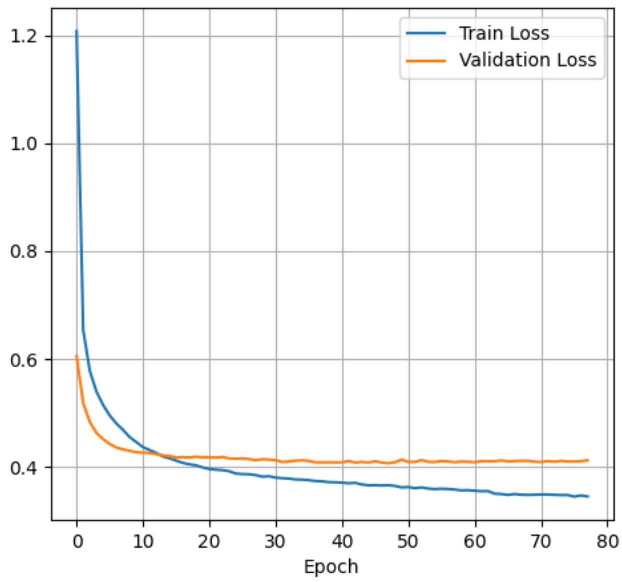

Figure 4: Training loss and validation loss for best-performing turbine

Tables 1a, 1b, and 1c present the Mean Absolute Error (MAE) values for the three models across the seven turbines. The CNN-LSTM Hybrid model consistently achieves the lowest MAE for 1-hour predictions, with values ranging from 361.56 to 399.89, significantly outperforming both individual CNN and LSTM models. For 12-hour and 24-hour horizons, while performance degrades across all models, the hybrid approach maintains its advantage, with the lowest MAE of 746.21 at 12 hours and 882.08 at 24 hours.

Tables 2a, 2b, and 2c show the Root Mean Squared Error (RMSE) results, which penalize larger errors more heavily. The CNN-LSTM Hybrid model demonstrated superior performance for 1-hour predictions across all turbines, achieving the lowest RMSE values between 552.78 and 599.20. As the prediction horizon extends to 12 and 24 h, all models show increased error, with the performance gap narrowing. However, the hybrid model maintained a slight edge, particularly for the 24-hour horizon where it achieved the lowest RMSE in five of the seven turbines.

Tables 3a, 3b, and 3c illustrate the R^2 scores, representing how well each model explains the variance in the target variable. For 1-hour predictions, the CNN-LSTM Hybrid model consistently delivered the highest scores, ranging from 0.8524 to 0.8824, outperforming both individual architectures. The predictive capability decreases substantially across all models for longer horizons, with 12-hour R^2 values dropping to the 0.55-0.60 range and 24-hour values further declining to 0.40-0.51.

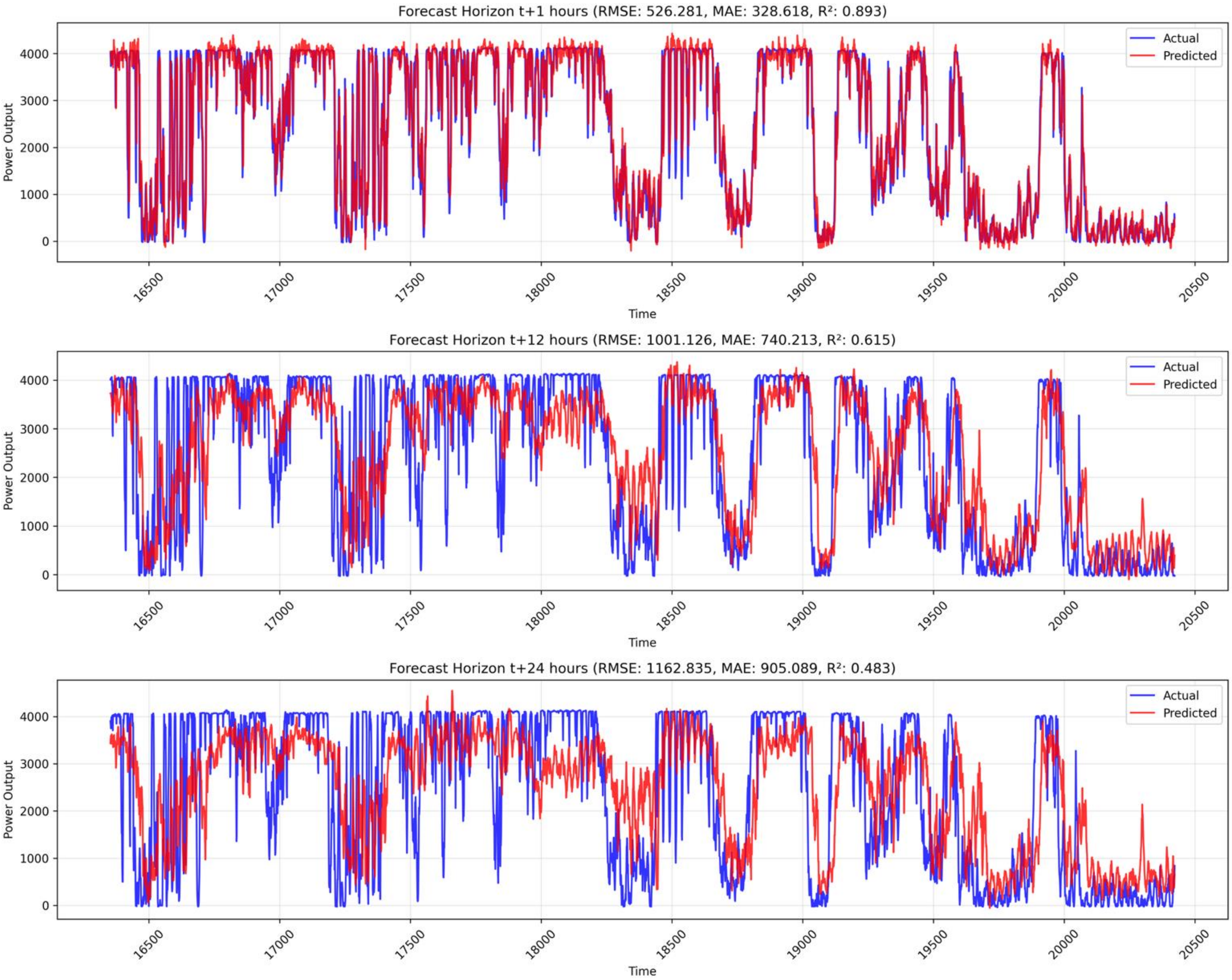

Figure 5: Predict and actual value of different time horizon

This performance degradation illustrates the inherent challenge in long-term time-series forecasting, although the hybrid model still maintains marginally better results at these extended horizons.

The CNN-LSTM Hybrid model outperformed both individual CNN and LSTM models across all evaluation metrics. For 1-hour predictions, the hybrid approach shows clear superiority in every turbine. Though all models face declining accuracy at 12-hour and 24-hour horizons, the hybrid model maintains its advantage, particularly for Turbine 7. The CNN-LSTM Hybrid also benefits from faster training speeds, making it more practical for real-world deployment. These results confirm that combining convolutional and recurrent neural networks creates a more effective architecture for wind power forecasting by capturing both spatial patterns and temporal relationships in the data. This approach offers significant potential for improving renewable energy prediction systems.

Table 1a: Mean Absolute Error 1 hour

		MAE	
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	417.7823	541.7214	**391.6927**
2	447.6763	583.4842	**361.5601**
3	473.1142	592.7705	**399.8889**
4	470.9223	684.0718	**377.3585**
5	468.8355	579.3439	**377.5320**
6	451.6157	463.9853	**371.7476**
7	497.8146	523.4405	**367.3138**

Table 1b: Mean Absolute Error 12 hour

		MAE	
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	833.2363	824.2859	**786.7181**
2	870.9351	859.2371	**786.1800**
3	858.4864	828.3308	**786.3106**
4	844.6105	918.7668	**802.8056**
5	872.8110	815.0673	**776.8602**
6	864.2609	799.2419	**795.4164**
7	835.5159	802.5258	**746.2086**

Table 1c: Mean Absolute Error 24 hour

	MAE		
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	931.4372	948.6808	**910.1708**
2	936.3844	973.7699	**942.7874**
3	974.1188	1010.0448	**916.2318**
4	**982.3638**	1003.8642	995.6389
5	963.1328	978.4497	**906.9044**
6	982.4880	**902.3207**	902.7538
7	925.1179	916.4536	**882.0771**

Table 2a: Root Mean Squared Error 1 hour

	RMSE		
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	595.5508	732.7060	**582.4673**
2	614.4274	734.7492	**562.2050**
3	639.8942	768.8626	**599.2042**
4	648.5389	883.4594	**587.6154**
5	612.7895	761.9235	**562.5412**
6	610.9427	654.9454	**556.9359**
7	633.5373	724.3458	**552.7840**

Table 2b: Root Mean Squared Error 12 hour

	RMSE		
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	1070.3712	1051.2151	**1019.7655**
2	1100.6915	1052.4253	**1056.0811**
3	1096.2578	1040.1313	**1013.6262**
4	1080.9441	1140.8448	**1048.4624**
5	1095.1883	1038.5645	**1020.3248**
6	1080.2566	1026.4905	**1012.1660**
7	1064.6763	**1018.7577**	1018.9082

Table 2c: Root Mean Squared Error 24 hour

	RMSE		
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	1170.5751	1171.3338	**1144.9914**
2	**1157.7894**	1167.6286	1215.0495
3	1206.9098	1194.0555	**1155.9110**
4	1202.1728	1216.8884	**1221.4901**
5	1175.8410	1178.2663	**1149.2204**
6	1191.2417	1164.4072	**1140.9521**
7	1152.4052	1139.4322	**1134.0292**

Table 3a: R^2 scores 1 hour

	R2		
Turbine	CNN	LSTM	CNN-LSTM Hybrid
1	0.8585	0.7859	**0.8647**
2	0.8492	0.7845	**0.8738**
3	0.8317	0.7570	**0.8524**
4	0.8318	0.6879	**0.8619**
5	0.8540	0.7744	**0.8770**
6	0.8522	0.8302	**0.8772**
7	0.8455	0.7981	**0.8824**

Table 3b: R^2 scores 12 hour

	R2		
Tu	CNN	LSTM	CNN-LSTM
1	0.5447	0.5609	**0.5868**
2	0.5177	**0.5591**	0.5561
3	0.5073	0.5565	**0.5788**
4	0.5340	0.4810	**0.5617**
5	0.5353	0.5822	**0.5968**
6	0.5396	0.5843	**0.5958**
7	0.5650	**0.6018**	0.6017

Table 3c R^2 scores 24 hour

	R2		
Tu	CNN	LSTM	CNN-LSTM
1	0.4561	0.4554	**0.4799**
2	**0.4671**	0.4580	0.4134
3	0.4034	0.4163	**0.4533**
4	**0.4242**	0.4102	0.4060
5	0.4652	0.4630	**0.4895**
6	0.4409	0.4658	**0.4874**
7	0.4915	0.5029	**0.5079**

6. Conclusion

This study presents a comprehensive and scalable pipeline for short-term wind power forecasting, combining robust preprocessing techniques with comparative model evaluation. Among the models tested, the hybrid CNN-LSTM architecture demonstrated superior performance by effectively capturing local temporal dynamics through convolutional layers and retaining long-range sequential dependencies via LSTM units. This dual capability enables the model to handle fluctuations in wind power with greater accuracy and stability, particularly across multiple turbines with varying patterns.

Beyond accuracy, the proposed approach holds practical implications for real-world energy operations. Improved forecasting precision can support better short-term dispatch planning, reduce the reliance on spinning reserves, and enhance grid stability-especially in systems with high renewable penetration. Additionally, the lightweight CNN component contributes to faster training and inference, making the model suitable for time-sensitive applications in control centers or embedded edge devices.

Future research could extend this work by integrating high-resolution meteorological data (e.g., wind shear, atmospheric pressure), employing transfer learning to generalize across geographically distinct wind farms, and deploying the model in online or streaming settings to evaluate its robustness under real-time conditions. Such advancements could further enhance the operational value of AI-driven forecasting systems in modern energy markets.

References

[1] E. G. A. Antonini, E. Virgüez, S. Ashfaq, L. Duan, T. H. Ruggles, and K. Caldeira, "Identification of reliable locations for wind power generation through a global analysis of wind droughts," *Commun Earth Environ*, vol. 5, no. 1, p. 103, Mar. 2024, doi: 10.1038/s43247-024-01260-7.

[2] A. T. Peiris, J. Jayasinghe, and U. Rathnayake, "Forecasting Wind Power Generation Using Artificial Neural Network: 'Pawan Danawi'—A Case Study from Sri Lanka," *Journal of Electrical and Computer Engineering*, vol. 2021, pp. 1–10, Mar. 2021, doi: 10.1155/2021/5577547.

[3] J. L. Uc Castillo, A. E. Marín Celestino, D. A. Martínez Cruz, J. Tuxpan Vargas, J. A. Ramos Leal, and J. Morán Ramírez, "A systematic review of Machine Learning and Deep Learning approaches in Mexico: challenges and opportunities," *Front. Artif. Intell.*, vol. 7, p. 1479855, Jan. 2025, doi: 10.3389/frai.2024.1479855.

[4] Y. Dou, S. Tan, and D. Xie, "Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: a mini review," *Front. Energy Res.*, vol. 11, p. 1218603, Jul. 2023, doi: 10.3389/fenrg.2023.1218603.

[5] M. Solas, N. Cepeda, and J. L. Viegas, "Convolutional Neural Network for Short-term Wind Power Forecasting," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, Bucharest, Romania: IEEE, Sep. 2019, pp. 1–5. doi: 10.1109/ISGTEurope.2019.8905432.

[6] X. Huang, Y. Zhang, J. Liu, X. Zhang, and S. Liu, "A Short-Term Wind Power Forecasting Model Based on 3D Convolutional Neural Network–Gated Recurrent Unit," *Sustainability*, vol. 15, no. 19, p. 14171, Sep. 2023, doi: 10.3390/su151914171.

[7] Y. Meng *et al.*, "Research on Ultra-Short-Term Prediction Model of Wind Power Based on Attention Mechanism and CNN-BiGRU Combined," *Front. Energy Res.*, vol. 10, p. 920835, May 2022, doi: 10.3389/fenrg.2022.920835.

[8] J. Zhang, Z. Zhao, J. Yan, and P. Cheng, “Ultra-Short-Term Wind Power Forecasting Based on CGAN-CNN-LSTM Model Supported by Lidar,” *Sensors*, vol. 23, no. 9, p. 4369, Apr. 2023, doi: 10.3390/s23094369.

[9] Z. Xiao, F. Tang, and M. Wang, “Wind Power Short-Term Forecasting Method Based on LSTM and Multiple Error Correction,” *Sustainability*, vol. 15, no. 4, p. 3798, Feb. 2023, doi: 10.3390/su15043798.

[10] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network,” 2018, doi: 10.48550/ARXIV.1808.03314.

[11] L. Cheng, R. Guo, R. Moraffah, P. Sheth, K. S. Candan, and H. Liu, “Evaluation Methods and Measures for Causal Learning Algorithms,” Feb. 07, 2022, *arXiv*: arXiv:2202.02896. doi: 10.48550/arXiv.2202.02896.

[12] H. L. Leka, Z. Fengli, A. T. Kenea, A. T. Tegene, P. Atandoh, and N. W. Hundera, “A Hybrid CNN-LSTM Model for Virtual Machine Workload Forecasting in Cloud Data Center,” in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China: IEEE, Dec. 2021, pp. 474–478. doi: 10.1109/ICCWAMTIP53232.2021.9674067.

[13] Ö. A. Karaman, “Prediction of Wind Power with Machine Learning Models,” *Applied Sciences*, vol. 13, no. 20, p. 11455, Oct. 2023, doi: 10.3390/app132011455.