electronics

MDPI

Article

Dynamic Gust Detection and Conditional Sequence Modeling for Ultra-Short-Term Wind Speed Prediction

Liwan Zhou, Di Zhang *, Le Zhang and Jizhong Zhu *

School of Electric Power Engineering, South China University of Technology, Guangzhou 510641, China; hualaobisi@gmail.com (L.Z.); wjtclwwf@163.com (L.Z.)
* Correspondence: dizhang@scut.edu.cn (D.Z.); zhujz@scut.edu.cn (J.Z.)

Abstract: As the foundation for optimizing wind turbine operations and ensuring energy stability, wind speed forecasting directly impacts the safe operation of the power grid, the rationality of grid planning, and the balance of supply and demand. Furthermore, gust events, characterized by sudden and rapid wind speed fluctuations, pose significant challenges for ultra-short-term wind speed forecasting, making the data more complex and thus harder to predict accurately. To address this issue, this paper proposes a novel hybrid model that combines dynamic gust detection with Conditional Long Short-Term Memory (Conditional LSTM) and incorporates dynamic window adjustment and wind speed difference threshold screening methods. The model dynamically adjusts the window size to accurately detect gust events and uses a conditional LSTM model to adjust predictions based on gust and non-gust conditions. Experimental results show that the proposed model exhibits higher prediction accuracy across various wind speed scenarios, particularly during gust events. Through detailed experiments using data from a single actual wind farm, the effectiveness and practicality of the proposed hybrid model are demonstrated. The experimental results indicate that the proposed model outperforms contrast models, especially in handling gust events, significantly enhancing the robustness of ultra-short-term wind speed predictions.

Keywords: wind speed forecasting; gust detection; ultra-short-term prediction; conditional long short-term memory (Conditional LSTM); hybrid model

check for updates

Citation: Zhou, L.; Zhang, D.; Zhang, L.; Zhu, J. Dynamic Gust Detection and Conditional Sequence Modeling for Ultra-Short-Term Wind Speed Prediction. *Electronics* **2024**, *13*, 4513. https://doi.org/10.3390/electronics13224513

Academic Editor: François Auger

Received: 8 October 2024
Revised: 8 November 2024
Accepted: 12 November 2024
Published: 18 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Wind energy plays a crucial role in the global renewable energy mix, significantly contributing to reducing carbon emissions and addressing climate change. As global wind power capacity continues to grow rapidly, the sudden variations in wind speed pose significant challenges to the sustainable utilization of wind energy and the stable operation of power grids [1,2]. Particularly, rapid fluctuations in wind speed, often manifesting as gusts, not only affect the stability of wind power generation but also impose higher demands on the operation and maintenance of wind farms and grid dispatching [3]. Therefore, accurate wind speed forecasting, especially ultra-short-term wind speed forecasting, is essential for the smooth operation of wind farms.

Research in wind power forecasting can be categorized into ultra-short-term (up to 1 h), short-term (1 h to 1 day), mid-term (1 day to 1 week), and long-term forecasting (more than 1 week), with ultra-short-term predictions being particularly vital for daily turbine operations. This forecasting accuracy is essential for the dispatching department to formulate generation plans and maintain the real-time balance of the power system. Current forecasting methods include physical approaches, reliant on meteorological data, statistical methods, focused on historical data analysis, artificial intelligence techniques, which leverage advanced computational capabilities, and hybrid methods, which combine elements from these different approaches to enhance prediction accuracy.

Physical methods typically rely on meteorological data and Numerical Weather Prediction (NWP), achieving good results in medium-term and long-term wind speed forecasting.

However, due to their high dependence on real-time meteorological data and complex computational requirements, physical methods struggle to meet the real-time needs of ultra-short-term forecasting, particularly in gust environments [2]. The rapid and unpredictable nature of gusts often leads to significant discrepancies between forecasted and actual wind speeds, as these methods may not be agile enough to adapt to sudden changes.

Statistical models, including Autoregressive (AR), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA) models, are commonly used in ultra-short-term forecasting due to their ability to capture linear trends in wind data. However, they exhibit poor performance when handling non-linear and highly dynamic events such as gusts [4,5].

In recent years, artificial intelligence (AI) methods have demonstrated superior capabilities in capturing non-linear patterns in wind data. Models such as Artificial Neural Networks (ANNs) [6], Long Short-Term Memory (LSTM) networks [7,8], and Convolutional Neural Networks (CNNs) [9–11] have been widely adopted for ultra-short-term wind speed prediction. LSTM has shown a strong ability to model time-dependent sequences by maintaining long-term dependencies. However, many AI-based methods still fall short in accurately predicting wind gust events due to their inability to adapt to the sudden and short-lived nature of gusts. Additionally, the training processes of these AI models may not adequately emphasize the importance of sudden fluctuations in wind speeds. As a result, they may generalize poorly in situations that demand immediate responsiveness to dynamic changes, such as gusts. This limitation highlights the need for more specialized approaches that can directly address the unique characteristics of gust events in wind speed forecasting.

Gust forecasting initially emerged within the domain of weather forecasting, where meteorological approaches primarily relied on statistical data and physical methods. However, these traditional methods are often ill-suited for ultra-short-term wind speed forecasting, particularly due to their reliance on a large number of parameters and the complexity involved in modeling atmospheric dynamics [12–16]. Moreover, it is particularly noteworthy that statistical methods often outperform physical modeling approaches in the context of gust prediction. This disparity arises from the inherent challenges in ensuring the accuracy of parameters within physical models, which can compromise their effectiveness [16]. Consequently, there has been a discernible shift toward utilizing machine learning techniques for gust forecasting [17], with a growing emphasis on hybrid methods that integrate the strengths of both statistical and computational approaches. Additionally, the incorporation of radar data has become increasingly important for tracking and predicting gust events [18]. Radar systems provide real-time information on wind patterns and atmospheric conditions, enabling more accurate and timely forecasts. However, for the wind energy sector, the use of radar also presents several challenges. The spatial resolution of radar can also limit its effectiveness in capturing small-scale gust phenomena, potentially overlooking critical local wind speed variations.

This transition underscores the importance of developing effective hybrid models that can simultaneously address the complexities of gust events while leveraging the predictive power of various algorithms. Hybrid methods often integrate statistical techniques with machine learning algorithms, allowing for better modeling of complex, non-linear patterns in wind data. For example, combining autoregressive models with neural networks can capture both linear trends and intricate dynamics, leading to improved robustness against data variability [6,19,20]. Recent studies have demonstrated the effectiveness of hybrid models not only in load forecasting [21,22] but also in wind prediction, utilizing approaches that adapt to fluctuations and uncertainties inherent in wind energy generation and demand patterns. These models not only improve forecast accuracy but also provide probabilistic estimates that are crucial for risk management and operational decision-making in wind energy contexts [23–25].

Additionally, some hybrid methods employ data decomposition techniques to break wind speed time series into more manageable components, enabling more focused pre-

dictions for different wind conditions. Commonly used decomposition methods include Empirical Mode Decomposition (EMD) [26–31], which decomposes time series into Intrinsic Mode Functions (IMFs) to effectively capture local oscillations and trends; Wavelet Transform (WT) [32–34], which analyzes signals at various scales for multi-resolution analysis; Variational Mode Decomposition (VMD) [1,35–38], which separates the signal into modes with specific bandwidths; and Ensemble Empirical Mode Decomposition (EEMD) [39–42], an extension of EMD that adds noise to enhance robustness and reliability. These decomposition methods significantly improve the capability of hybrid forecasting models to accurately predict wind power generation by isolating different frequency components of the wind speed signals.

The latest advancements in wind speed forecasting involve combining these decomposition methods with deep learning models. The decomposed components, which represent different frequency bands or modes of the wind speed signal, are fed into deep learning models for prediction. These models, such as LSTMs [39,43–46], GRU [46–48], or other neural networks [49,50], learn the underlying patterns of each component and produce individual forecasts for each. Finally, the results from the different components are combined to provide a more accurate and robust prediction of wind speed. This approach capitalizes on the strengths of both decomposition techniques and deep learning models to enhance predictive performance.

In the application of hybrid methods, although data decomposition techniques can effectively extract different frequency components from wind speed signals, they may struggle to accurately identify the rapid dynamic features associated with sudden events, such as gusts. This inability to recognize such changes can affect the model's predictive accuracy during gust events, leading to delayed responses to variations in wind speed. Furthermore, while decomposition techniques can isolate different frequency components, crucial temporal information may be overlooked in complex nonlinear dynamics, further diminishing overall predictive performance. Additionally, some decomposition methods, such as EEMD and VMD, involve complex computations and parameter selection, which can introduce delays in real-time forecasting and affect the model's practicality.

However, despite their advantages, these hybrid methods often lack a clear physical mechanism to explain gust phenomena. While data decomposition techniques can isolate various frequency components from wind speed signals, they frequently struggle to capture the rapid dynamics associated with gusts. This limitation can hinder the model's predictive accuracy during sudden wind speed variations, reducing their effectiveness in real-time applications. Therefore, the absence of a physically grounded approach in existing hybrid models highlights the need for more innovative solutions that can directly address the complexities of gust prediction.

In this paper, we propose an innovative solution that addresses the limitations of hybrid models, which rely heavily on data decomposition without offering clear physical insights. Our method focuses on detecting gusts directly from wind speed data by leveraging a dynamic window adjustment mechanism. This mechanism allows for the detection of gusts without the need for manual labeling, providing a more interpretable, physically grounded approach. By dynamically adjusting the observation window based on wind speed fluctuations, our method can accurately identify gust events, which are then used as input conditions for a conditional LSTM framework, as demonstrated in previous studies [51,52]. This approach integrates a novel combination of dynamic gust detection with the CLSTM model structure, specifically designed to adapt to rapid wind speed changes that occur during gust events. The CLSTM framework leverages these identified gust events as conditioning inputs, allowing the model to dynamically adjust its sequence predictions based on the presence and characteristics of gusts. This design enhances the model's ability to capture and respond to sudden fluctuations, providing a level of robustness and adaptability not typically found in standard LSTMs or hybrid models. Such a capability is particularly effective for ultra-short-term wind speed forecasting, where rapid changes in wind speed are critical for accurate prediction and reliable wind

turbine operation. The unique integration of the dynamic gust detection mechanism and the CLSTM architecture not only improves ultra-short-term wind speed prediction but also enhances the model's ability to handle rapid wind speed changes, making it particularly effective during gust events. The main contributions of this paper are as follows.

(1) The Conditional LSTM framework incorporates gust detection, adapting predictions to gust and non-gust conditions, which significantly improves forecasting accuracy during gust events.
(2) A novel approach to detecting gusts based on dynamic window adjustment is proposed, capturing rapid fluctuations in wind speed more accurately by adjusting the window size based on wind speed change rate.
(3) Systematic experiments and comparisons were conducted to perform an in-depth analysis of the performance of different models under gust and non-gust conditions, providing a comprehensive evaluation of various modeling approaches.

The rest of this paper is organized as follows. Section 2 proposes the methodology, including the dynamic gust detection algorithm and the Conditional LSTM model. Section 3 presents the experimental details and baseline models. Section 4 discusses the implications of the results. Section 5 concludes the paper with suggestions for future research directions.

2. Methodology

This section outlines the key components of the proposed framework, focusing on the Dynamic Gust Detection method and the Conditional Long Short-Term Memory (Conditional LSTM) model for ultra-short-term wind speed forecasting. The primary goal is to identify gust events with high accuracy and improve forecasting by using dynamic window adjustments and scenario-based modeling.

2.1. Task Definition

The primary task in this study is ultra-short-term wind speed prediction. The aim is to forecast wind speed in the next few minutes based on past wind speed and meteorological variables, focusing on accurately predicting gust events. Gust events are characterized by sudden and rapid changes in wind speed, which are particularly challenging to predict due to their short duration and highly dynamic nature.

Given a sequence of past wind speeds and other relevant variables, the objective is to predict wind speed over a brief horizon H (typically between 1 and 10 min). The input sequence for time t can be represented as $X_t = \{x_1, x_2, ..., x_t\}$, where x_t is the wind speed or meteorological measurement at time t. The goal is to predict the future wind speed sequence $\hat{X}_{t+1:t+H}$, such as:

$$\hat{X}_{t+1:t+H} = \{\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H}\} \quad (1)$$

Gust events are identified as rapid wind speed changes that exceed a specific threshold (e.g., 20%) of the average wind speed over a certain time period, with a maximum duration of 1 min. This average wind speed is calculated using the dynamic window, as introduced in Section 2.2. The task is to detect these events within the forecasted wind speed time series. The model not only needs to predict future wind speeds but also to identify and tag gust events, allowing for wind farm optimization.

The detection of gust events relies on both the magnitude of wind speed change and the rate of change. A gust event is detected if the wind speed changes by more than a threshold within a short window of time. This threshold is dynamically adjusted based on data using the dynamic window mechanism outlined in Section 2.2.

2.2. Gust Detection

Dramatic fluctuations in wind speed are usually manifested as localized maxima, i.e., the wind speed increases sharply and reaches a peak in a short period of time, which is known as the gust. To better identify gust events, this study adopts a gust identification

method based on local maxima and incorporates a dynamic sliding window technique, which enables the method to flexibly adjust the size of the window according to the rate of change of the wind speed, thus balancing noise suppression and gust retention. With this method, the occurrence of gusts can be effectively captured and the noise and outliers in the wind speed signal can be suppressed.

Gusts are usually characterized by several key features:

- Sudden: A gust is a sharp increase in wind speed over a short period of time that quickly reaches a peak.
- Localized Extremes: Gusts exhibit localized extremes of wind speed, rather than minima, meaning that wind speeds peak at some point.
- High-frequency Fluctuations: gusts are characterized by high frequencies compared to the overall change in wind speed.

Next, we will provide a detailed description of the specific implementation steps.

The rate of change of wind speed is an important indicator of drastic changes in wind speed. Given a time series of wind speed measurements $W(t)$, the wind speed change rate $\Delta W(t)$ at time t is computed as the relative difference between successive measurements:

$$\Delta W(t) = \frac{W(t) - W(t-1)}{W(t-1)} \tag{2}$$

This relative change rate provides a normalized measure of wind speed variability, which is crucial for adjusting the window size. The dynamic window size at each time step is determined by the magnitude of $\Delta W(t)$: if the absolute value of $\Delta W(t)$ exceeds a pre-defined change threshold $T_{\Delta W}$, the window size is reduced to focus on capturing high-frequency gusts; if $\Delta W(t)$ is below the threshold, the window size is increased to consider slower variations in wind speed. The dynamic window size $W_{\text{dyn}}(t)$ at time t is updated according to:

$$W_{dyn}(t) = \begin{cases} max(W_{\text{min}}, W_{\text{base}} - k \cdot \Delta W(t)) & \text{if } \Delta W(t) > T_{\Delta W}, \\ min(W_{\text{max}}, W_{\text{base}} + k \cdot (T_{\Delta W} - \Delta W(t))) & \text{otherwise.} \end{cases} \tag{3}$$

where W_{base} is the base window size, used to balance noise suppression with gust capture; W_{min} and W_{max} are the lower and upper limits of the window size, respectively, which ensure that the window is not too large or too small; k is the window adjustment factor, used to control the sensitivity of the window size to the rate of change of wind speed $\Delta W(t)$; and $T_{\Delta W}$ is the threshold value for the rate of change of wind speed. When the rate of change of the wind speed exceeds this threshold, it is considered that a drastic change (gust) has occurred.

By dynamically adjusting the window size, this method narrows the window when the wind speed changes drastically to better capture gusts and increases the window when the wind speed is smooth to reduce noise interference.

Once the dynamic window size has been determined for each time step, smoothed wind speed values are used to reduce the effect of noise. The wind speed smoothing process is crucial to ensure that minor fluctuations or random noise in the raw data do not trigger false gust detections. The smoothed wind speed at time step t, denoted as $W_{\text{smooth}}(t)$, is computed using a weighted moving average of the wind speed values within the dynamic window:

$$W_{\text{smooth}}(t) = \frac{\sum_{i=t-W_{\text{dyn}}}^{t} w_i \cdot W(i)}{\sum_{i=t-W_{\text{dyn}}}^{t} w_i} \tag{4}$$

where w_i represents the weight assigned to each wind speed value $W(i)$ within the window, with higher weights assigned to values closer to the current time step t. This ensures that more recent data has a greater influence on the smoothed result.

After applying the smoothing process, local maxima are identified within each dynamic window. A local maximum is defined as a smoothed wind speed value that exceeds all other smoothed values within the current window. For a given time t, the wind speed $W_{\text{smooth}}(t)$ is considered a local maximum if it satisfies the following condition:

$$W_{\text{smooth}}(t) = max(W_{\text{smooth}}(t - W_{\text{dyn}}(t)/2), W_{\text{smooth}}(t + W_{\text{dyn}}(t)/2)) \quad (5)$$

This condition ensures that the wind speed at time step t is the highest within the current dynamic window, indicating a potential gust event.

Furthermore, to confirm the presence of a gust, we apply a dynamic threshold on the wind speed difference $\Delta W_{diff}(t)$. This threshold adapts to the long-term wind speed variations in the dataset, ensuring that only significant wind speed changes are identified as gusts.

The threshold $T_{diff}(t)$ for each time step is calculated dynamically based on the long-term standard deviation $\sigma_W(t)$ of wind speed over a predefined period:

$$T_{diff}(t) = k_{threshold} \cdot \sigma_W(t) \quad (6)$$

where $k_{threshold}$ is a scaling factor and $\sigma_W(t)$ is the standard deviation of wind speed over the previous time steps. A detected peak is classified as a gust if the wind speed difference exceeds the dynamic threshold:

$$\Delta W_{diff}(t) = \left| W_{smooth}(t) - W_{smooth}(t - W_{dyn}(t)/2) \right| > T_{diff}(t) \quad (7)$$

By adjusting the threshold dynamically, this method can adapt to varying wind conditions, capturing gusts across different wind regimes and avoiding false positives caused by normal fluctuations.

2.3. Conditional Sequence Modeling

To improve the accuracy of wind speed predictions, especially in gust scenarios, we propose a Conditional LSTM model. This model integrates gust detection with sequence modeling to better capture the effects of abrupt changes in wind speed. By incorporating gust embeddings, attention mechanisms, and gating units, the model can effectively identify and predict gust events, enhancing its performance in gust conditions.

Figure 1 shows that the conditional LSTM model consists of the following key components:

1. Gust Embedding Layer: This layer embeds the gust labels (binary: gust or non-gust) into a continuous vector space. The gust state, represented as 0 (non-gust) or 1 (gust), is embedded into a 4-dimensional vector.
2. LSTM Layer: The input wind speed sequence, combined with the gust embeddings, is passed into the LSTM network. The LSTM processes the sequence of wind speed data while considering the gust state at each time step.
3. Attention Mechanism: After the LSTM layer, a multi-head attention mechanism is applied. This allows the model to assign different importance to each time step, especially focusing on periods when gusts are detected.
4. Conditional Gating Mechanism: The gust embedding is used in a gating mechanism to control the flow of information between the attention output and the final prediction. This helps dynamically adjust the model's sensitivity based on whether a gust is detected.

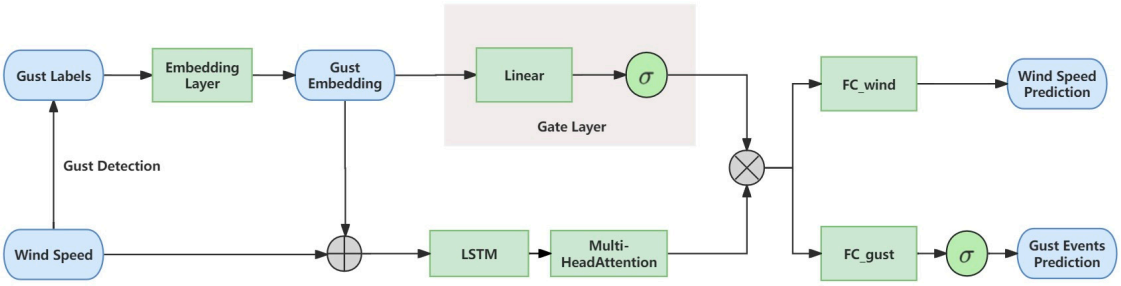


Figure 1. Flowchart of the Conditional LSTM model. The green boxes highlight the functional layers of the model, including the Gust Embedding Layer, LSTM Layer, Attention Mechanism, and Conditional Gating Mechanism. The blue boxes indicate the data flow through the model. The gray box highlights the Conditional Gating Mechanism, which uses the gust embedding to control the flow of information between the attention output and the final prediction.

2.3.1. Gust Embeddings

The embedding layer is a common component in deep learning models, especially when dealing with categorical or discrete inputs. Its role is to convert discrete gust labels g_t into continuous dense vector representations (embedding vectors). The dimension of these embedding vectors is d_g, and they are looked up using an embedding matrix $E \in \mathbb{R}^{2\times d_g}$, where 2 represents the two states of gusts (gust present and no gust):

$$g_t = E(g_t), g_t \in \mathbb{R}^{d_g} \tag{8}$$

The primary function of the embedding layer is to map discrete categories (gust labels) into a fixed-dimensional continuous space, allowing the model to capture the influence of gust states on wind speed dynamics during the learning process. These embedding vectors g_t are concatenated with the wind speed input x_t to form the input to the LSTM layer:

$$z_t = \begin{bmatrix} x_t \\ g_t \end{bmatrix}, z_t \in \mathbb{R}^{d_x+d_g} \tag{9}$$

where $x_t \in \mathbb{R}^{d_x}$ represents the wind speed input at time step t and $g_t \in \mathbb{R}^{d_g}$ represents the gust embedding at time step t.

2.3.2. Long Short-Term Memory

LSTM (Long Short-Term Memory) is a specialized neural network architecture designed for handling time series data. It captures long-term and short-term dependencies in sequences through the introduction of memory cells and gating mechanisms. Figure 2 shows that the internal structure of an LSTM includes three gates: the input gate, the forget gate, and the output gate. These gates control how information flows and is retained across time steps.

In our model, the input to the LSTM is the concatenated vector z_t of wind speed x_t and gust embedding g_t. The hidden state x_t and cell state c_t of the LSTM are updated at each time step based on the previous states and the current input.

The update process of the LSTM is as follows:

$$h_t, c_t = \text{LSTM}(z_t, h_{t-1}, c_{t-1}) \tag{10}$$

where $h_t \in \mathbb{R}^{d_h}$ is the hidden state at time step t, representing the model's memory of the current time step input, $c_t \in \mathbb{R}^{d_h}$ is the cell state at time step t, storing long-term memory, and d_h is the dimension of the LSTM's hidden layer.

Through its internal gating mechanisms, LSTM effectively captures the relationship between transient changes (via the hidden state x_t) and long-term dependencies (via the

cell state c_t). This makes it particularly suitable for time series prediction of wind speed and gust events.

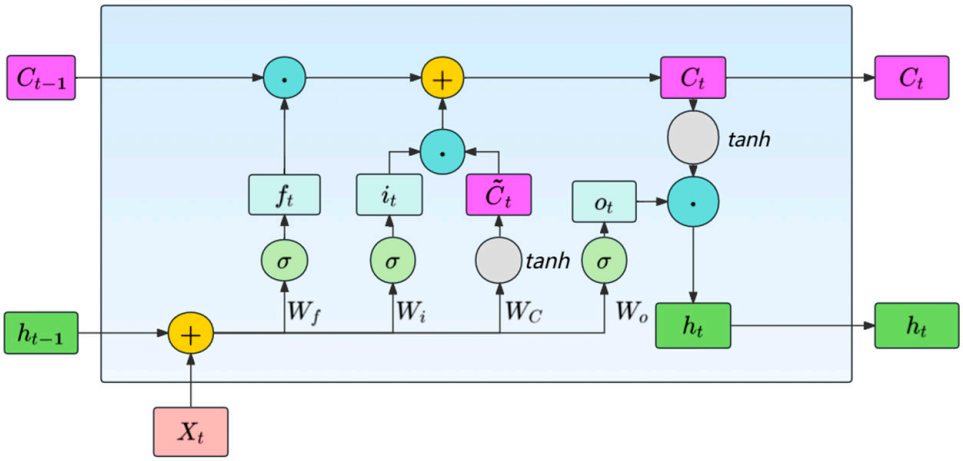


Figure 2. The structure of LSTM. The big blue box represents the LSTM cell for the current time step. The pink boxes represent the input at the current time step and the candidate cell state. The green boxes represent the hidden states from the previous and current time steps, while the purple boxes show the cell states. The cyan boxes represent the forget, input, and output gates. Yellow circles indicate element-wise operations, and gray circles represent activation functions (tanh).

2.3.3. Attention Mechanism

The attention mechanism is a technique that allows the model to assign different weights to different time steps, enabling it to focus on the most relevant ones. Our model employs a multi-head attention mechanism, which computes and aggregates attention weights in multiple subspaces, thereby enhancing the model's representational power.

Figure 3 shows that the core idea of the attention mechanism is to allocate weights based on the similarity between the current hidden state h_t (as the query Q) and all the hidden states across the sequence (as the keys K and values V). The specific computation is as in (6).

$$A = softmax\left(\frac{QK^\top}{\sqrt{d_h}}\right) \tag{11}$$

where $Q = h_t$ is the hidden state at the current time step, serving as the query matrix, $K = h$ are the hidden states of all time steps, serving as the key matrix, and d_h is the dimension of the hidden layer.

The attention output O_t is obtained by a weighted sum from the value matrix V:

$$O_t = AV, V = h_t \tag{12}$$

In our model, the multi-head attention mechanism computes multiple attention heads in parallel, capturing multi-scale dependencies between different time steps. This mechanism allows the model to handle long-term dependencies more flexibly, ensuring that it can focus on key time steps in wind speed prediction and gust detection tasks.

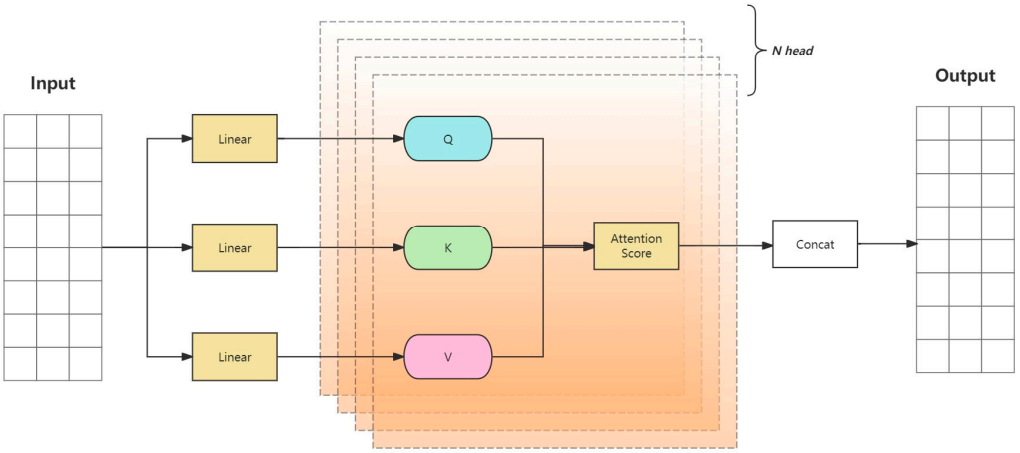


Figure 3. Diagram of the Multi-Head Attention Mechanism, where the input is the hidden state h_t from the LSTM output. The yellow boxes represent linear layers that project the input into query, key, and value spaces. The cyan box represents the query vectors (Q), the green box represents the key vectors (K), and the pink box represents the value vectors (V). The dotted-line boxes represent multiple attention heads, where attention is computed independently for each head. These heads are then concatenated and passed through a linear layer to produce the final output.

2.3.4. Conditional Gating Mechanism

To further adjust the output of the attention mechanism, we introduce a gating mechanism. The role of the gating mechanism is to modulate the impact of the attention output based on the gust embedding g_t. Through gating weights w_t, the model can dynamically adjust the prediction results according to the gust state. The gating weights are computed as in (8).

$$w_t = \sigma\left(W_g g_t + b_g\right), w_t \in \mathbb{R}^{d_h} \tag{13}$$

where W_g is a learned weight matrix, g_t is the gust embedding, and σ is the sigmoid activation function.

The final gated output O_t^{gated} is obtained by element-wise multiplication of the attention output O_t, and the gating weights w_t can be calculated as in (9).

$$O_t^{gated} = O_t \odot w_t \tag{14}$$

The gating mechanism allows the model to control the magnitude of the attention output based on the current gust state, thereby enhancing the flexibility and accuracy of the predictions.

2.4. Loss Function Design

The model is trained with a composite loss function, which is a weighted sum of the MSE for wind speed prediction and the BCE for gust event detection:

$$L = \lambda_1 L_{\text{wind}} + \lambda_2 L_{\text{gust}} \tag{15}$$

where L_{wind} is the MSE for wind speed prediction, which can be calculated by (11), L_{gust} is the BCE for gust event detection, which can be calculated by (12), and λ_1 and λ_2 are hyperparameters that balance the contributions of the two loss terms.

$$L_{\text{wind}} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t^{\text{wind}} - \hat{y}_t^{\text{wind}}\right)^2 \tag{16}$$

$$L_{\text{gust}} = -\frac{1}{T}\sum_{t=1}^{T}\left[\begin{array}{c} y_t^{\text{gust}}\log\hat{y}_t^{\text{gust}} + \\ \left(1-y_t^{\text{gust}}\right)\log\left(1-\hat{y}_t^{\text{gust}}\right)\end{array}\right] \tag{17}$$

where T is the total number of time steps.

The detail training process of the Conditional LSTM model is shown in Algorithm 1.

Algorithm 1 Training the Conditional LSTM Model

Require: Training dataset D, gust state labels G
Ensure: Optimal model parameters for LSTM, embedding, attention, and gating
1. θLSTM, θgust_embed, θattention, θgating ← initialize network parameters
2. repeat
3. x ← random mini-batch from D // Wind speed time series
4. gust_labels ← random mini-batch from G // Corresponding gust labels
5. gust_embed ← gust_embedding(gust_labels) // Get gust state embeddings
6. gust_embed_expanded ← expand(gust_embed) // Expand embeddings over time steps
7. combined_input ← concatenate(x, gust_embed_expanded) // Concatenate input and gust embeddings
8. lstm_output, (hn, cn) ← LSTM(combined_input) // Pass through LSTM
9. attn_output, _ ← MultiheadAttention(lstm_output, lstm_output, lstm_output) // Apply attention mechanism
10. attn_output_avg ← mean(attn_output, dim = 1) // Average over time dimension
11. gating_weights ← Sigmoid(gating_layer(gust_embed)) // Apply gating mechanism
12. gated_output ← attn_output_avg * gating_weights // Apply gating to attention output
13. wind_output ← FullyConnected(gated_output) // Predict wind speed
14. gust_output ← Sigmoid(FullyConnected(gated_output)) // Predict gust event (binary classification)
15. ℓwind ← ||x-wind_output||^2 // Compute wind speed loss (MSE)
16. ℓgust ← binary_cross_entropy(gust_output, gust_labels) // Compute gust event loss (Binary Cross-Entropy)
17. // Total loss for optimization
18. ℓtotal ← ℓwind + λ × ℓgust // Combine losses, where λ is a hyperparameter balancing losse
19. // Update parameters using gradient descent
20. θLSTM ← θLSTM-∇θLSTM(ℓtotal)
21. θgust_embed ← θgust_embed-∇θgust_embed(ℓtotal)
22. θattention ← θattention-∇θattention(ℓtotal)
23. θgating ← θgating-∇θgating(ℓtotal)
24. until convergence or max iterations

3. Experiment Setup

3.1. Dataset Description

The dataset utilized in this study comprises real-time observations from an offshore wind farm in a specific province in China, which includes 50 turbines, each with a capacity of 4 MW. Meteorological parameters such as wind speed, wind direction, and temperature were collected at one-minute intervals over a six-month period, yielding approximately 250,000 wind speed records. The measured wind speed may correspond to the hub-height wind speed, potentially differing from the wind speed experienced by the blades. To prepare the dataset for analysis, we identified the longest continuous time series without missing values to maintain the integrity of the time series and prevent model training biases. Additionally, all variables underwent normalization to a mean of 0 and a standard deviation of 1, mitigating the impact of scale differences on model performance. Gust events within the dataset were identified and labeled using a dynamic window approach, characterized by abrupt and substantial variations in wind speed over brief durations; the specific criteria for detecting these gust events are outlined in Section 3.2.

3.2. Experiment Settings

3.2.1. Dataset Split

The dataset is split into a training set, validation set, and test set based on chronological order to maintain time dependencies. The dataset is divided into 70% for training, 15% for

validation, and 15% for testing. This split ensures that the model can generalize well to unseen data, with validation performance guiding the hyperparameter selection.

3.2.2. Hyperparameter Settings

In this study, several hyperparameters were configured for the Conditional LSTM model to optimize performance in predicting wind speed and identifying gust events. The key hyperparameters used include the input size, determined by the number of features extracted from the dataset, and a hidden size of 64, allowing the model to capture complex patterns in the data. The gust embedding dimension was set to 16, providing a compact representation of gust event information. The output size was configured to 1, corresponding to the predicted wind speed, while the number of LSTM layers was kept to **1** to maintain a simpler model architecture.

The batch size during training was set to 32, balancing memory efficiency with convergence stability. The Adam optimizer was employed with a learning rate of 0.001, facilitating effective optimization of the loss functions. These hyperparameter settings were chosen based on preliminary experiments to enhance the model's performance.

3.2.3. Training Details

The model was trained for 10 epochs, each consisting of a training phase and a subsequent validation phase. During training, the model operated in training mode, allowing gradients to be computed for backpropagation. For each batch of data from the training set, the model's predictions for both wind speed and gust events were generated. The model was trained using the loss functions described in Section 2.4 for wind speed and gust event predictions. The combined loss was then backpropagated, updating the model parameters through the Adam optimizer.

After completing each epoch, the model underwent evaluation on the validation dataset to monitor its performance and generalization capabilities. Average validation losses for both wind speed and gust predictions were computed, allowing for the early detection of potential overfitting or underfitting issues. The training and validation losses were printed at the end of each epoch, providing insights into the model's learning process.

The experiments were implemented using PyTorch (2.3.1) and executed on a Windows PC equipped with a 13th Gen Intel Core i5-13500H CPU (Manufacturer: Intel Corporation, Santa Clara, CA, USA), featuring 12 cores and 16 logical processors operating at a base speed of 2.60 GHz. The training utilized an NVIDIA GeForce RTX 4050 Laptop GPU (Manufacturer: NVIDIA Corporation, Santa Clara, CA, USA) with 6 GB of dedicated memory, ensuring efficient processing and model evaluation.

3.2.4. Window Size Settings

In this study, we employed a dynamic window size approach for detecting gust events in wind speed data. The window size refers to the number of time steps considered for calculating the smoothed wind speed and identifying local maxima.

Initially, we assessed the range of realistic values based on the behavior of our dataset. From the original wind speed data of a randomly selected day in Figure 4, it is evident that there is a wind speed trend throughout the day, accompanied by random fluctuations, which necessitated the smoothing of wind speed. The base window size of five time steps was selected, and we found that changing this base window size in subsequent experiments had little to no impact. The minimum window size was set to one time step based on the time step of the dataset, meaning no further iterations were needed. The maximum window size was determined through iterative testing, and as shown in the last three plots in Figure 4, ten time steps proved optimal. Beyond this, excessive smoothing could lead to the smoothed wind speed diverging too far from the true data, causing misidentification of gusts. Given that the window size range is from 1 to 10, a window adjustment factor of 2 was chosen. For the critical change threshold, as shown in the first four plots, a threshold of 0.2 was found to be more accurate compared to 0.1 and 0.3 and thus was selected. Finally,

the long-term window for calculating the dynamic threshold was set to 1 h, or sixty time steps, based on empirical observation, as this better reflects long-term trends. Further changes in the long-term window size did not significantly affect the results.

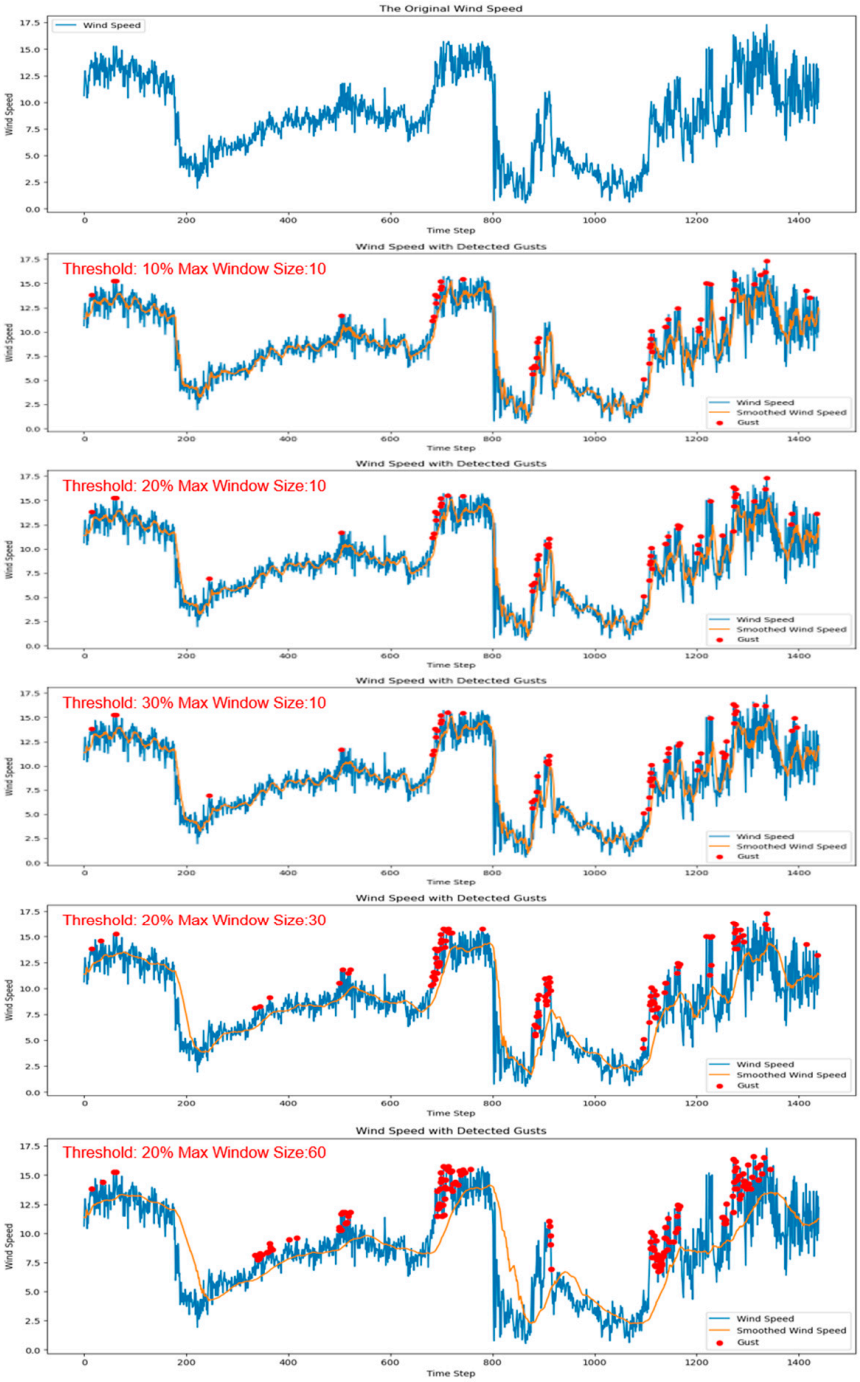


Figure 4. Original wind speed and wind speed with detected gusts. From top to bottom, the plots display the wind speed data for the same day: 1. The original wind speed; 2. Threshold: 10%, Max Window Size: 10—original wind speed, smoothed wind speed, and detected gust points; 3. Threshold: 20%, Max Window Size: 10; 4. Threshold: 30%, Max Window Size: 10; 5. Threshold: 20%, Max Window Size: 30; 6. Threshold: 20%, Max Window Size: 60. The blue line represents the original wind speed, while the orange line shows the smoothed wind speed derived from the dynamic window method. Red points indicate gust events detected using this same dynamic window approach.

The dynamic window size is adjusted based on the rate of change in wind speed, specifically using a change threshold of 0.2. If the change in wind speed exceeds this threshold, the window size is reduced to better focus on the rapid variations associated with gusts. Conversely, when changes are minimal, the window size increases, up to a maximum of ten time steps, ensuring that longer-term trends are not overlooked.

Additional parameters include a minimum window size of one time step to avoid excessively small windows during extreme fluctuations, and a window adjustment factor of 2, which governs the rate of adjustment in response to wind speed changes. This dynamic adjustment allows the model to remain responsive to varying conditions, improving the detection of gust events.

To enhance detection accuracy, a long-term window of sixty time steps is utilized to calculate a dynamic threshold based on the standard deviation of wind speed. This threshold aids in discerning significant gusts from normal variations, ensuring robustness in the detection process.

Overall, this approach of dynamic window sizing is crucial for capturing the transient nature of gust events, thereby enhancing the model's predictive capability in ultra-short-term wind speed forecasting.

3.3. Baseline Models

To comprehensively evaluate the predictive performance of the proposed method in gust scenarios, we selected multiple common time series prediction models as baselines, covering statistical models, classical machine learning models, and deep learning models.

The following models were used as baselines for comparison:

- AutoRegressive (AR): This statistical model predicts future values based on past values, capturing the linear relationships in the wind speed data.
- AutoRegressive Moving Average (ARMA): Combining AR and moving average components, ARMA models the time series by considering both past values and past errors, allowing for more nuanced predictions.
- Backpropagation Neural Network (BPNN): A feedforward neural network that uses backpropagation for training. This model is capable of capturing complex nonlinear relationships in the data.
- Long Short-Term Memory (LSTM): A recurrent neural network designed to capture long-range dependencies in time series data, LSTM is well-suited for sequential prediction tasks such as wind speed forecasting.
- Support Vector Machine (SVM): A robust regression technique that identifies the optimal hyperplane for prediction, SVM is effective in handling high-dimensional data and nonlinear relationships.
- Empirical Mode Decomposition-Support Vector Machine (EMD-SVM): This model first decomposes the wind speed signal into intrinsic modes using EMD, followed by SVM regression to predict wind speed, allowing for a detailed analysis of the underlying dynamics.
- Extreme Learning Machine (ELM): A single-hidden layer feedforward neural network that offers rapid training and good generalization performance, ELM is included for its efficiency in handling large datasets.

To ensure fair comparisons and optimal performance for each baseline, we employed Bayesian optimization to determine the ideal hyperparameters for models with tunable parameters. This method allowed for efficient exploration of the high-dimensional parameter space, reducing the computational cost of manual trial-and-error searches. Table 1 summarizes the structure and final hyperparameter values for each model after Bayesian optimization.

Table 1. Summary of model structures and parameter configurations.

Models	Structure (e.g., Layers/Nodes)	Parameters	Hyper-Parameters
AR	Lag order: p	p: 5	-
ARMA	AR order: p, MA order: q	p: 5, q: 1	-
BPNN	Input: n, Hidden: x, Output: y	Activation: ReLU	Learning Rate: 0.01, Batch Size: 16
LSTM	Layers: 3, Units: 100	-	Learning Rate: 0.003, Batch Size: 32
SVM	Kernel: rbf	C: 13.88, Gamma: 0.02	-
EMD-SVM	EMD + Kernel: rbf	C: 13.88, Gamma: 0.02	-
ELM	Input: n, Hidden: x, Output: y	Activation: Sigmoid	Hidden Dim: 123
CLSTM (Ours)	LSTM Layers: 1, Units: 64, Gust Embedding: 16	-	Learning Rate: 0.001, Batch Size: 32

Simpler models, such as AR and ARMA, contain fewer parameters, resulting in reduced computational demands but limited ability to capture complex, non-linear dependencies in wind speed data. Conversely, more complex models such as BPNN, LSTM, and our proposed CLSTM incorporate additional layers and nodes, which enhance their capability to model intricate temporal dynamics at the cost of increased computational resources. The CLSTM model in particular includes specialized gust embedding layers designed to address gust detection directly, setting it apart in terms of targeted complexity for this application. These variations in model architecture and hyperparameter choices reflect a deliberate balance between accuracy, efficiency, and interpretability, thereby providing a more comprehensive understanding of each model's suitability for ultra-short-term wind forecasting.

3.4. Evaluation Metrics

To comprehensively evaluate the predictive performance of the models, we selected the following evaluation metrics, including overall metrics and specific metrics for gust events:

1. Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{18}$$

This metric measures the average of the squared differences between the predicted values and the actual values. It amplifies larger prediction errors. A smaller MSE indicates higher prediction accuracy.

2. Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \tag{19}$$

RMSE provides a measure of the average magnitude of the prediction errors, allowing for a more interpretable assessment of model performance.

3. Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}|\hat{y}_i - y_i| \tag{20}$$

MAE measures the average absolute differences between the predicted values and the actual values. It reflects the overall prediction error size.

Given that our model is specifically tailored for gust scenarios, it is essential to assess its performance during gust periods. The model not only predicts gust events but also encounters an imbalance in sample sizes, necessitating careful consideration of these factors in the evaluation process. Therefore, we have selected evaluation metrics specifically designed for imbalanced regression [53] and classification to ensure that the model's performance under gust conditions is accurately reflected.

4. Squared Error-Relevance Area (SERA):

$$\text{SERA} = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot (y_i - \hat{y}_i)^2 \tag{21}$$

where N is the total number of samples, w_i is the weight of the i-th sample, reflecting its importance, y_i is the true value of the i-th sample, and $\hat{y}_i$ is the predicted value of the i-th sample. By assigning different weights to different samples, SERA can better reflect the model's performance in handling imbalanced data. For example, in gust event prediction, the prediction errors of gust events (the minority class) are assigned higher weights, thus giving more emphasis to the accuracy of these events in evaluating model performance.

5. Residual Analysis:

$$\text{Residual}_i = y_i - \hat{y}_i \tag{22}$$

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i) \tag{23}$$

$$\sigma_r = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\text{Residual}_i - \bar{r})^2} \tag{24}$$

The residual represents the prediction error for each sample. The mean residual provides an overall measure of bias in the model across all samples; a positive value indicates an overall overestimation, while a negative value indicates underestimation. The standard deviation quantifies the dispersion of the residuals, reflecting the stability of the model's predictions.

Residual analysis is a crucial step in evaluating the performance of predictive models, as it provides insights into the discrepancies between predicted and actual values. By examining the distribution and patterns of residuals, we can identify potential biases in the model's predictions and assess whether the assumptions of the regression model hold true.

In the context of gust predictions, analyzing the residuals associated with both gust and non-gust events allows for a more nuanced understanding of model performance. For instance, if residuals exhibit a systematic pattern, this may indicate that the model is failing to capture certain dynamics associated with gust events or that it may be overfitting to non-gust conditions. Moreover, evaluating the mean and standard deviation of residuals can help assess the model's ability to handle varying wind conditions, particularly during sudden fluctuations.

6. Statistical Significance Test:

Since the time series data may not meet the assumption of normality, the Wilcoxon signed-rank test method was used. The Wilcoxon signed-rank test is a non-parametric statistical test used to determine whether there is a significant difference between paired samples.

To perform the test, the differences between the paired values are calculated, the absolute values of these differences are ranked, and the signs of the differences are assigned to the ranks. The test statistic $\boldsymbol{W}$ is the smaller of the sum of the positive ranks T_+ and the sum of the negative ranks T_-. The p-value is calculated to determine if the performance of different models is significantly different from the baseline model (in this case, CLSTM). For small sample sizes, the p-value is derived using exact methods from the Wilcoxon signed-rank distribution table. For large sample sizes, the test statistic is approximated by a normal distribution and the Z-score is calculated using the expected value and variance of $\boldsymbol{W}$:

$$Z = \frac{W - \mu_W}{\sigma_W} \tag{25}$$

where μ_W and σ_W are the expected value and standard deviation of W, respectively.

The *p*-value is then computed based on the Z-score using the standard normal cumulative distribution function:

$$P = 2 \times (1 - \Phi(|Z|)) \tag{26}$$

where $\Phi(|Z|)$ is the CDF of the standard normal distribution.

The obtained *p*-value determines if there is a statistically significant difference between the models. A *p*-value below a chosen significance level (e.g., 0.05) indicates that the performance of the model being tested is significantly different from the baseline model (in this case, CLSTM).

4. Results and Discussion

This section presents the results of our wind speed prediction models and discusses their performance in both overall and gust events. We compare the performance metrics of eight different models: AR, ARMA, BPNN, LSTM, SVM, EMS-SVM, ELM, and CLSTM (our proposed model).

4.1. Overall Performance Comparison

As shown in Table 2, the proposed CLSTM model achieves the lowest mean squared error (MSE) of 0.4893, with a root mean square error (RMSE) of 0.6995 and a mean absolute error (MAE) of 0.542. The Squared Error-Relevance Area (SERA) is recorded at 0.3613. This series of metrics indicates that the CLSTM demonstrates superior accuracy in wind speed forecasting. In contrast, the SERA values introduce a higher weight for errors during gust events, allowing for a reflection of the model's performance in the face of sudden changes.

Table 2. Overall performance comparison of wind speed prediction models.

Models	MSE	RMSE	MAE	SERA	*p*-Value
AR	14.667	3.8298	3.2496	10.1960	0.0
ARMA	14.9524	3.8668	3.2636	10.4607	0.0
BPNN	0.6469	0.8043	0.6245	0.8219	0.403
LSTM	0.5394	0.7344	0.5686	0.9320	0.0109
SVM	0.5627	0.7501	0.582	0.9343	0.0702
EMD-SVM	1.0838	1.041	0.8809	0.8602	3.41×10^{-123}
ELM	0.6815	0.8255	0.6502	1.8556	1.22×10^{-51}
CLSTM (Ours)	0.4893	0.6995	0.542	0.3613	base

Notably, only the AR, ARMA, EMD-SVM, and CLSTM models exhibit lower errors during gust periods compared to overall errors. Figure 5 reveals that the AR and ARMA models demonstrate this behavior because they generally predict higher wind speeds, aligning well with the characteristics of gusts. Consequently, their SERA values are lower than their respective MSEs, with MSEs of 14.667 and 14.9524 corresponding to SERA values of 10.1960 and 10.4607, respectively, highlighting a significant difference between the two. The *p*-value of 0 for both AR and ARMA models indicates that the difference between their predictions and the CLSTM model is statistically significant, reinforcing the idea that these models have notable differences in performance. This suggests that while these models capture some aspects of gust behavior, their general prediction accuracy remains inferior to CLSTM's more precise gust detection, as reflected in the overall lower SERA values.

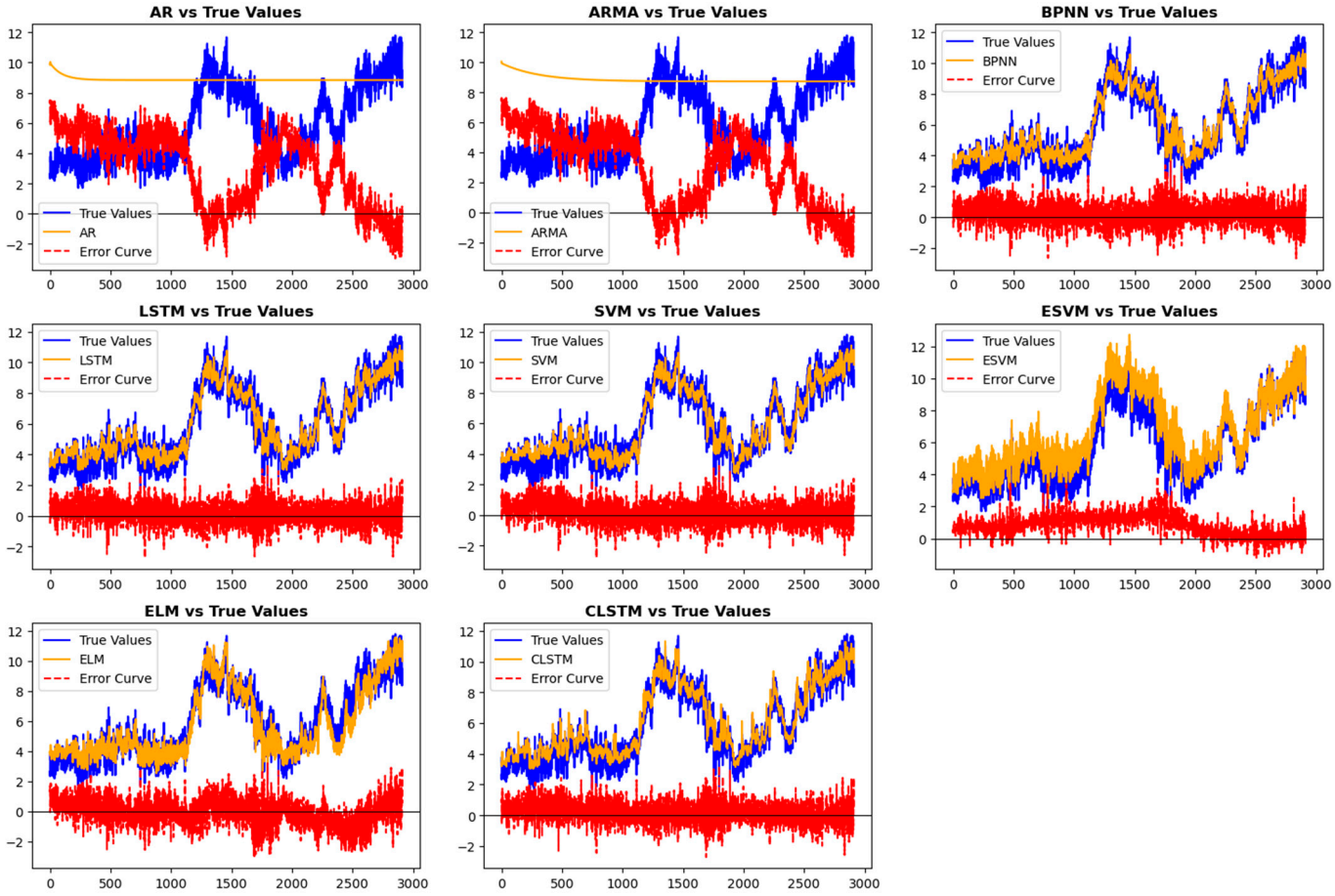


Figure 5. Comparison of prediction and true value curves for all models. The blue line represents the true curve, the orange line shows the model prediction curve, the red dashed line indicates the error curve and the black line represents the zero level on the y-axis. This figure provides a visual comparison of each model's performance in the wind speed prediction task. It is evident that the EMD-SVM hybrid model generally predicts higher values than the true wind speed, with its errors mostly positive compared to other models. Apart from the AR and ARMA models, which exhibit significantly large errors, and the EMD-SVM model, which consistently predicts higher values, the other models tend to be more conservative in their predictions. Notably, the proposed model makes more assertive predictions during wind speed peaks.

On the other hand, the EMD-SVM model performs relatively well during gust events, likely due to its ability to extract components through EMD decomposition, which enhances forecasting accuracy. However, as seen in Figure 4, the predictions of the EMD-SVM model are generally higher than the actual wind speeds, although the trend of the predictions aligns with the actual values. Therefore, the MSE is relatively high at 1.0838, but it is precisely this overprediction that results in a decrease in error during gust periods, leading to a SERA of 0.8809. The p-value of 3.41×10^{-123} further confirms that the EMD-SVM model's performance is statistically significantly different from the CLSTM model. Despite the alignment in the prediction trend, the EMD-SVM's inability to accurately predict gust magnitudes results in less precise predictions compared to CLSTM, which captures the wind speed changes more accurately, as reflected by its much lower SERA value of only 0.3613.

For models such as LSTM, BPNN, SVM, and ELM, although their overall performance is relatively good, as indicated by their MSE values, they show an interesting trend during gust events. Specifically, these models have a SERA value that increases compared to their MSE, suggesting that the errors during gust periods are more pronounced. BPNN's

p-value is 0.403, indicating that the difference between it and the CLSTM model is not statistically significant. LSTM's *p*-value is 0.0109, meaning that there is a statistically significant difference between it and CLSTM. SVM's *p*-value is 0.0702, which, while lower, still does not meet the 0.05 threshold, indicating that its significance is weak. ELM's *p*-value is 1.22×10^{-51}, showing a highly significant difference.

This phenomenon could be attributed to the fact that LSTM, with its recursive structure for time series modeling, can capture the temporal dependencies in wind speed, but it lacks additional mechanisms to handle gusts. As a result, it performs well in wind speed prediction but exhibits larger errors during gust predictions. On the other hand, CLSTM improves upon LSTM by adding conditional modeling and attention mechanisms, allowing the model to adapt to sudden changes in wind speed and better capture the dynamics of gust events. This explains why CLSTM's SERA is significantly lower than LSTM's and why the difference between the two models is statistically significant.

SVM, as a regression tool, adapts to data changes by finding the optimal decision boundary. BPNN, being a basic neural network model, is weaker in handling sequential data and capturing temporal dependencies. However, both models have relatively uniform errors during training, making their differences from CLSTM less statistically significant than LSTM.

To further analyze the robustness of the model, we conducted an error analysis under different wind speed conditions, and the results are shown in Table 3. Based on the wind speed distribution in the test set shown in Figure 6, we divided the data into four wind speed intervals using an equal-width approach: Very Low (0–3 m/s), Low (3–6 m/s), Moderate (6–9 m/s), High (9–12 m/s).

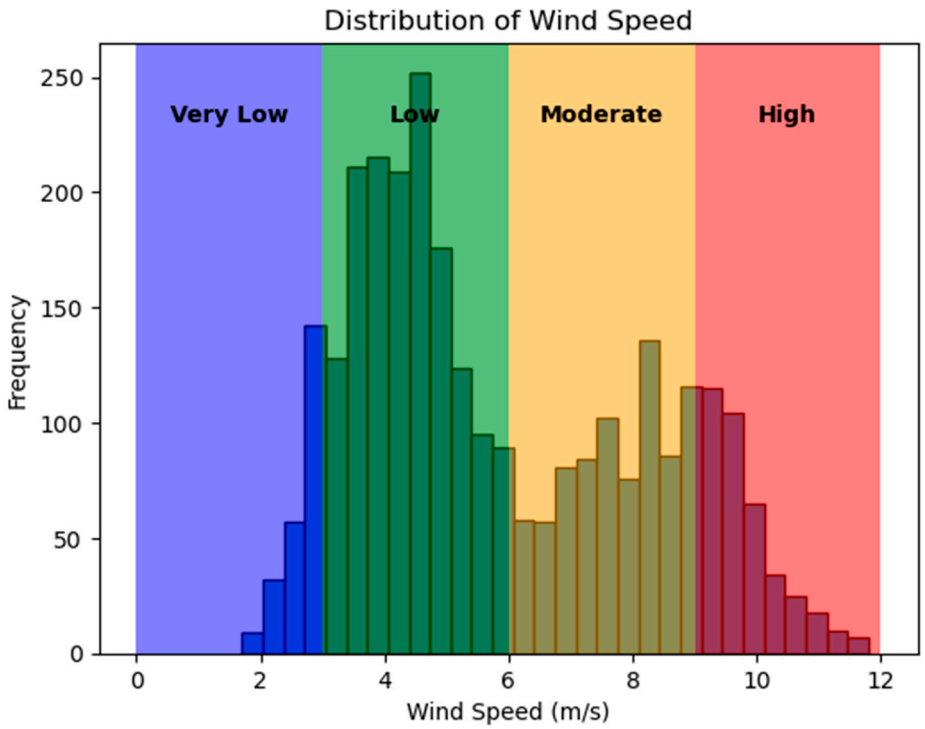


Figure 6. Wind speed distribution in Testset. Dividing into four wind speed intervals of equal width allows for testing all model performances under different wind conditions.

Table 3. Performance comparison under different wind speed conditions.

Wind Speed Condition	Models	MSE	MAE	*p*-Value
Very Low (0–3 m/s)	AR	41.392288	6.420835	2.26×10^{-36}
	ARMA	43.489571	6.578476	2.26×10^{-36}
	BPNN	1.679833	1.18879	5.34×10^{-2}
	LSTM	1.674148	1.192171	4.33×10^{-3}
	SVM	1.879408	1.282047	6.04×10^{-3}
	EMD-SVM	1.26687	1.00098	3.38×10^{-8}
	ELM	1.638098	1.179153	5.09×10^{-1}
	CLSTM (Ours)	1.728294	1.213017	base
Low (3–6 m/s)	AR	21.136846	4.524484	1.62×10^{-248}
	ARMA	21.453719	4.544553	1.62×10^{-248}
	BPNN	0.421482	0.50075	1.88×10^{-1}
	LSTM	0.381913	0.478559	3.72×10^{-56}
	SVM	0.405006	0.495406	5.75×10^{-99}
	EMD-SVM	1.087931	0.911125	1.41×10^{-180}
	ELM	0.476769	0.549381	1.15×10^{-139}
	CLSTM (Ours)	0.37987	0.480563	base
Moderate (6–9 m/s)	AR	1.976805	1.145495	6.60×10^{-96}
	ARMA	1.753842	1.061309	2.99×10^{-85}
	BPNN	0.491729	0.545699	1.15×10^{-42}
	LSTM	0.524757	0.561436	4.62×10^{-1}
	SVM	0.53245	0.565359	4.91×10^{-4}
	EMD-SVM	1.203638	0.894072	2.05×10^{-75}
	ELM	1.214961	0.891385	6.97×10^{-48}
	CLSTM (Ours)	0.486924	0.541944	base
High (9–12 m/s)	AR	1.23193	0.923843	3.59×10^{-55}
	ARMA	1.443489	1.031548	6.58×10^{-15}
	BPNN	0.515516	0.563487	4.44×10^{-1}
	LSTM	0.533526	0.579675	4.33×10^{-4}
	SVM	0.523497	0.572579	3.71×10^{-47}
	EMD-SVM	0.73916	0.679174	1.60×10^{-16}
	ELM	0.846833	0.71868	3.59×10^{-55}
	CLSTM (Ours)	0.509791	0.562863	base

In the Very Low wind speed interval (0–3 m/s), the performance of each model shows significant differences. The AR and ARMA models have MSE and MAE that are markedly higher than those of the other models, reaching 41.39 and 6.42 (AR) and 43.49 and 6.58 (ARMA), respectively. This indicates that these traditional time series models are ineffective in predicting at very low wind speeds, likely due to their limitations in handling non-stationary data. The *p*-values for these models are 2.26×10^{-36}, indicating that their differences from the CLSTM model are statistically significant. The BPNN, LSTM, SVM, ELM, and EMD-SVM models, as well as the CLSTM model, have relatively low MSE and MAE values, showing better predictive capability. Particularly, the EMD-SVM model, with MSE and MAE values of 1.27 and 1.00, respectively, performs the best, likely because the EMD method has decomposed low-frequency components. The *p*-value for EMD-SVM is 3.38×10^{-8}, also indicating a statistically significant difference from CLSTM. Other models show minimal differences in MSE and MAE. The *p*-values indicate that LSTM and SVM exhibit significant differences from CLSTM, while BPNN and ELM do not, suggesting that their predictive performance is relatively close to CLSTM under low wind speed conditions.

The advantage of CLSTM lies in its ability to adapt to sudden changes, such as gusts. However, very low wind speed conditions are relatively stable and have less fluctuation, which may not require complex conditional modeling. As a result, CLSTM's prediction accuracy in very low wind speed conditions is not as high as other models optimized for more stable data, such as BPNN and ELM, which perform more steadily with simpler input

features. Additionally, the structure of CLSTM may introduce higher model complexity under very low wind speed conditions, increasing sensitivity to noise and thus affecting its performance.

In the Low wind speed interval (3–6 m/s), CLSTM's MSE and MAE are 0.38 and 0.48, respectively, which is the best performance among all models. The MSE for AR and ARMA is 21.14 and 21.45 and the MAE is 4.52 and 4.54, significantly higher than the other models, indicating that these two traditional models still perform poorly under low wind speed conditions. The MSE and MAE for BPNN, LSTM, and SVM are 0.42, 0.38, and 0.41 and 0.50, 0.48, respectively, showing similar performance. The MSE for EMD-SVM is 1.09 and the MAE is 0.91, while the MSE for ELM is 0.48 and the MAE is 0.55. Although EMD-SVM performs well in very low wind speed conditions, it begins to perform poorly under low wind speed conditions.

CLSTM significantly outperforms LSTM, SVM, EMD-SVM, and ELM, with p-values showing substantial differences (e.g., LSTM is 3.72×10^{-56}, SVM is 5.75×10^{-99}). This demonstrates the clear advantage of CLSTM in the Low wind speed range, providing more accurate predictions. However, with a p-value of 0.188 compared to BPNN, the prediction performance difference between CLSTM and BPNN is not statistically significant. Therefore, it can be inferred that BPNN also adapts well to wind speed fluctuations under low wind speed conditions, and its performance is similar to that of CLSTM. This result suggests that the BPNN model performs stably in low wind speed conditions and can achieve good predictions without the need for a complex structure.

Under Moderate wind speed conditions (6–9 m/s), the MSE and MAE of CLSTM are 0.49 and 0.54, respectively, performing relatively well and ranking among the top models. Compared to other models, CLSTM is more stable and accurate, especially excelling in MSE over LSTM, SVM, EMD-SVM, and ELM. The MSE of AR and ARMA are 1.98 and 1.75, with MAE of 1.15 and 1.06, respectively, showing that these traditional models still perform poorly under moderate wind speed conditions, particularly in MAE, indicating their weaker ability to capture wind speed fluctuations. Their p-values are extremely small (6.60×10^{-96} and 2.99×10^{-85}), indicating a very significant difference between these models and CLSTM. BPNN has an MSE of 0.49 and MAE of 0.55, which is similar to CLSTM, but its p-value is 1.15×10^{-42}, showing a significant difference between CLSTM and BPNN. LSTM has an MSE of 0.52 and MAE of 0.56, with a p-value of 4.62×10^{-1}, indicating no significant difference between LSTM and CLSTM under moderate wind speed conditions. This suggests that the simple structure of LSTM provides similar prediction results to CLSTM in this wind speed range. SVM has an MSE of 0.53 and MAE of 0.57, with a p-value of 4.91×10^{-4}, showing a significant difference between SVM and CLSTM, though the difference is not as pronounced as with BPNN, AR, or ARMA. EMD-SVM and ELM have MSE values of 1.20 and 1.21 and MAE values of 0.89 and 0.89, respectively. Their performance is worse, with EMD-SVM's p-value being 2.05×10^{-75} and ELM's being 6.97×10^{-48}, both of which indicate significant differences, showing that these models do not perform as well as CLSTM.

In High wind speed conditions (9–12 m/s), the CLSTM achieves an MSE of 0.51 and MAE of 0.56, performing well and ranking among the top models. The MSE for AR and ARMA are 1.23 and 1.44, respectively, with MAE values of 0.92 and 1.03, which are significantly higher than CLSTM, indicating that these traditional models perform poorly under high wind speed conditions. The p-values for AR and ARMA are 3.59×10^{-55} and 6.58×10^{-15}, respectively, showing a highly significant statistical difference between them and CLSTM. The MSE for BPNN is 0.52 and MAE is 0.56, very close to CLSTM, suggesting that BPNN performs similarly to CLSTM in high wind speed conditions. However, the p-value for BPNN is 4.44×10^{-1}, indicating no significant difference between CLSTM and BPNN, meaning that BPNN can also provide similar prediction results to CLSTM within this wind speed range. For LSTM, the MSE is 0.53 and MAE is 0.58, with a p-value of 4.33×10^{-4}, indicating a significant difference from CLSTM. This suggests that LSTM performs worse than CLSTM under high wind speed conditions. SVM has an MSE of

0.52 and MAE of 0.57, with a p-value of 3.71×10^{-47}, showing a significant difference from CLSTM, though not as pronounced as the differences between AR, ARMA, and CLSTM. Finally, EMD-SVM and ELM have MSEs of 0.74 and 0.85, and MAEs of 0.68 and 0.72, respectively, both performing worse than CLSTM. The p-values for EMD-SVM and ELM are 1.60×10^{-16} and 3.59×10^{-55}, respectively, indicating statistically significant differences from CLSTM, demonstrating that these two models perform poorly under high wind speed conditions compared to CLSTM.

Unlike models such as LSTM and BPNN, which rely on fixed structures to predict wind speed, CLSTM incorporates an adaptive mechanism that better captures the dynamic nature of wind gusts. This is particularly evident in wind speed conditions where rapid changes occur, which other models may fail to predict accurately due to their reliance on more static data patterns.

CLSTM performs excellently under most wind speed conditions, especially under moderate and high wind speed conditions. However, to gain a comprehensive understanding of its adaptability and reliability, we must further explore the residual analysis during gust and non-gust periods. This will provide more targeted insights for model optimization and practical applications. Next, we will focus on the residuals in these two scenarios to better understand the performance differences among the models under varying meteorological conditions.

4.2. Performance Comparison During Gust Events

Figure 6 presents the residual plots and distributions for these models. Each model's performance is evaluated by plotting the residuals against the sample index and showing the distribution of residuals for both non-gust and gust events. From Table 3 and Figures 4 and 6, it is evident that the CLSTM model performs exceptionally well during gust periods, as both the magnitude of the residuals and the degree of dispersion are the lowest among all models. Specifically, the mean residual for CLSTM during gust events is -0.2361, with a standard deviation of 0.3709, indicating its effective ability to capture wind speed variations and maintain stability.

However, in non-gust periods, the performance of CLSTM is relatively inferior. Although its mean residual is -0.1784 and the standard deviation is 0.7053, this does not demonstrate a significant advantage, as the differences compared to other models are not substantial. For instance, the mean residual for BPNN is -0.2035 with a standard deviation of 0.7036, LSTM has a mean residual of -0.155 and a standard deviation of 0.7066, and SVM shows a mean residual of -0.1303 with a standard deviation of 0.7312. These values indicate that the predictive capabilities of these models in non-gust conditions are relatively similar.

Notably, the non-gust mean residual for ELM is 0.0605, with a standard deviation of 0.8815, suggesting that its overall predicted values tend to be lower than the actual values. This relatively large standard deviation further reflects ELM's instability in handling non-gust data. In contrast, the non-gust mean residual for EMD-SVM is -0.8405; despite its overall predictions being biased high, its standard deviation is only 0.6197. Additionally, the predictive curve in Figure 7 illustrates that the model demonstrates good stability and consistency in predictive errors during non-gust conditions. This stability may be attributed to the EMD decomposition technique, which aids EMD-SVM in effectively extracting features from the signal, allowing it to maintain a relatively consistent predictive performance amidst varying meteorological conditions. From Table 4 and Figure 8, it can be observed that the non-gust standard deviation of EMD-SVM is lower than that of other models, particularly given its mean residual of -0.8405, indicating that despite high estimates, the model's outputs are relatively consistent, enhancing its reliability.

Table 4. Performance comparison during gust events.

Models	Non-Gust Mean Residual	Non-Gust Standard Deviation	Gust Mean Residual	Gust Standard Deviation
AR	−3.0137	2.391	−1.3771	1.9225
ARMA	−2.9926	2.4759	−1.338	2.0113
BPNN	−0.2035	0.7036	0.9696	0.409
LSTM	−0.155	0.7066	1.0853	0.4035
SVM	−0.1303	0.7312	1.0702	0.4141
EMD-SVM	−0.8405	0.6197	−0.4469	0.656
ELM	0.0605	0.8815	1.5718	0.6782
CLSTM (Ours)	−0.1784	0.7053	−0.2361	0.3709

In summary, while CLSTM exhibits excellent performance during gust periods, its performance in non-gust conditions does not significantly surpass that of other models. In non-gust states, the wind speed variations are minimal, and the conditional modeling and attention layers are primarily focused on gust events, which leads to its predictive capability not exceeding that of simpler models. Nevertheless, in practical applications, gusts typically have a greater impact on wind power generation and equipment operation. Therefore, the modeling strategy that focuses on gust events can better address the effects of gust phenomena, thus providing greater advantages in wind speed forecasting.

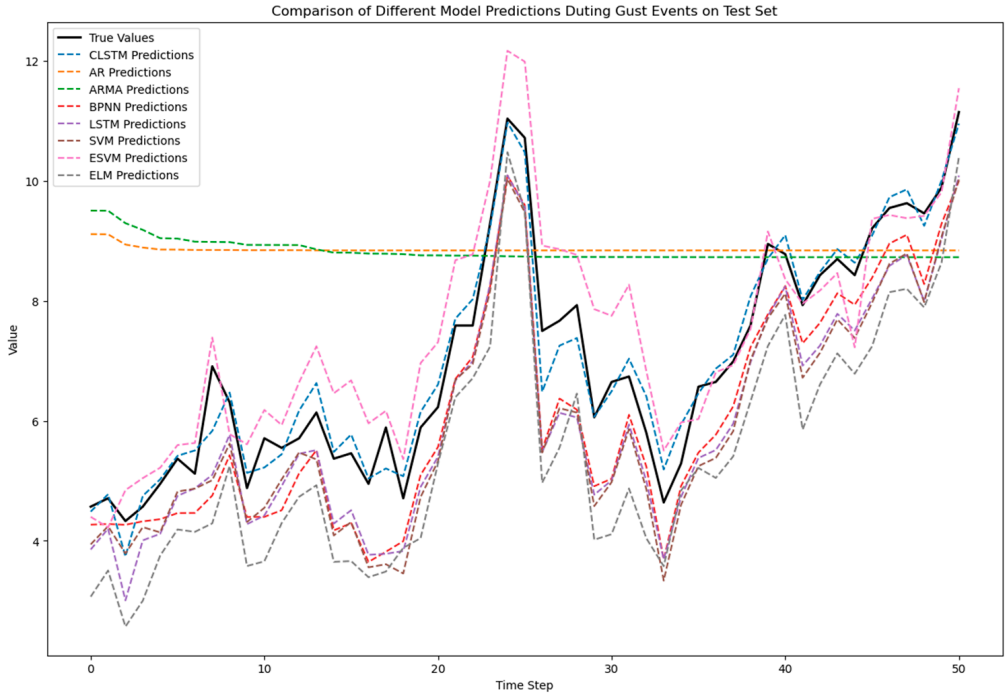


Figure 7. Predicted values of each model during gust points. It is evident that CLSTM is the closest to the true values at gust points and occasionally makes predictions that are higher than the actual values.

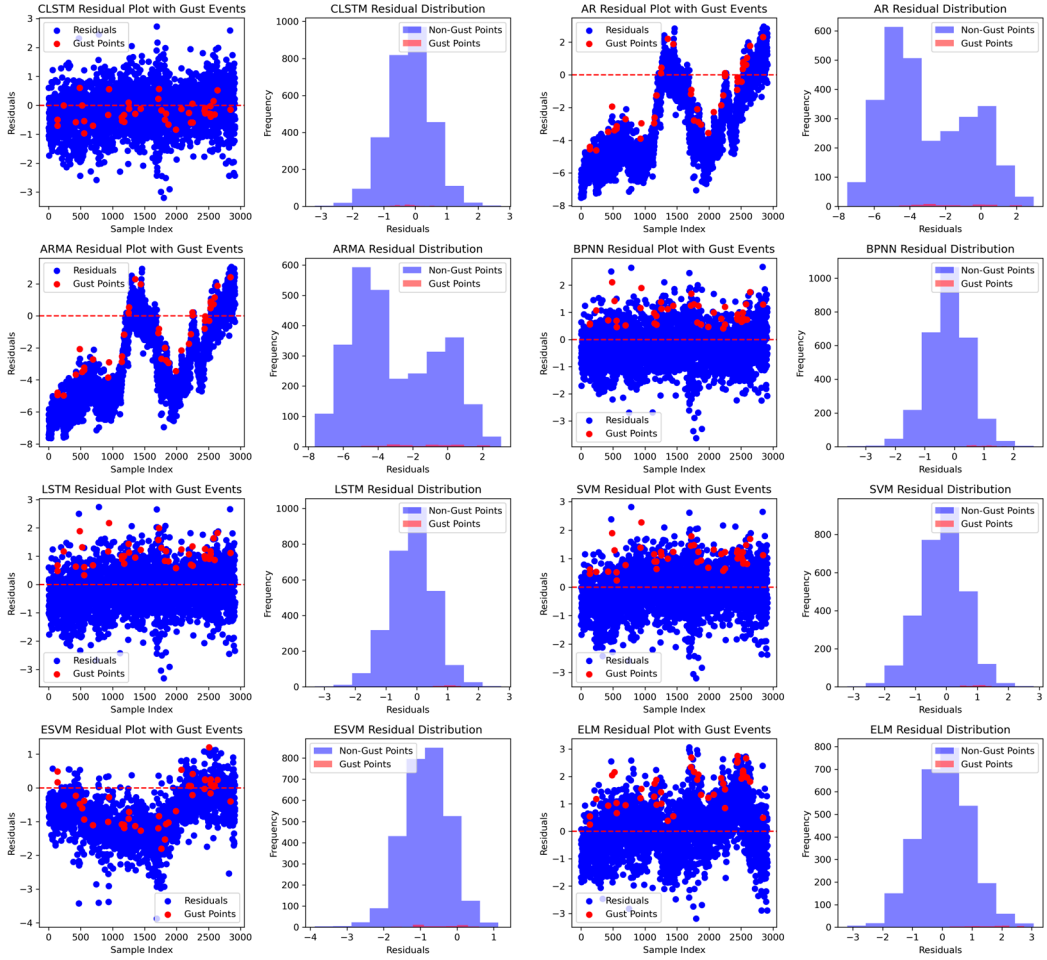


Figure 8. Residual scatter plot and distribution plot of the models. In both plots, blue represents non-gusts, red represents gusts, and the red solid line represents the zero-residual line, indicating no error between the predicted and true values. The distribution plot shows that the CLSTM model has the smallest residuals at gust points.

4.3. Reasons for Performance Variations

The variations in model performance can be attributed to several factors:

- CLSTM's architecture, utilizing conditional embeddings and long short-term memory (LSTM) units, allows it to adaptively respond to both general wind patterns and rapid gust fluctuations, maintaining high accuracy across scenarios.
- In non-gust scenarios, models such as BPNN and LSTM, while demonstrating overall good performance, may still be prone to overfitting to the training data. This overfitting can hinder their ability to generalize effectively, particularly when faced with gust conditions. Their limitations in capturing the rapid changes inherent in gusts can result in poorer performance in high-variability situations.
- The SVM model performs reasonably well in both non-gust and gust conditions, but its mean residuals indicate a tendency to slightly underpredict wind speeds during gust events. The standard deviation for SVM is also notable, reflecting its stability in predictions. However, while it demonstrates good generalization, it may lack the responsiveness to extreme fluctuations compared to more specialized models such as CLSTM.
- EMD-SVM's method may inherently emphasize sudden changes, which could explain its relatively better performance during gusts despite its overall shortcomings. This model might rely on capturing local patterns effectively, making it responsive in specific scenarios even when its broader applicability is limited.

- The ELM model shows a non-gust mean residual close to zero, indicating that its predictions are generally lower than the actual values. However, its high standard deviation suggests instability in its predictions, reflecting difficulties in adapting to the nuanced changes in wind speed during stable conditions.

Emphasizing gust prediction is critical, as gusts can impose sudden high loads on wind turbines, increasing fatigue stress, particularly on the blades and drivetrain. Without accurate forecasts, these load fluctuations may lead to equipment damage or more frequent maintenance requirements. Accurate gust predictions allow operations and maintenance teams to take proactive measures, such as adjusting control strategies to reduce the impact of load fluctuations on equipment lifespan. Additionally, gust-induced wind speed fluctuations can destabilize the turbine's power output. Incorporating gust impacts into power load forecasts helps optimize grid dispatch, ensuring smoother power delivery and improved generation efficiency. The Conditional LSTM model in particular stands out during gust events. While it may show a larger overall bias compared to other models, its greatest strength lies in its ability to make more confident predictions during gusts, which are short-lived but have significant impacts. This is in contrast to other models that may hesitate or show less accuracy in forecasting such rapid changes. By focusing on the unique characteristics of gust events and the strengths and weaknesses of each model, we can develop strategies to improve predictive accuracy and operational effectiveness in real-world applications. This holistic understanding will not only advance forecasting methodologies but also contribute to more reliable and efficient wind energy production.

5. Conclusions

This study presents an innovative approach to wind speed prediction, highlighting the critical importance of accurately forecasting gusts. Our proposed model, CLSTM, demonstrates superior performance compared to traditional forecasting methods, particularly during gust events, achieving the lowest Mean Squared Error (MSE) of 0.1404. This underscores CLSTM's ability to effectively adapt to rapid fluctuations in wind speed, a vital factor for optimizing wind turbine operations.

In contrast, models such as BPNN and LSTM, while exhibiting competitive overall performance, struggled to capture the dynamics of gust events, resulting in significant drops in performance. Conversely, EMD-SVM, despite its overall limitations, showed promising results during gust scenarios, emphasizing the importance of model selection based on specific forecasting needs.

The implications of accurate gust prediction extend beyond wind speed forecasting; they are crucial for enhancing the operational efficiency of wind turbines. By optimizing turbine settings in response to gust forecasts, operators can mitigate potential mechanical loads and reduce the risk of fatigue damage.

While this study has focused on gust prediction, it is essential to recognize that multiple factors influence the performance and efficiency of wind energy systems. Future research should aim to develop more comprehensive models that consider a wide range of variables, including atmospheric conditions, turbulence, temperature variations, and mechanical characteristics of wind turbines.

Integrating these factors into predictive models will enhance the robustness and accuracy of wind speed forecasts, enabling more informed decision-making in turbine operation and maintenance. Additionally, exploring the interactions between these variables can provide deeper insights into optimizing turbine settings and improving overall system resilience.

Furthermore, incorporating real-time data and advanced machine learning techniques can refine predictions, allowing for dynamic adjustments in response to changing conditions. By broadening the scope of investigation beyond gusts, we can significantly advance the efficiency and sustainability of wind energy systems, ultimately contributing to a more effective transition to renewable energy sources.

Author Contributions: Conceptualization, L.Z. (Liwan Zhou) and J.Z.; methodology, L.Z. (Liwan Zhou) and D.Z.; formal analysis, L.Z. (Liwan Zhou) and L.Z. (Le Zhang); investigation, L.Z. (Liwan Zhou); resources, L.Z. (Liwan Zhou); writing—original draft preparation, L.Z. (Liwan Zhou); writing—review and editing, D.Z. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangdong Basic and Applied Basic Research Foundation (2022B1515250006) and the National Natural Science Foundation of China (52177087).

Data Availability Statement: Relevant data has been shared in the paper.

Conflicts of Interest: Liwan Zhou, Di Zhang, Le Zhang, and Jizhong Zhu were employed by the South China University of Technology. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Abdoos, A.A. A new intelligent method based on combination of VMD and ELM for short term wind power forecasting. *Neurocomputing* **2016**, *203*, 111–120. [CrossRef]
2. Allen, D.J.; Tomlin, A.S.; Bale, C.S.E.; Skea, A.; Vosper, S.; Gallani, M.L. A boundary layer scaling technique for estimating near-surface wind energy using numerical weather prediction and wind map data. *Appl. Energy* **2017**, *208*, 1246–1257. [CrossRef]
3. Anvari, M.; Lohmann, G.; Wächter, M.; Milan, P.; Lorenz, E.; Heinemann, D.; Tabar, M.R.R.; Peinke, J. Short term fluctuations of wind and solar power systems. *New J. Phys.* **2016**, *18*, 063027. [CrossRef]
4. Viet, D.T.; Phuong, V.V.; Duong, M.Q.; Tran, Q.T. Models for Short-Term Wind Power Forecasting Based on Improved Artificial Neural Network Using Particle Swarm Optimization and Genetic Algorithms. *Energies* **2020**, *13*, 2873. [CrossRef]
5. Shi, J.; Qu, X.; Zeng, S. Short-Term Wind Power Generation Forecasting: Direct Versus Indirect Arima-Based Approaches. *Int. J. Green Energy* **2011**, *8*, 100–112. [CrossRef]
6. Cadenas, E.; Rivera, W. Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model. *Renew. Energy* **2010**, *35*, 2732–2738. [CrossRef]
7. Qu, X.; Kang, X.; Zhang, C.; Jiang, S.; Ma, X. Short-term prediction of wind power based on deep long short-term memory. In Proceedings of the 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Xi'an, China, 25–28 October 2016; IEEE: New York, NY, USA, 2016; pp. 1148–1152.
8. Han, L.; Jing, H.; Zhang, R.; Gao, Z. Wind power forecast based on improved Long Short Term Memory network. *Energy* **2019**, *189*, 116300. [CrossRef]
9. Harbola, S.; Coors, V. One dimensional convolutional neural network architectures for wind prediction. *Energy Convers. Manag.* **2019**, *195*, 70–75. [CrossRef]
10. Trebing, K.; Mehrkanoon, S. Wind speed prediction using multidimensional convolutional neural networks. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, Australia, 1–4 December 2020; IEEE: New York, NY, USA, 2020; pp. 713–720.
11. Yildiz, C.; Acikgoz, H.; Korkmaz, D.; Budak, U. An improved residual-based convolutional neural network for very short-term wind power forecasting. *Energy Convers. Manag.* **2021**, *228*, 113731. [CrossRef]
12. Brasseur, O. Development and Application of a Physical Approach to Estimating Wind Gusts. *Mon. Wea. Rev.* **2001**, *129*, 5–25. [CrossRef]
13. Ágústsson, H.; Ólafsson, H. Forecasting wind gusts in complex terrain. *Meteorol. Atmos. Phys.* **2009**, *103*, 173–185. [CrossRef]
14. Fawbush, E.J.; Miller, R.C. A Basis for Forecasting Peak Wind Gusts in Non-Frontal Thunderstorms. *Bull. Am. Meteorol. Soc.* **1954**, *35*, 14–19. [CrossRef]
15. Sheridan, P. Current gust forecasting techniques, developments and challenges. *Adv. Sci. Res.* **2018**, *15*, 159–172. [CrossRef]
16. Schulz, B.; Lerch, S. Machine Learning Methods for Postprocessing Ensemble Forecasts of Wind Gusts: A Systematic Comparison. *Mon. Weather Rev.* **2022**, *150*, 235–257. [CrossRef]
17. Thorarinsdottir, T.L.; Johnson, M.S. Probabilistic Wind Gust Forecasting Using Nonhomogeneous Gaussian Regression. *Mon. Weather Rev.* **2012**, *140*, 889–897. [CrossRef]
18. Zhou, K.; Cherukuru, N.; Sun, X.; Calhoun, R. Wind Gust Detection and Impact Prediction for Wind Turbines. *Remote Sens.* **2018**, *10*, 514. [CrossRef]
19. Alencar, D.B.; Affonso, C.M.; Oliveira, R.C.; Jose Filho, C.R. Hybrid approach combining SARIMA and neural networks for multi-step ahead wind speed forecasting in Brazil. *IEEE Access* **2018**, *6*, 55986–55994. [CrossRef]
20. Zaman, U.; Teimourzadeh, H.; Sangani, E.H.; Liang, X.; Chung, C.Y. Wind speed forecasting using ARMA and neural network models. In Proceedings of the 2021 IEEE Electrical Power and Energy Conference (EPEC), Virtually, 22–24 October and 30–31 October 2021; IEEE: New York, NY, USA, 2021; pp. 243–248.
21. Dong, H.; Zhu, J.; Li, S.; Chen, Z.; Wu, W.; Li, X. Dynamic Load Forecasting with Adversarial Domain Adaptation Based LSTM Neural Networks. In Proceedings of the 2023 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia), Chongqing, China, 7–9 July 2023; IEEE: New York, NY, USA, 2023; pp. 1130–1135.

22. Dong, H.; Zhu, J.; Li, S.; Luo, T.; Li, H.; Huang, Y. Ultra-Short-Term Load Forecasting Based on Convolutional-LSTM Hybrid Networks. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AK, USA, 1–3 June 2022; IEEE: New York, NY, USA, 2022; pp. 142–148.
23. Zhang, L.; Zhu, J.; Zhang, D.; Liu, Y. An incremental photovoltaic power prediction method considering concept drift and privacy protection. *Appl. Energy* **2023**, *351*, 121919. [CrossRef]
24. Wang, H.; Zhang, Y.-M.; Mao, J.-X.; Wan, H.-P. A probabilistic approach for short-term prediction of wind gust speed using ensemble learning. *J. Wind Eng. Ind. Aerodyn.* **2020**, *202*, 104198. [CrossRef]
25. Zhu, J.; Zhang, L.; Zhang, D.; Chen, Y. Probabilistic Wind Power Prediction Using Incremental Bayesian Stochastic Configuration Network Under Concept Drift Environment. *IEEE Trans. Ind. Appl.* **2024**, 1–11. [CrossRef]
26. Guo, Z.; Zhao, W.; Lu, H.; Wang, J. Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model. *Renew. Energy* **2012**, *37*, 241–249. [CrossRef]
27. Jiang, Z.; Che, J.; Wang, L. Ultra-short-term wind speed forecasting based on EMD-VAR model and spatial correlation. *Energy Convers. Manag.* **2021**, *250*, 114919. [CrossRef]
28. Wang, J.; Zhang, W.; Li, Y.; Wang, J.; Dang, Z. Forecasting wind speed using empirical mode decomposition and Elman neural network. *Appl. Soft Comput.* **2014**, *23*, 452–459. [CrossRef]
29. Zhang, C.; Wei, H.; Zhao, J.; Liu, T.; Zhu, T.; Zhang, K. Short-term wind speed forecasting using empirical mode decomposition and feature selection. *Renew. Energy* **2016**, *96*, 727–737. [CrossRef]
30. Ren, Y.; Suganthan, P.N.; Srikanth, N. A novel empirical mode decomposition with support vector regression for wind speed forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *27*, 1793–1798. [CrossRef]
31. An, X.; Jiang, D.; Zhao, M.; Liu, C. Short-term prediction of wind power using EMD and chaotic theory. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 1036–1042. [CrossRef]
32. Saroha, S.; Aggarwal, S.K. Wind power forecasting using wavelet transforms and neural networks with tapped delay. *CSEE J. Power Energy Syst.* **2018**, *4*, 197–209. [CrossRef]
33. Singh, S.N.; Mohapatra, A. Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renew. Energy* **2019**, *136*, 758–768.
34. Li, L.-L.; Chang, Y.-B.; Tseng, M.-L.; Liu, J.-Q.; Lim, M.K. Wind power prediction using a novel model on wavelet decomposition-support vector machines-improved atomic search algorithm. *J. Clean. Prod.* **2020**, *270*, 121817. [CrossRef]
35. Han, L.; Zhang, R.; Wang, X.; Bao, A.; Jing, H. Multi-step wind power forecast based on VMD-LSTM. *IET Renew. Power Gener.* **2019**, *13*, 1690–1700. [CrossRef]
36. Sun, Z.; Zhao, S.; Zhang, J. Short-term wind power forecasting on multiple scales using VMD decomposition, K-means clustering and LSTM principal computing. *IEEE Access* **2019**, *7*, 166917–166929. [CrossRef]
37. Sun, Z.; Zhao, M. Short-term wind power forecasting based on VMD decomposition, ConvLSTM networks and error analysis. *IEEE Access* **2020**, *8*, 134422–134434. [CrossRef]
38. Zhang, Y.; Pan, G.; Chen, B.; Han, J.; Zhao, Y.; Zhang, C. Short-term wind speed prediction model based on GA-ANN improved by VMD. *Renew. Energy* **2020**, *156*, 1373–1388. [CrossRef]
39. Chen, Y.; Dong, Z.; Wang, Y.; Su, J.; Han, Z.; Zhou, D.; Zhang, K.; Zhao, Y.; Bao, Y. Short-term wind speed predicting framework based on EEMD-GA-LSTM method under large scaled wind history. *Energy Convers. Manag.* **2021**, *227*, 113559. [CrossRef]
40. He, Y.; Wang, Y. Short-term wind power prediction based on EEMD–LASSO–QRNN model. *Appl. Soft Comput.* **2021**, *105*, 107288. [CrossRef]
41. Hu, J.; Wang, J.; Zeng, G. A hybrid forecasting approach applied to wind speed time series. *Renew. Energy* **2013**, *60*, 185–194. [CrossRef]
42. Wang, S.; Zhang, N.; Wu, L.; Wang, Y. Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method. *Renew. Energy* **2016**, *94*, 629–636. [CrossRef]
43. Hanifi, S.; Zare-Behtash, H.; Cammarano, A.; Lotfian, S. Offshore wind power forecasting based on WPD and optimised deep learning methods. *Renew. Energy* **2023**, *218*, 119241. [CrossRef]
44. Kumar, B.; Yadav, N.; Sunil. A novel hybrid algorithm based on empirical fourier decomposition and deep learning for wind speed forecasting. *Energy Convers. Manag.* **2024**, *300*, 117891. [CrossRef]
45. Chen, X.; Wang, Y.; Zhang, H.; Wang, J. A novel hybrid forecasting model with feature selection and deep learning for wind speed research. *J. Forecast.* **2024**, *43*, 1682–1705. [CrossRef]
46. Barjasteh, A.; Ghafouri, S.H.; Hashemi, M. A hybrid model based on discrete wavelet transform (DWT) and bidirectional recurrent neural networks for wind speed prediction. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107340. [CrossRef]
47. Xiao, Y.; Wu, S.; He, C.; Hu, Y.; Yi, M. An effective hybrid wind power forecasting model based on "decomposition-reconstruction-ensemble" strategy and wind resource matching. *Sustain. Energy Grids Netw.* **2024**, *38*, 101293. [CrossRef]
48. Sun, Y.; Zhang, S. A multiscale hybrid wind power prediction model based on least squares support vector regression–regularized extreme learning machine–multi-head attention–bidirectional gated recurrent unit and data decomposition. *Energies* **2024**, *17*, 2923. [CrossRef]
49. Parri, S.; Teeparthi, K. SVMD-TF-QS: An efficient and novel hybrid methodology for the wind speed prediction. *Expert Syst. Appl.* **2024**, *249*, 123516. [CrossRef]

50. Sarangi, S.; Dash, P.K.; Bisoi, R. Probabilistic prediction of wind speed using an integrated deep belief network optimized by a hybrid multi-objective particle swarm algorithm. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107034. [CrossRef]
51. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
52. Wen, T.-H.; Gasic, M.; Mrksic, N.; Su, P.-H.; Vandyke, D.; Young, S. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. *arXiv* **2015**, arXiv:1508.01745.
53. Ribeiro, R.P.; Moniz, N. Imbalanced regression and extreme value prediction. *Mach Learn* **2020**, *109*, 1803–1835. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.