processes

MDPI

Article

A Hybrid DSCNN-BiLSTM Model for Accurate Wind Turbine Temperature Prediction

Xinping Li [1,2,3], Zhihui Qi [1], Zhengrong Zhou [1] and Jun Hu [1,2,3,*]

[1] School of Civil Engineering and Architecture, Wuhan University of Technology, Wuhan 430062, China; xinpingli@whut.edu.cn (X.L.); zhihuiqi0308@163.com (Z.Q.); guiyangfx@sina.cn (Z.Z.)
[2] China Sanya Science and Education Innovation Park, Wuhan University of Technology, Sanya 572025, China
[3] Hainan Engineering Research Center for Construction and Protection of Islands and Reefs, Sanya 572024, China
* Correspondence: junhu22@whut.edu.cn

Abstract: The temperature variations in wind turbine motors and gearboxes are closely related to their health status, making accurate temperature prediction essential for operational monitoring and early fault detection. However, conventional deep learning-based temperature prediction methods, such as recurrent neural networks (RNN) and convolutional neural networks (CNN) and their hybrid models, often face challenges in capturing both local feature dependencies and long-term temporal patterns in complex, nonlinear temperature fluctuations. To address these limitations, this paper proposes a hybrid model based on depthwise separable convolutional neural networks (DSCNNs) and bidirectional long short-term memory (BiLSTM) networks. The DSCNN module enhances feature extraction from temperature signals, while the BiLSTM module captures long-term dependencies, improving prediction accuracy and robustness. Experimental validation using temperature data from a wind farm in Shaanxi, China, demonstrates that the proposed model outperforms existing deep learning approaches, achieving superior prediction accuracy, better adaptability to temperature fluctuations, and greater robustness in handling complex nonlinear dynamics. Furthermore, the proposed model provides an effective solution for early fault detection in wind turbines, including both mechanical faults (e.g., gearbox wear, bearing overheating) and electrical faults (e.g., winding short circuits, overload conditions), contributing to more reliable condition monitoring in industrial applications.

Keywords: temperature prediction; wind turbine; deep separable convolutional neural network; bidirectional long short-term memory network

check for updates

Academic Editor: Krzysztof Rogowski

Received: 6 March 2025
Revised: 2 April 2025
Accepted: 8 April 2025
Published: 10 April 2025

Citation: Li, X.; Qi, Z.; Zhou, Z.; Hu, J. A Hybrid DSCNN-BiLSTM Model for Accurate Wind Turbine Temperature Prediction. *Processes* **2025**, *13*, 1143. https://doi.org/10.3390/pr13041143

Copyright: © 2025 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

The increasing global demand for renewable energy, such as wind power, as a clean and efficient energy source, is playing an increasingly significant role in the global energy transition [1–3]. As the core equipment responsible for converting wind energy into electrical energy, the stability and efficient operation of wind turbines directly affect the overall performance of wind power generation [4,5]. Over the long-term operation of wind turbines, the health of mechanical components, particularly the gearbox and generator, often determines the turbine's service life and maintenance costs [6,7]. Therefore, efficient monitoring and prediction of the operational status of these critical components, especially temperature variations, have become essential research topics in modern wind power generation [8]. Temperature fluctuations serve as a crucial indicator of wind turbine operational status, reflecting the condition of mechanical components, particularly under

long-term operation. Abnormal temperature fluctuations or sudden changes often signal potential failures [9]. For example, excessive gearbox temperatures can lead to the degradation of the lubricating oil, increasing wear on mechanical parts and potentially causing system failures [10,11]. Consequently, real-time monitoring of wind turbine temperature and accurate forecasting of future temperature trends are vital for ensuring smooth turbine operation and minimizing downtime caused by failures [12].

In recent years, researchers have proposed various traditional mathematical modeling techniques for temperature prediction, which have yielded notable results under specific conditions and have laid a solid foundation for the application of temperature forecasting [13]. These traditional methods primarily rely on models based on physical principles or statistical techniques, which can address temperature variation to some extent [14]. For instance, Gu et al. [15] proposed a method that combines the HTcT model with multi-output least squares support vector regression (MOLSSVR). By replacing traditional temperature factors with temperature field features extracted through clustering, they overcame the issue of multicollinearity and introduced a gray wolf optimization (GWO) algorithm to optimize hyperparameters, thus improving prediction efficiency and accuracy. Chen et al. [16] introduced a novel hybrid model, S-GM-ARIMA, which integrates the GM model with the ARIMA model, optimizing it through a linear combination weight calculation method. Their research compared different weight calculation techniques and ultimately adopted the standard deviation method to compute the weights, enhancing prediction accuracy. Das et al. [17] proposed a model-free temperature prediction approach. Other studies have focused on developing one-step prediction models and prediction intervals to improve temperature forecasting accuracy, particularly for long, local stationary time series data, outperforming the widely used RAMPFIT algorithm. While these methods have made significant progress in certain areas, traditional mathematical models generally rely on prior physical knowledge or assumptions. When confronted with complex and nonlinear temperature variations, these models struggle to fully and accurately capture the underlying patterns in the data [18]. As a result, traditional methods still face substantial challenges in handling the complexities encountered in practical applications, limiting their broader adoption in industrial settings.

Compared to traditional analytical methods, machine learning techniques have shown considerable advantages in handling complex, nonlinear data, as they can automatically identify and extract key features [19]. Among various machine learning models, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory networks (LSTMs) have been widely applied across numerous tasks [20]. In particular, LSTM is particularly well-suited for time series prediction tasks. By incorporating memory units, LSTM can effectively capture long-term dependencies, making it especially suitable for processing dynamic sequential data with such dependencies. This significantly enhances the stability and accuracy of the model in long-term forecasting [21]. For example, Wang et al. [22] proposed a short-term water quality prediction model that combines variational mode decomposition (VMD) with an improved grasshopper optimization algorithm (IGOA). This model optimized the LSTM network, significantly improving short-term prediction accuracy. Shahid et al. [23] developed a short-term wind power prediction model based on LSTM, incorporating a wavelet kernel function to capture the dynamic characteristics of wind power data. By combining LSTM with wavelet transformers, the model successfully addressed the nonlinear mapping problem, thereby enhancing prediction accuracy. Huang et al. [24] designed an oil and gas production prediction model based on LSTM, optimized through particle swarm optimization (PSO) to adjust LSTM configurations. This model effectively captured the dependencies in time series data related to oil and gas production while integrating production constraints. It demonstrated high

prediction accuracy in multiple real-world applications, particularly in complex oil and gas production systems, outperforming traditional prediction methods and showcasing strong adaptability to various production environments.

Although the LSTM model excels at handling time series data, it can only capture unidirectional temporal dependencies, which may limit its ability to recognize certain complex sequential patterns. To overcome this limitation, the bidirectional long short-term memory network (BiLSTM) was introduced. BiLSTM learns both forward and backward information from a time series, enabling it to capture richer contextual information. This enhances its capacity to model complex temporal features, ultimately improving prediction accuracy. For instance, Cui et al. [25] proposed a method that combines singular spectrum analysis (SSA) with BiLSTM for accurately predicting missing values in MODIS land surface temperature (LST) data. The validation results showed that this method maintained high prediction accuracy even under high missing rates. Zhang et al. [26] introduced a monthly average temperature prediction model based on CEEMDAN-BO-BiLSTM, which was applied to temperature forecasting in Jinan city. The results demonstrated that this model significantly outperformed other models in terms of prediction accuracy and adaptability, offering an effective solution for temperature forecasting. Similarly, Jiang et al. [27] proposed a method combining an elite-preserving genetic algorithm (EGA) with BiLSTM for temperature prediction in battery energy storage power stations (BESP). Validated with real-world data, this method effectively improved temperature prediction accuracy, providing reliable forecasts for the safe operation of BESP.

Although the BiLSTM model has shown success in various fields, its limitations become evident when dealing with complex temperature data. Temperature data often exhibit characteristics such as nonlinearity and periodicity, and a single model may not fully exploit the complementary features across different types of data. As a result, hybrid models have become increasingly popular, combining the strengths of multiple models to enhance prediction accuracy [28]. For example, Tabrizchi et al. [29] proposed an efficient temperature prediction model for data centers by combining CNNs with multi-layer BiLSTM, significantly improving prediction accuracy and reducing errors. Ji et al. [30] introduced a novel hybrid prediction model that combines CNNs, BiLSTM, and squeeze-and-excitation (SE) networks, aiming to leverage the strengths of various deep learning models to enhance furnace temperature prediction accuracy. Similarly, Jiang et al. [31] proposed a deep learning model combining LSTM, an encoder–decoder structure, and an attention mechanism for short-term indoor temperature forecasting. This model outperformed traditional LSTM and GRU models, demonstrating higher prediction accuracy and greater stability.

In addition to traditional CNNs, depthwise separable convolutional neural networks (DSCNNs) have gained increasing attention for their ability to efficiently extract spatial features while significantly reducing computational complexity. In recent years, DSCNN has also been explored for time series signal processing. For example, Yu et al. [32] combined GRUDMU with DSCNNs and deployed the model on edge devices, enhancing real-time fault diagnosis performance in edge computing scenarios. Xie et al. [33] combined the 1D-DSCNN with Global Max Pooling (GMP) to create the 1D-DSCNN-GMP model, which was optimized using TensorRT and deployed on edge devices, achieving improved fault diagnosis with reduced model size and faster inference time. Wang et al. [34] combined Principal Component Analysis (PCA) with Gramian Angular Field (GAF) methods and DSCNN for operation state recognition of hydroelectric generating units, achieving high accuracy in fault diagnosis.

These studies highlight the growing potential of DSCNNs in diverse applications, demonstrating their effectiveness in improving both the efficiency and accuracy of fault diagnosis and state recognition tasks. Building on these advancements, this paper introduces

an innovative hybrid model that combines DSCNNs with BiLSTM for accurate wind turbine gearbox temperature prediction. Like previous applications, the DSCNN in this model significantly reduces the parameter count by utilizing depthwise separable convolutions, improving computational efficiency while preserving the ability to extract robust spatial features. This synergy of DSCNNs and BiLSTM enables the model to effectively handle large-scale, high-dimensional temperature data. The BiLSTM component, in turn, captures bidirectional dependencies in time series data, further strengthening the model's ability to capture periodic and long-term dependencies. By leveraging the strengths of both models, the proposed DSCNN-BiLSTM hybrid model integrates spatial and temporal learning, enabling precise wind turbine temperature predictions. To evaluate the performance of the proposed DSCNN-BiLSTM model, this study conducts temperature prediction experiments using two real-world datasets from a wind farm in Shaanxi. The experimental results show that the DSCNN-BiLSTM model significantly outperforms traditional methods in terms of prediction accuracy and generalization, underscoring its feasibility and effectiveness for real-world engineering applications. The main contributions of this paper are as follows: (1) A novel DSCNN-BiLSTM hybrid model is proposed, enhancing prediction accuracy by combining spatial feature extraction with time series modeling. (2) An efficient model architecture is designed to handle large-scale and complex temperature data, customized to the characteristics of wind turbine temperature data. (3) Experimental validation demonstrates the model's effectiveness in predicting wind turbine motor and gearbox temperatures, highlighting its potential for practical applications.

The structure of this article is organized as follows: Section 2 presents the basic theory of the proposed model; Section 3 details the implementation of the proposed method; Section 4 conducts several experiments to evaluate the performance of the proposed model; and, finally, Section 5 provides the conclusion.

2. Related Work

2.1. Depthwise Separable Convolution

Depthwise separable convolutional neural networks are an efficient variant of traditional CNNs that significantly reduce computational complexity while preserving feature extraction capabilities. Unlike conventional CNNs, which apply a single convolution operation to both spatial and channel dimensions simultaneously [35], DSCNN decomposes this process into two separate steps: depthwise convolution and pointwise convolution. The depthwise convolution operates independently on each input channel, capturing spatial features, while the pointwise convolution (1×1 convolution) fuses information across channels. This structure reduces the number of parameters and floating-point operations, making DSCNNs particularly suitable for real-time applications and large-scale data processing. First, a brief introduction to the basic principles of traditional convolutional neural networks. The traditional convolution operation, as shown in Figure 1a, combines spatial convolution and channel convolution. The input variable $\boldsymbol{X}$ and convolution kernel parameters $\boldsymbol{W}_i$ can both be represented in matrix form, where $*$ denotes the convolution operation, $\boldsymbol{b}_i$ is the bias term, σ is the activation function, and $\boldsymbol{Y}_i$ is output. The processing procedure of the traditional convolutional network can be described by the following equations:

$$\boldsymbol{Y}_i = \sigma(\boldsymbol{X} * \boldsymbol{W}_i + \boldsymbol{b}_i) \tag{1}$$

Assume the input signal sequence is $\boldsymbol{X}_t$, where $i = 1, 2, \ldots, N$, and the convolution kernel $\boldsymbol{W}_k$ is indexed by $i = 1, 2, \ldots, M$. The output $\boldsymbol{Y}_t$ of the convolution operation is as follows:

$$\boldsymbol{Y}_t = \sum_{k=1}^{M} \boldsymbol{W}_k * \boldsymbol{X}_{i+k} \tag{2}$$

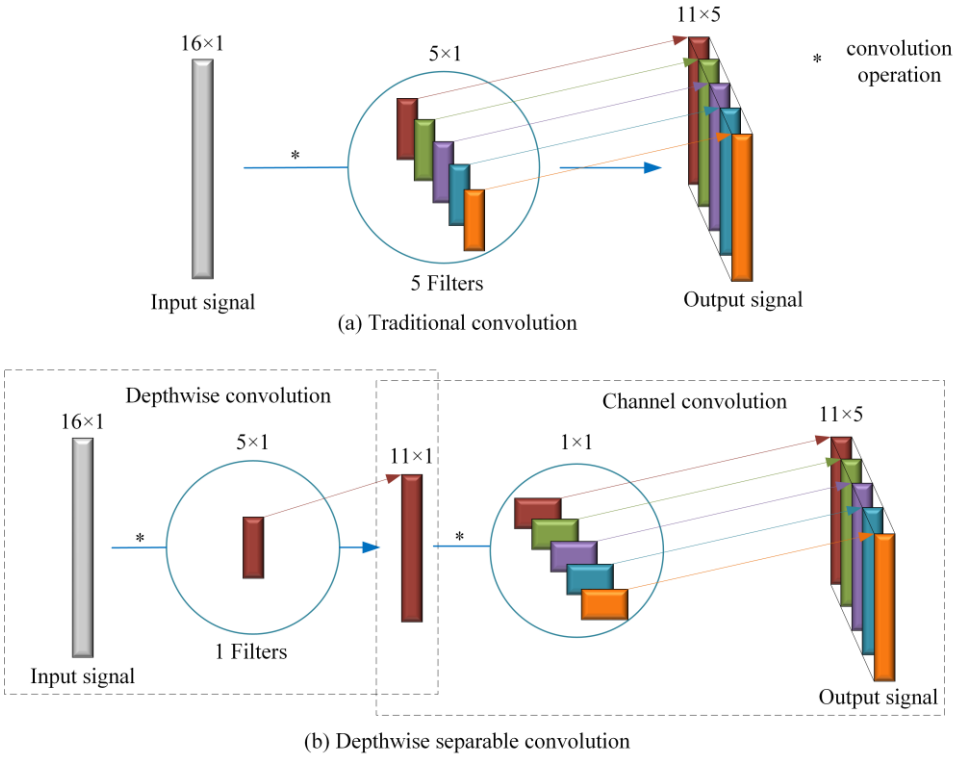


Figure 1. A comparison of the traditional convolution and depthwise separable convolution.

In a traditional convolutional layer, each neuron is connected to the neurons in the local region of the previous layer, forming a local connection network. To perform nonlinear feature mapping, convolutional layers are typically equipped with an activation function $f(x)$. As shown in Figure 1b, the use of depthwise separable convolution significantly reduces the required parameters compared to traditional convolution. The key to depthwise separable convolution lies in splitting the traditional convolution into depthwise convolution and pointwise convolution, first performing depthwise convolution and then pointwise convolution, which achieves separation between channels. To more intuitively compare the number of parameters between depthwise separable convolution and traditional convolution, the mathematical calculation is as follows:

The computational complexity of traditional convolution operations is the following:

$$\boldsymbol{D}_K \times M \times N \times \boldsymbol{D}_F \tag{3}$$

The computational complexity of depthwise separable convolution operations is the following:

$$\boldsymbol{D}_K \times M \times \boldsymbol{D}_F + M \times N \times \boldsymbol{D}_F \tag{4}$$

where $\boldsymbol{D}_K$ is the size of the convolution kernel, M is the number of input channels, N is the number of convolution kernels, and $\boldsymbol{D}_F$ is the size of the input data. The ratio of the computational complexity of depthwise separable convolution to traditional convolution is as follows:

$$\frac{\boldsymbol{D}_K \times M \times \boldsymbol{D}_F + M \times N \times \boldsymbol{D}_F}{\boldsymbol{D}_K \times M \times N \times \boldsymbol{D}_F} = \frac{1}{N} + \frac{1}{\boldsymbol{D}_K} < 1 \tag{5}$$

As shown from Equation (3) to Equation (5), depthwise separable convolution significantly reduces the computational complexity and improves the system's efficiency. If the kernel size is three and the number of kernels is five, according to Equation (3), depthwise separable convolution can reduce the number of parameters to approximately 50% of

the original parameters. Therefore, the depthwise separable convolution model shows a significant improvement in speed.

2.2. BiLSTM

To handle sequential data, recurrent neural networks endow the network with memory, allowing it to retain information from the previous layer. By combining the output from the previous layer with the current layer's output, feature vectors are formed, which preserve more valuable features and, in turn, improve prediction accuracy. However, issues such as vanishing and exploding gradients significantly impact prediction performance, limiting the effectiveness in practical applications. LSTM, a special type of RNN, effectively addresses long-term dependencies in data and avoids the common vanishing and exploding gradient problems found in standard RNN models [36]. A more detailed implementation and application of these methods in our study are discussed in Section 3. LSTM introduces three novel gating mechanisms in its network architecture, including the input gate, forget gate, and output gate, designed to protect and regulate the flow of information within the time series data. Through these gating structures, LSTM can adaptively determine which information is useful in the short and long term, making it particularly suitable for handling long time series data. The structure of an LSTM neural network is shown in Figure 2.

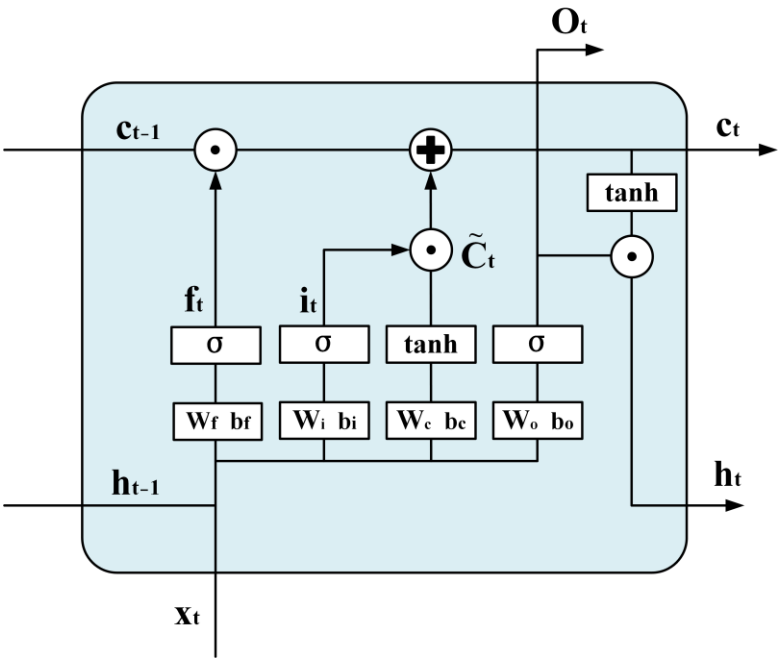


Figure 2. The structure of the LSTM neural network.

As shown in Figure 2, the LSTM structure includes three control gates: the forget gate, input gate, and output gate. The forget gate calculates the degree of forgetfulness by processing the previous time step's hidden state $\boldsymbol{h}_{t-1}$ and the current time step's input $\boldsymbol{x}_t$ through a sigmoid unit. The forget gate output is denoted as $\boldsymbol{f}_t$. The calculation formula is as follows:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f\right) \tag{6}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function, $\boldsymbol{W}_f$ is the weight matrix for the forget gate, and $\boldsymbol{b}_f$ is the bias term for the forget gate. The input gate is denoted as $\boldsymbol{i}_t$. The updated formulas for the input gate and candidate state are as follows:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{7}$$

where W_i is the weight matrix for the input gate, and b_i is the bias term for the input gate. The candidate cell state is denoted as $\widetilde{C}_t$, is calculated as follows:

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{8}$$

where W_C is the weight matrix for the candidate cell state, and b_C is the bias term for the candidate cell state. Then, the cell state C_t is updated based on the outputs of the forget gate and input gate. The previous cell state C_{t-1} represents the memory of the LSTM from the previous time step, which is selectively retained or forgotten based on the forget gate output f_t, and the update formula is as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{9}$$

Finally, the output hidden state h_t of the LSTM is determined based on the cell state C_t, the output gate, the previous hidden state h_{t-1}, and the current input x_t. The output gate is denoted as o_t, which determines how much information from the cell state C_t is passed to the next hidden state. The output formula is as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{10}$$

$$h_t = o_t \odot tanh(C_t) \tag{11}$$

The symbol $\odot$ represents the element-wise multiplication. LSTM can effectively handle the long-range dependency problem in time series data, but it can only capture unidirectional dependencies in the data. To further improve the model's performance, BiLSTM was introduced. By combining a forward LSTM and a backward LSTM, BiLSTM can better capture both forward and backward dependencies in the data [37]. Specifically, the BiLSTM model consists of two LSTMs: one processes the input sequence from left to right, while the other processes it from right to left. After processing, the outputs from the two LSTMs are concatenated and passed to the next layer of the network. The network structure of BiLSTM is shown in Figure 3. By incorporating the backward LSTM, BiLSTM not only considers information from the past but also utilizes information from future time steps [38]. This can significantly improve the model's accuracy in many practical applications, especially when the data exhibits strong temporal dependencies or periodicity. Like LSTM, BiLSTM also uses three gating structures, which help enhance the model's ability to learn and represent time series data.

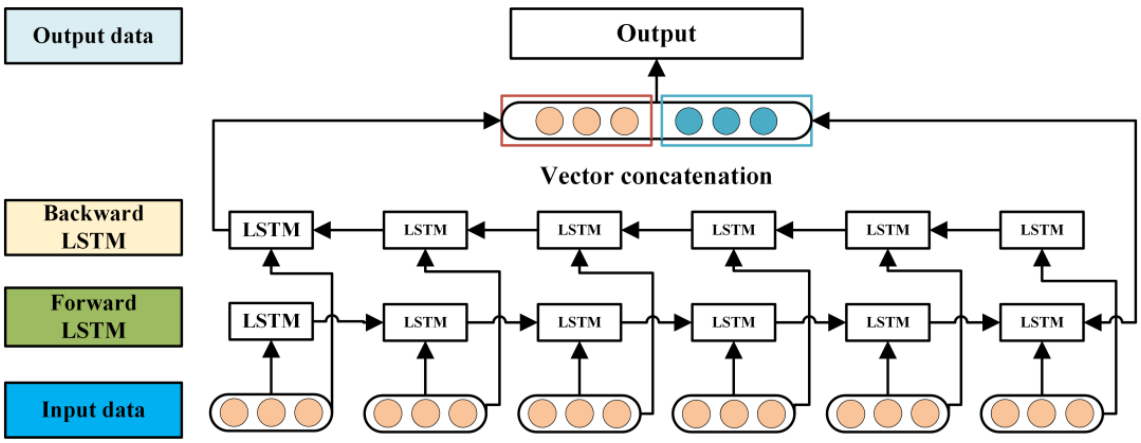


Figure 3. The structure of the BiLSTM.

3. Methodology

3.1. The Proposed DSCNN-BiLSTM Model

In this paper, we propose a hybrid model that combines depthwise separable convolutional neural networks (DSCNNs) and bidirectional long short-term memory (BiLSTM) for temperature prediction. The model retains the advantages of both DSCNNs and BiLSTM, enabling efficient extraction of spatial and temporal features while reducing the number of untrained parameters, thus accelerating model training and enabling it to reach optimal performance more quickly and efficiently. In this hybrid model, the DSCNN is used for spatial feature extraction. As an enhanced version of the traditional convolutional neural network, DSCNNs effectively capture spatial features from the input data. Compared to conventional convolutional networks, DSCNNs significantly reduce the number of required convolutional kernels, thereby decreasing the number of parameters during training. This not only makes the training process more efficient but also improves the model's performance when handling high-dimensional data. BiLSTM is employed to extract temporal features. By integrating both forward and backward network structures, BiLSTM allows the model to capture information from both past and future data. This capability enables BiLSTM to effectively model long-term dependencies and periodic patterns in time series data, thereby enhancing the accuracy of temperature predictions. The hybrid model combines the strengths of both DSCNNs and BiLSTM, significantly improving performance in temperature prediction tasks. The workflow of the model is shown in Figure 4.

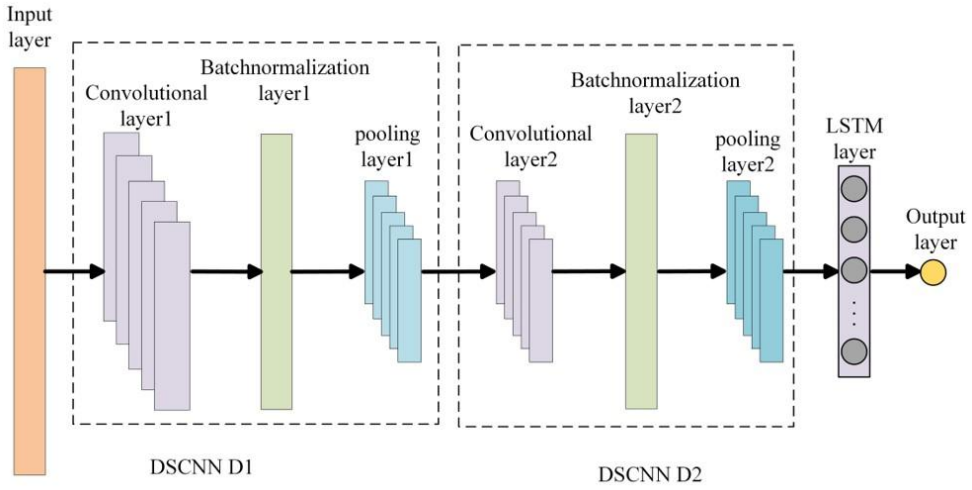


Figure 4. The workflow of the DSCNN-BiLSTM.

3.2. Data Preprocessing

In this study, the preprocessing of temperature data involves two key steps: data normalization and slicing. Data preprocessing is crucial to ensure the model can effectively learn from the data features, avoid potential issues during training, and improve the model's overall performance. The following provides a detailed description of these two preprocessing techniques. Normalization of the temperature data is an essential step. Since output values from different sensors can vary significantly and the units of different features may not be consistent, unnormalized data could lead to certain features dominating the model training process, potentially affecting the model's convergence and stability. To address this, the min-max normalization method was employed. Min-max normalization compresses the data into a specified range, typically [0, 1], ensuring that all features are scaled uniformly during training. This process is implemented using the following formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{12}$$

In this equation, X represents the original data, X_{min} and X_{max} are the minimum and maximum values of the data, and X_{norm} is the normalized data. By employing this method, we can effectively mitigate issues arising from differences in data scales or extreme values, ensuring that each feature contributes equally during the model training process. Additionally, normalization enhances the efficiency of the gradient descent optimization algorithm, reduces the model's convergence time, and prevents numerical instability during training. However, if the input data during inference falls outside the [0, 1] range, the model may encounter problems such as erroneous predictions or instability. To address this, we implement a clipping strategy, where values below 0 are set to 0, and values above 1 are set to 1. This ensures that the input data remains within the expected range, preserving the model's stability and performance.

After data normalization, to effectively prepare the dataset for model training, this study further divides the data using a time-window-based sampling strategy. As illustrated in Figure 5, we applied a sliding window approach to transform the original time series data into training samples and corresponding labels. This strategy splits the original time series into multiple fixed-length samples, with each sample corresponding to a prediction label at a specific time step. The steps are as follows:

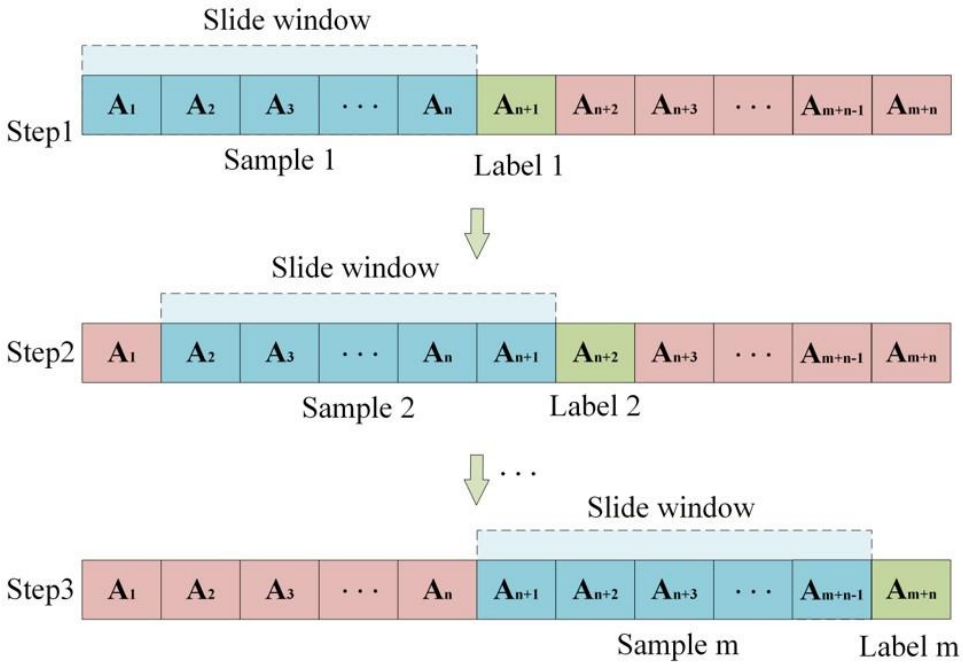


Figure 5. Illustration of the time-window-based sample generation strategy.

Step 1: Initial Time Window Selection. The first n consecutive data points from the dataset are selected as the input sample, with the data point immediately following these points serving as the label. These data points represent the historical input of the model, while the label corresponds to the prediction target.

Step 2: Sliding Window Update. The first data point in the dataset is removed, and the window of the training sample slides forward. The new training sample consists of the next n consecutive data points, with the label being the data point immediately following these points. Each sliding window generates a new sample-label pair.

Step 3: Sample Generation Repetition. Steps 1 and 2 are repeated until there are insufficient remaining data points to form a new training sample. The sliding window continues to generate new sample-label pairs until all available data have been processed.

3.3. Model Configuration and Training Parameters

In this study, the proposed DSCNN-BiLSTM model leverages carefully chosen training parameters and strategies to ensure efficient temperature prediction. The overall network architecture and parameter configuration are summarized in Table 1. To optimize performance, we explored various network configurations, including different depths for the DSCNN layers, the number of BiLSTM units, and the design of the fully connected layers. The final architecture was selected based on its validation performance and computational efficiency. Hyperparameter tuning was used to determine the optimal dropout rate, achieving a balance between generalization and training stability. No significant overfitting was observed during the validation phase, as further elaborated in Section 4.

Table 1. Parameters of the proposed DSCNN-BiLSTM.

Definition of Layers	Parameters of Layers
Input layer	Size: (1, n)
DSCNN D1	Channel number: 32, kernel size: 3, padding: 1
Batch Normalization layer	-
Pooling layer	Kernel size: 2, stride: 2
DSCNN D2	Channel number: 64, kernel size: 3, padding: 1
Pooling layer	Kernel size: 2, stride: 2
Batch Normalization layer	-
BiLSTM	hidden units: 64
Activation layer	Activation function: ReLu
Dropout layer	Dropout number: 0.3
Output layer	-

Remarks: where n represents the number of input features, corresponding to the number of sensor readings or time series points considered for each sample.

During the training process, the Adam optimizer was used in this study. Due to its adaptive learning rate feature, the Adam optimizer automatically adjusts the step size based on the different updating requirements of the parameters, thus accelerating the convergence process of the model. Specifically, the learning rate was set to 0.001, the batch size to 32, and the number of training epochs was set to 200 to ensure that the model could effectively learn the deeper features of the data through sufficient iterations. The DSCNN-BiLSTM model was implemented using Python 3.9 and PyTorch 1.10, which is a popular deep learning framework. All experiments were carried out on a workstation running Windows 11, equipped with an Intel i5-12400F CPU and a GTX 3060 Ti GPU.

In terms of the loss function, this study adopted mean squared error (MSE) as the loss function for the temperature prediction task. MSE is a widely used loss function for regression problems, as it quantifies the difference between the model's predictions and the actual values, making it suitable for continuous numerical data. The loss function L, which represents the average squared difference between the predicted values and the actual values, is defined as follows:

$$\mathrm{L} = \frac{1}{n}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{13}$$

Here, $\hat{y}_i$ represents the predicted output of the i-th sample by the model, y_i represents the true value of the i-th sample, and N is the total number of samples. By minimizing the MSE, the model can better approximate the true temperature variations, thereby improving the prediction accuracy.

4. Experiment and Discussion

4.1. Data Collection

In this section, operational data from the motor and gearbox of a wind turbine at a wind farm in Shaanxi, China, were collected for experimental validation to assess the performance of the proposed model in real-world applications. To ensure the reliability and representativeness of the experimental results, temperature data were recorded at 60 s intervals, with each data collection consisting of 3000 time points. Figures 6 and 7 display the temperature data for the motor and gearbox, respectively. The temperature was measured using industrial-grade thermocouples installed at the generator bearings and gearbox housing, ensuring accurate and consistent readings. The collected data were directly obtained from the wind farm's Supervisory Control and Data Acquisition (SCADA) system, which continuously monitors the turbine's operational status.

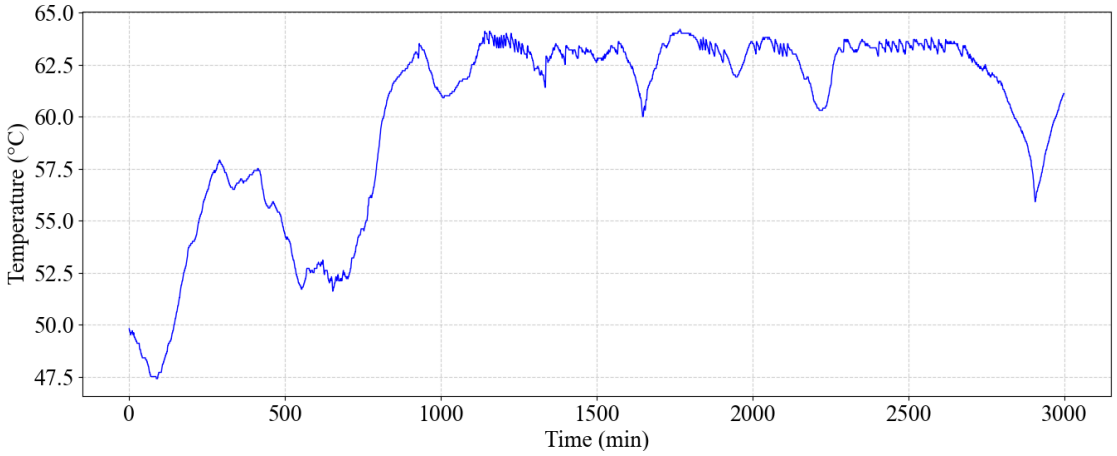


Figure 6. Temperature data of the motor.

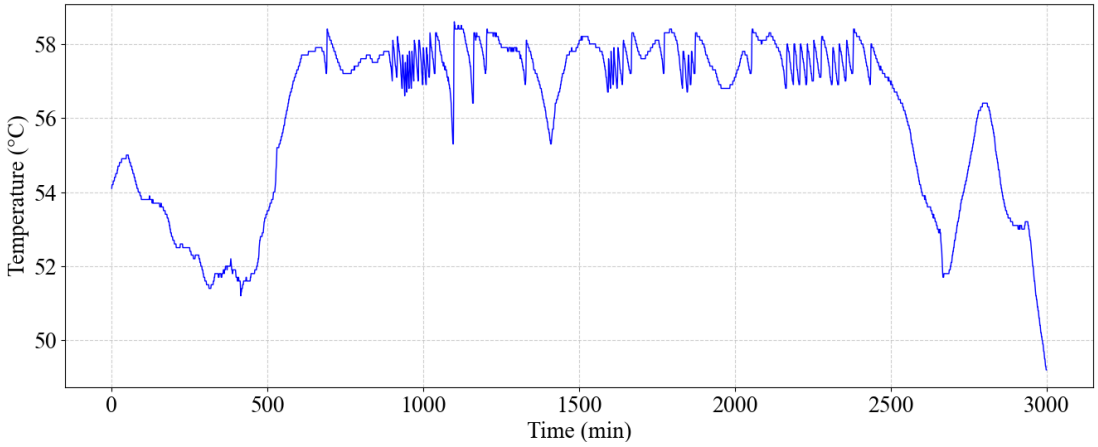


Figure 7. Temperature data of the gearbox.

To maintain fairness in the experiments and optimize model training, the collected data were divided into training and testing sets in an 8:2 ratio. Additionally, to fully capture the temporal characteristics of the time-series data, the sliding window size for each sample was set to 16-time steps. This window size was selected as it effectively incorporates periodic and trend variations, enhancing the model's ability to predict temperature fluctuations. To prevent data leakage and ensure consistency, min-max normalization was applied using the minimum and maximum values derived solely from the training set. The same scaling parameters were then used for the test set, ensuring consistent data distribution throughout the training and evaluation phases.

4.2. Case Study

4.2.1. Case Study 1: Wind Turbine Motor Temperature Predict

In the first experiment of this study, motor temperature data from the wind turbine were selected to assess the practicality and reliability of the proposed model. To provide a comprehensive evaluation of the model's performance, we employed a consistent validation method using real temperature datasets. Additionally, to further verify the performance improvements resulting from the optimized model architecture, several comparative experiments were conducted with different model configurations. These experiments included the following models: DSCNN-BiLSTM (M0), DSCNN (M1), RNN (M2), BiLSTM (M3), and the CNN-BiLSTM hybrid model (M4). The selection of these models was based on their relevance to the time-series prediction tasks and their frequent use in prior studies related to temperature forecasting and fault diagnosis. To ensure a fair comparison, all models were trained using the same training hyperparameters and preprocessing conditions. A thorough quantitative and qualitative analysis was then performed to evaluate the performance of each model in the temperature prediction task. This analysis aimed to assess the feasibility, advantages, and limitations of each model in practical applications, ultimately confirming the superiority of the proposed DSCNN-BiLSTM hybrid model.

As shown in Figure 8, the training curves of each model exhibit varying degrees of fitting accuracy. While all models capture the general trend of temperature variations, significant differences in fitting precision are observed. The proposed DSCNN-BiLSTM model (M0) demonstrates superior learning ability and robustness, with the smallest discrepancy between the training curve and the actual temperature. This indicates a higher level of fitting accuracy during the training phase. Moreover, when the temperature experiences rapid changes, such as sharp increases or decreases, M0's predictions closely align with the actual values, especially near extreme points, where it is fitting performance outperforms other models. In contrast, the other models (M1, M2, M3, M4) exhibit certain limitations during training. Specifically, M1 shows larger errors, while M2 and M3 gradually approach the actual temperature but lack sufficient responsiveness to finer temperature variations. Although M4 performs well during some stages, it shows larger errors in the latter half of the training period when faced with complex temperature fluctuations, revealing some instability. A comparison between the training curves of M0 and M4 highlights that M0, by incorporating the DSCNN module, significantly reduces model parameters while enhancing local feature extraction. This improvement enables M0 to achieve more accurate fitting when handling complex temperature variations. In contrast, the single-model structures of M1, M2, and M3 limit their learning capacity, preventing them from fully capturing the intricate characteristics of temperature changes during training.

As shown in Figure 9, both the proposed method (M0) and the comparison method (M4) exhibit a high degree of alignment between the predicted temperature curves and the actual temperature curve in the test data, significantly outperforming the other three comparison methods (M1, M2, M3). Throughout the entire test period, both M0 and M4 demonstrate strong predictive performance, particularly during the stable temperature phase (2400–2800 min), where both models closely follow the actual temperature curve with high accuracy. During the rapid temperature drop phase (around 2900 min), M0 and M4 also show superior trend-capturing ability compared to the other methods, with their predicted curves closely aligning with the actual values. In contrast, M1, M2, and M3 exhibit weaker prediction performance across the entire period, especially during phases of rapid temperature change, where their predicted curves deviate significantly from the actual values.

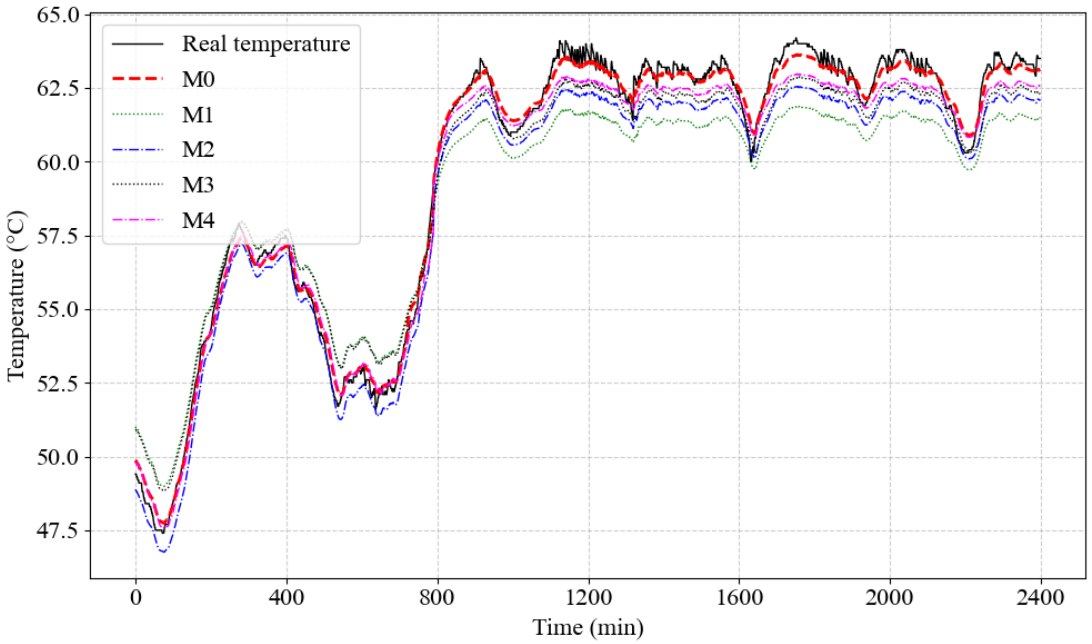


Figure 8. Training data curve of real temperature and predictions from five methods in Case 1.

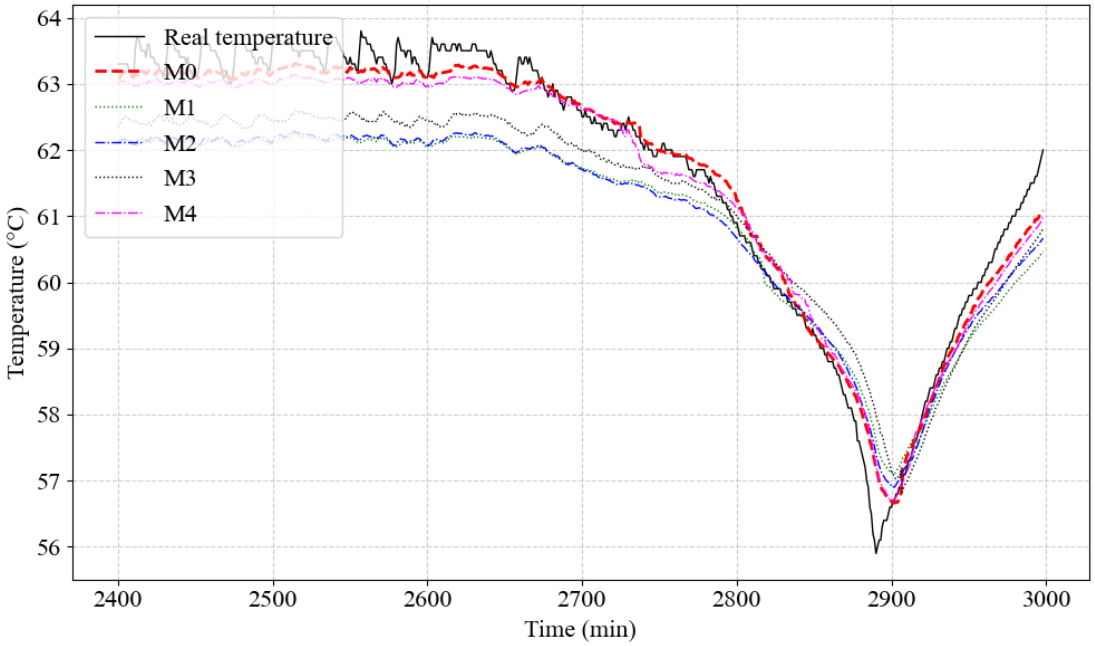


Figure 9. Test data curve of real temperature and predictions from five methods in Case 1.

Further analysis of the error curves (Figure 10) reveals that the error range of M0 and M4 stays within ±0.5 °C, with both models demonstrating significantly lower errors than the other three methods across most time periods. Notably, during the stable temperature region (2400–2800 min), although M4 exhibits relatively small errors, M0 maintains more stable errors with smaller fluctuations, indicating that M0 has higher robustness and accuracy when handling complex nonlinear dynamic features. In contrast, the error fluctuation range of M1, M2, and M3 is wider, with errors exceeding 1.5 °C during certain periods. This is especially evident during rapid temperature changes, where the performance of these methods is noticeably insufficient, failing to effectively capture the temperature dynamics. The experimental results strongly validate the superior performance of the proposed method (M0). M0 excels in overall prediction accuracy, robustness, and dynamic

feature capture, especially in complex nonlinear scenarios, highlighting its greater potential for practical applications. These findings further demonstrate that the proposed method offers an efficient, accurate, and robust solution for predicting mechanical component temperatures, providing valuable technical support for monitoring industrial equipment's operating conditions and early fault warning.

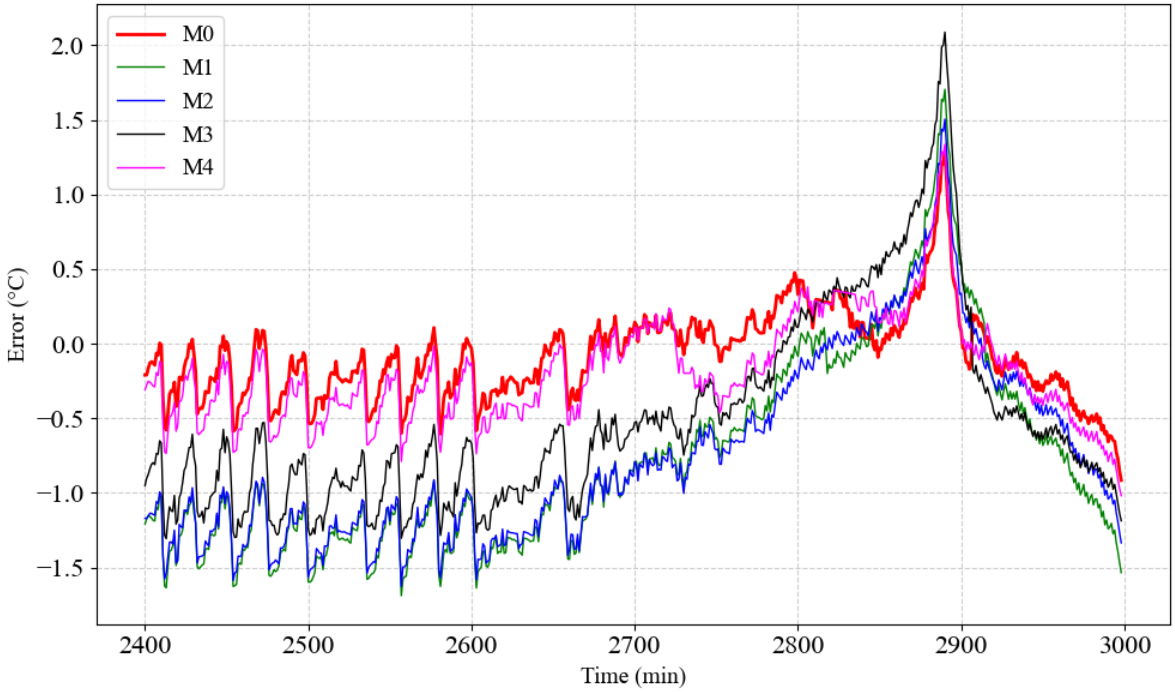


Figure 10. The error between predicted and true temperatures for the five methods on the test data in Case 1.

To quantitatively evaluate the prediction accuracy of the model in this study, two commonly used error metrics—root mean squared error (RMSE) and mean absolute error (MAE)—are employed. These metrics provide a clear and objective assessment of the model's performance. Additionally, to assess the model's suitability for predicting temperatures in this specific wind turbine dataset and to evaluate its ability to fit the data, the coefficient of determination (R^2) is used as a validation metric. The following mathematical expressions define these evaluation criteria:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{X}_i - X_i)^2} \tag{14}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{X}_i - X_i\right| \tag{15}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_i (X_i - \hat{X}_i)^2}{\sum_i \left(X_i - \frac{1}{n}\sum_{i=1}^{n}\hat{X}_i\right)^2} \tag{16}$$

where $\hat{X}_i$ and X_i represent the predicted and actual values of the i-th data point, respectively, and N is the total number of data points. The coefficient of determination R^2, ranges from 0 to 1, with higher values indicating better model performance and a stronger fit to the data.

From the data in Table 2, it is evident that the proposed model, M0, significantly outperforms the comparative models in terms of prediction accuracy. While the prediction

errors of the comparative models are relatively small, they exhibit lower sensitivity to large temperature fluctuations, and their overall prediction accuracy is inferior to that of M0, with larger errors in localized regions. In contrast, the proposed model, M0, not only demonstrates smaller overall errors but also shows superior precision in localized regions and a better ability to respond to rapid temperature changes. Particularly during phases of rapid temperature fluctuation, M0 effectively captures the details of temperature variations, providing more stable and accurate predictions.

Table 2. The corresponding calculated values for the evaluation metrics in Case 1.

Model	RMSE	MAE	R2
M0	**0.309937**	**0.240874**	**0.977622**
M1	1.010780	0.898270	0.761995
M2	0.968196	0.858003	0.781627
M3	0.817695	0.740345	0.844240
M4	0.400906	0.335414	0.962558

Remarks: Bold values represent the best performance metrics achieved by the methods proposed.

This indicates that M0 offers a significant advantage in the temperature prediction tasks for wind farm motors, enabling more effective early warning systems and facilitating the timely detection of potential equipment faults or anomalies. Further analysis using the coefficient of determination R^2 reveals significant differences between the models. With an R^2 value of 0.9776, M0 considerably outperforms the other models, indicating its superior ability to fit the actual data and its higher adaptability in the specific application scenario. Therefore, M0 not only exhibits high prediction accuracy but also provides more reliable temperature predictions in complex wind farm environments, making it a valuable tool for operational monitoring and fault prediction in industrial equipment.

4.2.2. Case Study 2: Wind Turbine Gearbox Temperature

In the second case study, the gearbox temperature dataset is utilized to evaluate the performance of the proposed model. Figure 11 illustrates the predicted values and actual values of the gearbox training temperature data for five different methods.

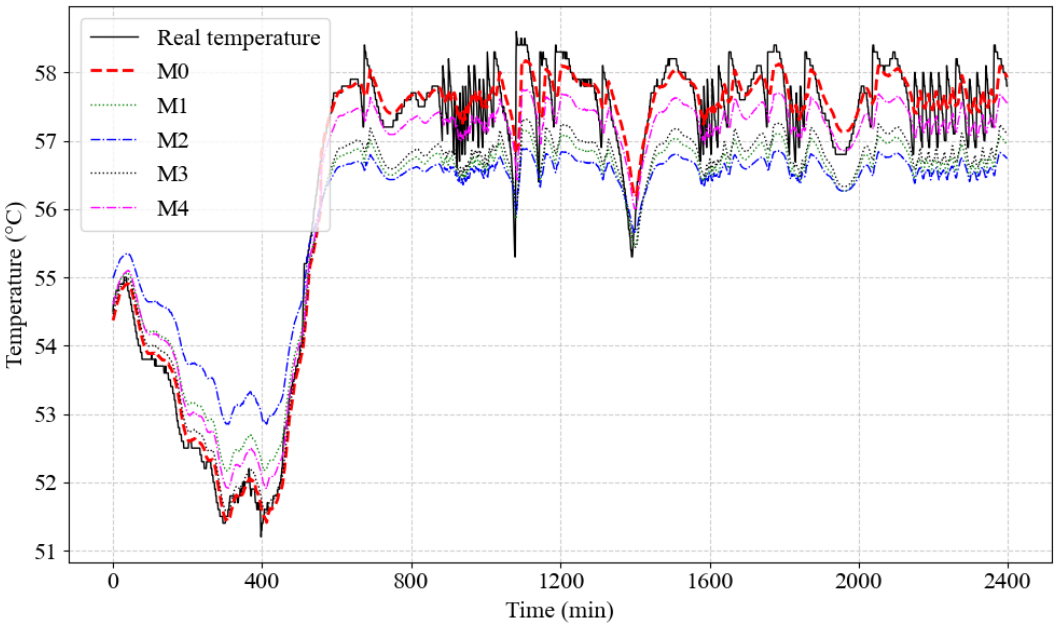


Figure 11. Training data curve of real temperature and predictions from five methods in Case 2.

Figure 11 presents a comparison between the predicted values and the actual values of the gearbox training temperature data for five different methods. While the prediction curves of all methods generally follow the trend of the actual temperature curve, the proposed model demonstrates a significantly better fit between the predicted and actual values compared to the other four methods. During phases of large temperature fluctuations, the proposed model more accurately captures the temperature variation trend with smaller prediction errors. In contrast, the other methods exhibit varying degrees of bias during the prediction process, especially during periods of rapid temperature changes, where the gap between the predicted and actual values becomes notably larger. Additionally, some methods show lower prediction accuracy in localized regions, failing to effectively capture the detailed temperature variations. Overall, the experimental results highlight that the proposed model outperforms the other methods in terms of prediction accuracy and robustness, particularly for gearbox temperature prediction tasks.

From Figures 12 and 13, it can be observed that the proposed model M0 demonstrates significant advantages in the gearbox temperature prediction task. Figure 12 shows the comparison between the predicted temperatures and the actual temperatures for five models. The prediction curve of M0 aligns closely with the actual temperature curve, especially in regions with rapid temperature changes (e.g., the sharp drop around 2700 min and the rapid rise around 2800 min). M0 accurately captures the temperature fluctuation trends, exhibiting outstanding modeling capability and adaptability to complex variations. In contrast, the prediction accuracy of the other models is significantly lower. Particularly during rapid temperature changes, the predicted peaks and troughs of these models generally show delays or deviations. Although M1 can follow temperature changes in some regions, its overall trend-fitting capability is insufficient. M2 and M3 show weaker trend-capturing abilities in regions with complex temperature fluctuations, with large deviations between predicted and actual values. While M4 performs relatively well overall, its prediction accuracy is slightly inferior to M0 in some rapid fluctuation regions, which may be due to its limited ability to suppress noise.

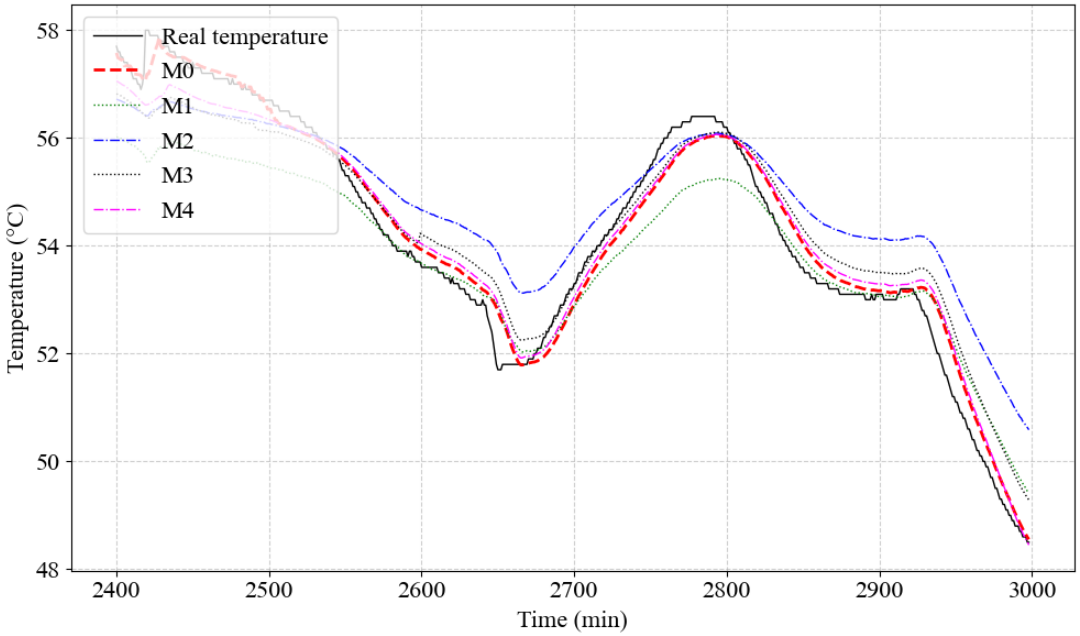


Figure 12. Test data curve of real temperature and predictions from five methods in Case 2.

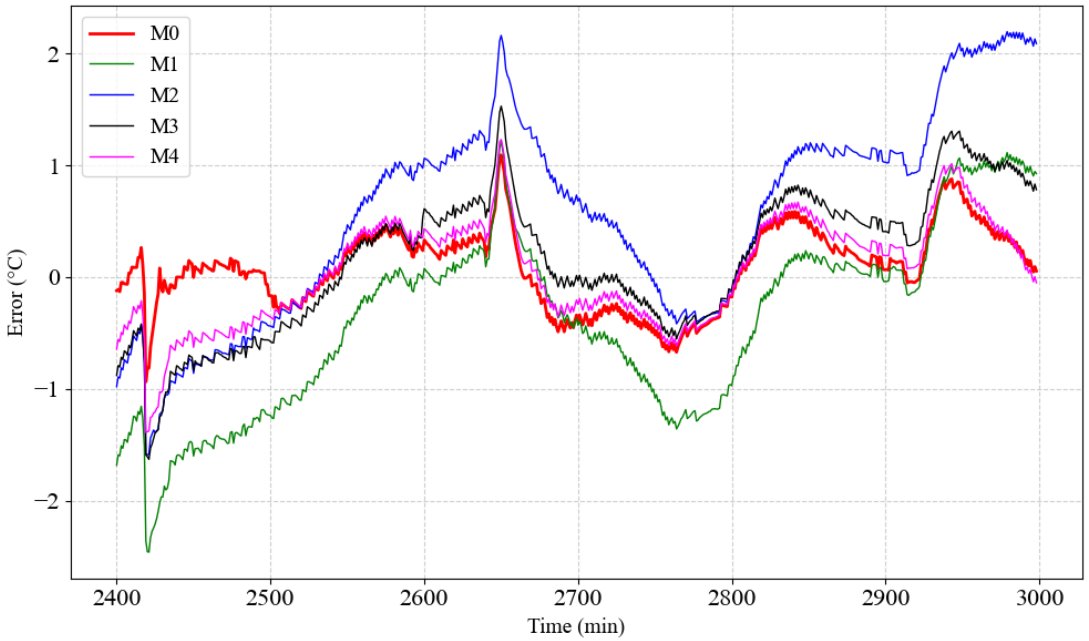


Figure 13. The error between predicted and true temperatures for the five methods on the test data in Case 2.

Figure 13 further illustrates the error trends for each model, providing a more intuitive view of the superiority of M0. The absolute error of M0 remains mostly within ±0.5 °C, even in the highly fluctuating region around 2700 min, where the peak error only slightly exceeds 1 °C. This demonstrates the strong ability of M0 to fit complex fluctuations. In contrast, the error ranges of other models are significantly larger: M1 and M2 show the poorest performance, with overall errors exceeding ±1.5 °C and peak errors surpassing 2 °C in the regions of rapid fluctuation, revealing their limitations in capturing complex nonlinear changes. The error trend of M3 lies between M2 and M4, with a peak error of approximately 1.3 °C, indicating its limited capability in handling rapid fluctuations. M4's error is generally close to M0, with most errors within ±1.0 °C, but it exhibits larger errors in highly fluctuating regions, possibly due to limitations in model robustness. In summary, the proposed DSCNN-BiLSTM model, by combining the advantages of depthwise separable convolution and bidirectional long short-term memory networks, demonstrates superior modeling capability and adaptability to complex temperature variations. It significantly reduces prediction errors, validating its effectiveness and reliability in the gearbox temperature prediction task.

Table 3 presents the performance metrics (RMSE, MAE, and R^2) of the five models on the gearbox test dataset. The results demonstrate that the proposed DSCNN-BiLSTM model M0 achieves the best performance across all metrics, with an RMSE of 0.364, an MAE of 0.301, and an R^2 of 0.973, indicating its significant advantage in accurately predicting gearbox temperatures. In contrast, the other models exhibit noticeably inferior performance. M1 and M2 show RMSE values of 0.893 and 1.068, respectively, with R^2 values of only 0.837 and 0.767, highlighting their limited ability to capture the complex temperature variations. Although M3 shows some improvement, it still lags behind M0, while M4 achieves performance close to M0, with an R^2 of 0.954. However, its prediction accuracy remains slightly lower. In summary, the proposed DSCNN-BiLSTM model effectively integrates the strengths of depthwise separable convolution and bidirectional long short-term memory networks, showcasing superior modeling capability and prediction accuracy. This validates its effectiveness and reliability in gearbox temperature prediction tasks.

Table 3. The corresponding calculated values for the evaluation metrics in Case 2.

Model	RMSE	MAE	R2
M0	**0.364288**	**0.300988**	**0.972877**
M1	0.892547	0.697479	0.837182
M2	1.067547	0.906167	0.767076
M3	0.641056	0.545006	0.916009
M4	0.472086	0.407460	0.954451

Remarks: Bold values represent the best performance metrics achieved by the methods proposed.

4.2.3. Model Performance with Noisy Data

To assess the robustness of the proposed model under noisy conditions, an additional experiment was designed where synthetic Gaussian noise was introduced into the temperature data. The noise level was controlled using the Signal-to-Noise Ratio (SNR), which is expressed in decibels (dB). The purpose of this experiment was to investigate how the model's performance is affected by various noise levels in real-world settings. In this experiment, Gaussian noise was added to the original motor temperature data at three different SNR levels: 5 dB, 0 dB, and −5 dB, representing low, medium, and high noise levels, respectively. The SNR values were calculated based on the original signal power and the noise power as follows:

$$SNR = 10log_{10}(\frac{P_{signal}}{P_{noise}}) \quad (17)$$

where P_{signal} is the power of the original temperature signal, and P_{noise} is the power of the added noise.

The experimental results presented in Tables 4–6 demonstrate the performance of different models under varying noise levels (5 dB, 0 dB, and −5 dB). As the noise level increases, all models show a degradation in performance. However, the DSCNN-BiLSTM model (M0) consistently outperforms the other models across all noise conditions. Under 5 dB noise, M0 achieves the lowest RMSE (0.4745), MAE (0.4121), and the highest R^2 (0.9534), indicating its robustness. As noise increases to 0 dB and −5 dB, the performance of all models declines, but M0 still maintains the best performance with an RMSE of 0.6388 and 0.7152, respectively, and an R^2 of 0.9153 and 0.8849. The other models, such as M1 and M2, experience a more significant drop in performance, especially under higher noise levels, with a considerable increase in RMSE and a decrease in R^2. Overall, the DSCNN-BiLSTM model demonstrates superior noise resilience and predictive accuracy, outperforming other models even as the noise level increases, highlighting its potential for practical applications in noisy environments.

However, while M0 excels in noisy conditions, it also has some limitations. The model's complexity, due to the hybrid architecture combining DSCNN and BiLSTM, may lead to longer training times and require more computational resources compared to simpler models like M1 and M3. Furthermore, although the model is robust to noise, it may still struggle in environments with extremely high noise or when there is a lack of sufficient training data. Therefore, future work could focus on optimizing the model for even higher noise levels or exploring methods to reduce computational costs while maintaining performance.

Table 4. The prediction performance of different models under −5 dB noise in Case 2.

Model	SNR (dB)	RMSE	MAE	R2
M0	**5 dB**	**0.474502**	**0.412057**	**0.953419**
M1	5 dB	0.937174	0.746303	0.820438
M2	5 dB	1.120924	0.969599	0.751734
M3	5 dB	0.693109	0.613156	0.897689
M4	5 dB	0.505690	0.435982	0.935362

Remarks: Bold values represent the best performance metrics achieved by the methods proposed.

Table 5. The prediction performance of different models under 0 dB noise in Case 2.

Model	SNR (dB)	RMSE	MAE	R2
M0	**0 dB**	0.638792	0.538120	0.915324
M1	0 dB	1.275849	1.034128	0.735204
M2	0 dB	1.509732	1.362498	0.648291
M3	0 dB	0.902756	0.782401	0.823914
M4	0 dB	0.766503	0.639325	0.877184

Remarks: Bold values represent the best performance metrics achieved by the methods proposed.

Table 6. The prediction performance of different models under 5 dB noise in Case 2.

Model	SNR (dB)	RMSE	MAE	R2
M0	**−5 dB**	0.715184	0.612340	0.884921
M1	−5 dB	1.495673	1.267981	0.690475
M2	−5 dB	1.832104	1.703849	0.598214
M3	−5 dB	1.115320	1.008732	0.762108
M4	−5 dB	0.878129	0.691936	0.839917

Remarks: Bold values represent the best performance metrics achieved by the methods proposed.

4.2.4. Ablation Study

To further evaluate the impact of key architectural choices on the predictive performance of the proposed model, an ablation study was conducted using a wind turbine gearbox dataset. In this study, the DSCNN-BiLSTM framework was systematically compared with alternative architectures, including TCN, GRU, and Transformer-based models. By varying the network structure and assessing its predictive accuracy on the gearbox dataset, the results presented in Table 4 were obtained.

Table 7 presents the performance comparison of different network architectures on the wind turbine gearbox temperature dataset, demonstrating the effectiveness of the proposed DSCNN-BiLSTM model. The results indicate that DSCNN-BiLSTM achieves the lowest RMSE and MAE, along with the highest R^2 score, highlighting its superior ability to model temperature variations and detect anomalies.

Table 7. The ablation study of results among the six models in Case 2.

Model	RMSE	MAE	R2
DSCNN-BiLSTM	**0.364288**	**0.300988**	**0.972877**
DSCNN	0.892547	0.697479	0.837182
BiLSTM	0.641056	0.545006	0.916009
TCN	0.910234	0.705328	0.829672
GRU	0.678923	0.563872	0.904112
Transformer	0.598712	0.489331	0.928374

Remarks: Bold values represent the best performance metrics achieved by the methods proposed.

Compared to TCN, DSCNN achieves better performance, as its depthwise separable convolutions enhance feature extraction efficiency while reducing computational cost. TCN,

despite incorporating temporal modeling, has a limited receptive field and lacks a memory mechanism, making it less effective for capturing long-term temperature trends. On the other hand, RNN-based models (BiLSTM, GRU) show noticeable improvements over CNN-based models, as they can effectively learn sequential dependencies. However, GRU, despite being computationally efficient, has a simpler gating mechanism compared to BiLSTM, which may limit its ability to retain long-term dependencies in complex temperature sequences. While Transformer-based models perform competitively, their self-attention mechanism introduces high computational overhead, making them less suitable for real-time industrial applications where efficiency is crucial.

By integrating DSCNN for spatial feature extraction and BiLSTM for sequential modeling, the proposed DSCNN-BiLSTM model effectively captures both local temperature fluctuations and long-term temporal dependencies, leading to more accurate and stable temperature predictions. These results confirm that DSCNN-BiLSTM provides the best balance between feature extraction capability, sequential modeling effectiveness, and computational efficiency, making it the optimal choice for wind turbine gearbox temperature monitoring and fault diagnosis.

5. Conclusions

This paper addresses the temperature prediction problem for wind turbines by proposing a hybrid model based on depthwise separable convolutional neural networks and bidirectional long short-term memory networks. The model's performance is validated in two experimental scenarios: motor and gearbox. With the widespread deployment of wind turbines, equipment temperature monitoring has become a key factor in improving operational efficiency and reliability. Traditional temperature prediction methods face challenges in handling complex nonlinear temperature variations and noise interference. Therefore, the hybrid architecture presented in this study combines convolutional neural networks and long short-term memory networks to enhance the accuracy and robustness of temperature forecasts, especially in large-scale industrial applications.

The experimental results demonstrate that the proposed DSCNN-BiLSTM hybrid model significantly outperforms traditional models in temperature prediction tasks for both the motor and gearbox scenarios. Specifically, the experiments show the following:

1. The DSCNN-BiLSTM hybrid model effectively extracts local features using depthwise separable convolutions and captures temporal dependencies with BiLSTM, combining the strengths of both for accurate temperature trend tracking.
2. The end-to-end temperature prediction method processes raw signals directly, avoiding manual feature engineering and excessive prior knowledge. This enhances the model's generalization and adaptability for diverse operating conditions, especially in complex industrial equipment like wind turbines.
3. Experimental results in both motor and gearbox scenarios show that the proposed model outperforms baseline models in prediction accuracy, local precision, and response to temperature fluctuations, highlighting its strong potential for real-world industrial applications.
4. Despite its strong performance, the model has certain limitations. Its robustness under extreme noise conditions remains to be fully explored, and its relatively high computational complexity may hinder deployment in resource-constrained environments. Future work could enhance noise resistance through advanced denoising techniques, improve efficiency via model pruning, and incorporate multi-sensor fusion for better generalization. Addressing these aspects will further strengthen the applicability of deep learning-based temperature prediction in industrial settings.

Author Contributions: Conceptualization, X.L.; Validation, Z.Z.; Formal analysis, Z.Z.; Investigation, Z.Q.; Writing—original draft, Z.Q.; Writing—review & editing, J.H.; Supervision, X.L. and J.H. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this article was partially supported by a grant from the National Key Research and Development Program of China [Grant NO. 2023***3106505]; Key Research and Development Program of Hubei Province [Grant NO. 2023BCB047]; Hainan Provincial Natural Science Foundation of China [Grant NO. 522RC879]; Hainan Provincial Natural Science Foundation of China [Grant NO. 522CXTD517].

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, M.; Shi, J.; Yang, S.; Chen, M.; Tang, Y.; Liu, S. Current Status and Future Trends in Installation, Operation and Maintenance of Offshore Floating Wind Turbines. *J. Mar. Sci. Eng.* **2024**, *12*, 2155. [CrossRef]
2. Zhang, W.; Li, B.; Xue, R.; Wang, C.; Cao, W. A systematic bibliometric review of clean energy transition: Implications for low-carbon development. *PLoS ONE* **2021**, *16*, e0261091. [CrossRef]
3. Liu, F.; Sun, F.; Wang, X. Impact of turbine technology on wind energy potential and CO_2 emission reduction under different wind resource conditions in China. *Appl. Energy* **2023**, *348*, 121540. [CrossRef]
4. Gao, Z.; Odgaard, P. Real-time monitoring, fault prediction and health management for offshore wind turbine systems. *Renew. Energy* **2023**, *218*, 119258. [CrossRef]
5. Yang, J.; Fang, L.; Song, D.; Su, M.; Yang, X.; Huang, L.; Joo, Y.H. Review of control strategy of large horizontal-axis wind turbines yaw system. *Wind Energy* **2021**, *24*, 97–115. [CrossRef]
6. Su, Y.; Meng, L.; Kong, X.; Xu, T.; Lan, X.; Li, Y. Small sample fault diagnosis method for wind turbine gearbox based on optimized generative adversarial networks. *Eng. Fail. Anal.* **2022**, *140*, 106573. [CrossRef]
7. Liu, Z.; Zhang, L. Naturally damaged wind turbine blade bearing fault detection using novel iterative nonlinear filter and morphological analysis. *IEEE Trans. Ind. Electron.* **2019**, *67*, 8713–8722. [CrossRef]
8. Wen, X.; Xu, Z. Wind turbine fault diagnosis based on ReliefF-PCA and DNN. *Expert Syst. Appl.* **2021**, *178*, 115016. [CrossRef]
9. Liu, H.; Yu, C.; Yu, C. A new hybrid model based on secondary decomposition, reinforcement learning and SRU network for wind turbine gearbox oil temperature forecasting. *Measurement* **2021**, *178*, 109347. [CrossRef]
10. Inturi, V.; Shreyas, N.; Chetti, K.; Sabareesh, G.R. Comprehensive fault diagnostics of wind turbine gearbox through adaptive condition monitoring scheme. *Appl. Acoust.* **2021**, *174*, 107738. [CrossRef]
11. Jiang, G.; Li, W.; Fan, W.; He, Q.; Xie, P. TempGNN: A temperature-based graph neural network model for system-level monitoring of wind turbines with SCADA data. *IEEE Sens. J.* **2022**, *22*, 22894–22907. [CrossRef]
12. Zhang, X.; Zhong, J.; Li, W.; Bocian, M. Nonlinear dynamic analysis of high-speed gear pair with wear fault and tooth contact temperature for a wind turbine gearbox. *Mech. Mach. Theory* **2022**, *173*, 104840. [CrossRef]
13. Foley, A.M.; Leahy, P.G.; Marvuglia, A.; McKeogh, E.J. Current methods and advances in forecasting of wind power generation. *Renew. Energy* **2012**, *37*, 1–8. [CrossRef]
14. González-Sopeña, J.M.; Pakrashi, V.; Ghosh, B. An overview of performance evaluation metrics for short-term statistical wind power forecasting. *Renew. Sustain. Energy Rev.* **2021**, *138*, 110515. [CrossRef]
15. Gu, C.; Zhu, M.; Wu, Y.; Chen, B.; Zhou, F.; Chen, W. Multi-output displacement health monitoring model for concrete gravity dam in severely cold region based on clustering of measured dam temperature field. *Struct. Health Monit.* **2023**, *22*, 3416–3436. [CrossRef]
16. Chen, X.; Jiang, Z.; Cheng, H.; Zheng, H.; Cai, D.; Feng, Y. A novel global average temperature prediction model—Based on GM-ARIMA combination model. *Earth Sci. Inform.* **2024**, *17*, 853–866. [CrossRef]
17. Das, S.; Politis, D.N. Predictive inference for locally stationary time series with an application to climate data. *J. Am. Stat. Assoc.* **2021**, *116*, 919–934. [CrossRef]
18. Zou, L.; Lam, H.F.; Hu, J. Adaptive resize-residual deep neural network for fault diagnosis of rotating machinery. *Struct. Health Monit.* **2023**, *22*, 2193–2213. [CrossRef]
19. Shinde, P.P.; Shah, S. A review of machine learning and deep learning applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; IEEE: New York, NY, USA, 2018; pp. 1–6. [CrossRef]

20. Dargan, S.; Kumar, M.; Ayyagari, M.R.; Kumar, G. A survey of deep learning and its applications: A new paradigm to machine learning. *Arch. Comput. Methods Eng.* **2020**, *27*, 1071–1092. [CrossRef]
21. Graves, A.; Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45. [CrossRef]
22. Wang, Z.; Wang, Q.; Wu, T. A novel hybrid model for water quality prediction based on VMD and IGOA optimized for LSTM. *Front. Environ. Sci. Eng.* **2023**, *17*, 88. [CrossRef]
23. Shahid, F.; Zameer, A.; Mehmood, A.; Raja, M.A.Z. A novel wavenets long short term memory paradigm for wind power prediction. *Appl. Energy* **2020**, *269*, 115098. [CrossRef]
24. Huang, R.; Wei, C.; Wang, B.; Yang, J.; Xu, X.; Wu, S.; Huang, S. Well performance prediction based on Long Short-Term Memory (LSTM) neural network. *J. Pet. Sci. Eng.* **2022**, *208*, 109686. [CrossRef]
25. Cui, J.; Zhang, M.; Song, D.; Shan, X.; Wang, B. MODIS land surface temperature product reconstruction based on the SSA-BiLSTM model. *Remote Sens.* **2022**, *14*, 958. [CrossRef]
26. Zhang, X.; Ren, H.; Liu, J.; Zhang, Y.; Cheng, W. A monthly temperature prediction based on the CEEMDAN–BO–BiLSTM coupled model. *Sci. Rep.* **2024**, *14*, 808. [CrossRef]
27. Jiang, L.; Yan, C.; Zhang, X.; Zhou, B.; Cheng, T.; Zhao, J.; Gu, J. Temperature prediction of battery energy storage plant based on EGA-BiLSTM. *Energy Rep.* **2022**, *8*, 1009–1018. [CrossRef]
28. Zheng, H.; Lin, F.; Feng, X.; Chen, Y. A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 6910–6920. [CrossRef]
29. Tabrizchi, H.; Razmara, J.; Mosavi, A. Thermal prediction for energy management of clouds using a hybrid model based on CNN and stacking multi-layer bi-directional LSTM. *Energy Rep.* **2023**, *9*, 2253–2268. [CrossRef]
30. Ji, Z.; Tao, W.; Ren, J. Boiler furnace temperature and oxygen content prediction based on hybrid CNN, biLSTM, and SE-Net models. *Appl. Intell.* **2024**, *54*, 8241–8261. [CrossRef]
31. Jiang, B.; Gong, H.; Qin, H.; Zhu, M. Attention-LSTM architecture combined with Bayesian hyperparameter optimization for indoor temperature prediction. *Build. Environ.* **2022**, *224*, 109536. [CrossRef]
32. Yu, Z.; Wang, Y.; Zong, X.; Wu, J.; Zhou, Q. GRUDMU-DSCNN: An edge computing method for fault diagnosis with missing data. *Appl. Intell.* **2025**, *55*, 140. [CrossRef]
33. Xie, F.; Tang, X.; Xiao, F.; Luo, Y.; Shen, H.; Shi, Z. Online diagnosis method for open-circuit fault of NPC inverter based on 1D-DSCNN-GMP lightweight edge deployment. *IEEE J. Emerg. Sel. Top. Power Electron.* **2023**, *11*, 6054–6067. [CrossRef]
34. Wang, Y.; Liu, Y.; Li, X.; Wang, T.; Xu, Z.; Guo, P.; Liao, B. Intelligent Recognition for Operation States of Hydroelectric Generating Units Based on Data Fusion and Visualization Analysis. *Int. J. Intell. Syst.* **2025**, *2025*, 8850566. [CrossRef]
35. Li, Z.; Ren, A.; Li, J.; Qiu, Q.; Wang, Y.; Yuan, B. Dscnn: Hardware-oriented optimization for stochastic computing based deep convolutional neural networks. In Proceedings of the 2016 IEEE 34th International Conference on Computer Design (ICCD), Scottsdale, AZ, USA, 2–5 October 2016; IEEE: New York, NY, USA, 2016; pp. 678–681. [CrossRef]
36. Song, B.; Liu, Y.; Fang, J.; Liu, W.; Zhong, M.; Liu, X. An optimized CNN-BiLSTM network for bearing fault diagnosis under multiple working conditions with limited training samples. *Neurocomputing* **2024**, *574*, 127284. [CrossRef]
37. Cheng, Y.; Hu, K.; Wu, J.; Zhu, H.; Shao, X. A convolutional neural network based degradation indicator construction and health prognosis using bidirectional long short-term memory network for rolling bearings. *Adv. Eng. Inform.* **2021**, *48*, 101247. [CrossRef]
38. Xu, Z.; Mei, X.; Wang, X.; Yue, M.; Jin, J.; Yang, Y.; Li, C. Fault diagnosis of wind turbine bearing using a multi-scale convolutional neural network with bidirectional long short term memory and weighted majority voting for multi-sensors. *Renew. Energy* **2022**, *182*, 615–626. [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.