IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 January 2024, accepted 2 February 2024, date of publication 8 February 2024, date of current version 15 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3364395

RESEARCH ARTICLE

Short-Term Fault Prediction of Wind Turbines Based on Integrated RNN-LSTM

V SIVA BRAHMAIAH RAMA, SUNG-HO HUR, (Member, IEEE), AND JUNG-MIN YANG
School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea

Corresponding authors: Sung-Ho Hur (shur@knu.ac.kr) and Jung-Min Yang (jmyang@ee.knu.ac.kr)

This work was supported in part by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) Grant funded by the Korean Government (MOTIE) under Grant 20224000000040, and in part by Korea Electric Power Corporation under Grant R21XO01-17.

ABSTRACT This paper presents a data-driven approach to short-term wind turbine fault prediction and condition monitoring based on a hybrid architecture of recurrent neural network and long short-term memory. The proposed architecture is established by utilizing time series data from the supervisory control and data acquisition system and a Bladed model of a 5 MW wind turbine to predict faults occurring to the wind generator. The recurrent neural network-long short-term memory training procedure is enhanced with self-organizing maps and long short-term memory auto encoder so as to describe the complex interaction between the mechanical system and unpredictable wind speed. To verify the performance of the proposed scheme, we conduct in-depth numerical experiments by applying the hybrid architecture to the Bladed 5 MW wind turbine model with rated wind speed of 11.8 m/s. Experimental results confirm that the proposed scheme has superior accuracy and practicality of fault prediction compared with eminent existing machine learning algorithms such as extreme gradient boost and random forest regressor.

INDEX TERMS Extreme gradient boosting, fault prediction, long short-term memory autoencoder, random forest regressor, self-organizing maps, wind turbine.

I. INTRODUCTION

Due to increasing concerns about climate change, wind power, a representative renewable energy, has been widely explored [1]. However, unpredictable nature of wind patterns makes it difficult to harness wind energy effectively, which necessitates reliable wind turbine monitoring. This technique is also crucial to reducing costs and optimizing performance, as maintenance accounts for up to 35% of overall wind farm project costs [2]. Indeed, wind turbines are complex systems with numerous components that must be maintained on-line for sturdy operations [3].

A short-term prediction model for wind turbines aims to predict the dynamic responses of the targeted wind turbine in a time scale of several seconds to minutes [4]. As wind turbines are increasingly equipped with intelligent real-time control logic and connected to smart grids, short-term prediction of wind turbine responses becomes an indispensable part [5]. Predicting short-term wind turbine responses is regarded more difficult than other time scale predictions because of chaotic and stochastic characteristics of turbulent flow [6]. Compared to traditional methodologies based on physical-based models, machine learning and deep learning have provided simpler and more effective solutions to predictive maintenance without explicit programming [7], [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Ehab Elsayed Elattar.

A. OBJECTIVE AND MOTIVATION

In this article, we address a data-driven mechanism for wind turbine condition monitoring by presenting a hybrid deep learning model that combines recurrent neural network (RNN) and long short-term memory (LSTM) networks. In the proposed scheme, the self-organizing map (SOM) and long short-term memory auto encoder (LSTM-AE) are first employed for data cleaning and feature extraction. We then utilize both supervisory control and data acquisition (SCADA) and Bladed [9] simulation data to provide a comprehensive learning model of the turbine behavior.

To evaluate the effectiveness of the proposed scheme, we conduct numerical experiments on fault detection for a 5 MW wind turbine regarding wind speed, generator speed and output power. Comparative studies with other established machine learning algorithms are also provided to demonstrate the superiority of the proposed scheme.

The choice of the RNN-LSTM hybrid architecture over conventional ones such as transformer-based models is rather deliberate. The primary motivation is to strike a balance between model complexity, computational efficiency, and performance accuracy. In fault detection for real-time systems like wind turbines, response time is critical. While highly expressive and accurate, transformer-based models often come with an increased computational overhead [10]. This can potentially slow down the fault detection process, especially when dealing with large volumes of SCADA data. The RNN-LSTM fusion, on the other hand, can trade off its performance, namely, harnessing the depth of time series data without overburdening computational resources [11].

In recent years, deep learning models have demonstrated significant promise in modeling sequential data [12] among which LSTM networks have emerged as a favorite by virtue of their ability to pinpoint long-range dependencies. However, solely relying on LSTMs may not always be optimal [13]. By contrast, hybrid models of RNN and LSTM leverage the capability of the methods such as offering better mitigation against the vanishing gradient problem [14]. Further, unlike existing approaches relying only on LSTM models, the proposed method incorporates SOM and the AE-based LSTM model to detect and remove outliers and anomalies from the wind turbine data. This will result in cleaner data, which is then applied to deriving the RNN-LSTM integrated hybrid model emulating wind turbine performance.

B. RELATED WORK

In the field of wind turbine condition monitoring using time series data, several techniques have been reported in recent years. In the realm of wind turbines, [15] introduces a fault diagnosis approach for gearboxes using vibration signal analysis, combining wavelet packet decomposition and convolution neural networks (CNNs). For wind power forecasting, [16] employs an echo state network-based ensemble, focusing on improved accuracy through time series analysis of wind power generation. Anomaly detection in wind turbines is addressed [17] using LSTM-based stacked denoising AEs and extreme gradient boosting (XGBoost), utilizing SCADA data to enhance detection and diagnosis accuracy. Adaptive kernel spectral clustering paired with deep LSTM networks is used in [18] for early anomaly detection and failure prediction in machinery, marking an advancement in industrial health monitoring. Exploring fault diagnosis in wind energy converters, [19] applies PCA and HMM for effective feature extraction and precise classification. Reference [20] similarly adopts adversarial representation learning for intelligent condition monitoring, focusing on operational data analysis in signal and latent spaces. For fault diagnosis in wind turbine transmission systems, [21] utilizes manifold learning and a Shannon wavelet support vector machine, improving fault detection accuracy by analyzing complex signal data. These diverse approaches underscore the evolving landscape of machine learning applications in wind turbine monitoring.

The proposed scheme integrating RNN and LSTM bears some resemblance to existing methods for wind turbine condition monitoring based on hybrid learning models. [22] introduces a fault detection method for wind turbines based on SCADA data, utilizing CNN and LSTM with an attention mechanism. This method emphasizes the extraction of dynamic changes from SCADA data, offering a refined model for early anomaly detection and fault diagnosis. Reference [23] presents a novel genetic LSTM model for wind power forecasting, incorporating a genetic algorithm to optimize LSTM parameters and enhance forecasting accuracy. Reference [24] describes an innovative fault diagnosis approach for reciprocating compressors using a Bayesian optimized LSTM model, employing a time series dimensionality reduction technique to process vibration signal data efficiently. Finally [25] introduces a methodology integrating temporal change information (CIL) with LSTM for dynamic sequence modeling, focusing on capturing abrupt and gradual changes in time series data.

Aside from single SOM and LSTM-AE, many methods have been developed for detecting and removing outliers and anomalies. Reference [26] provides a comprehensive survey of clustering algorithms, offering insights into the effectiveness of different algorithms in pattern recognition. Reference [27] highlights the application of SOM in visualizing high-dimensional data, demonstrating its ability to uncover complex relationships where standard visualization falls short. In the avenue of high-dimensional distribution support, [28] discusses the intricate use of Support Vector Machines, noting the need for careful optimization. Complementing this, [29] examines the use of LSTM-AE in forecasting and anomaly detection. This research emphasizes LSTM-AE's capacity for managing long-term dependencies in sequential data, proving its effectiveness in anomaly detection scenarios.

C. CONTRIBUTIONS

In comparison with the prior work, this article has the following contributions.

1) The proposed methodology advances the field by leveraging an integrated approach of RNN and LSTM. This hybrid model excels in handling complex time series data specific to wind turbine operations, thereby enhancing feature extraction and predictive capabilities. It sets a new benchmark in understanding long-term data dependencies for fault prediction accuracy, offering more refined predictions than standard LSTM networks and existing models that primarily focus on SCADA data alone.

2) Our research broadens the spectrum of utilized data by incorporating outputs from Bladed simulators along with

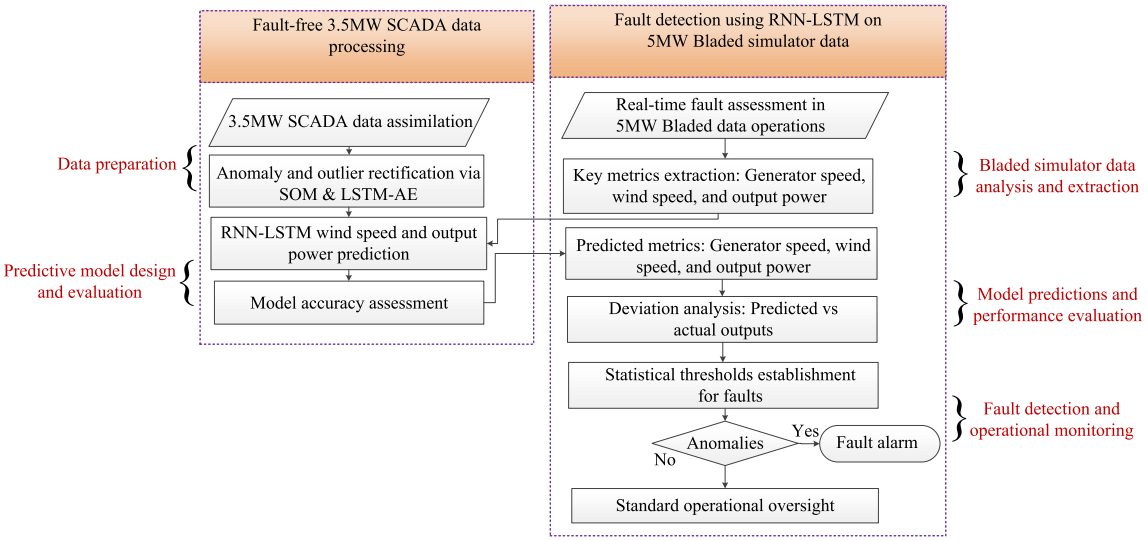


FIGURE 1. Flow chart of predictive maintenance for the wind turbine.

traditional data sources. This comprehensive approach offers a deeper insight into wind turbine dynamics compared to previous schemes that rely on a limited array of data sources.

3) The proposed hybrid model signifies an improvement in term of computational burden over previous ones. It is distinct in its reduced computational demands while maintaining high accuracy in predictions. This streamlined design eliminates complications such as the vanishing gradient problem and overfitting that are common in other models.

4) The present study innovates in outlier and anomaly detection by employing SOM and LSTM-AE. This method contrasts with traditional techniques that often struggle with high-dimensional data and require extensive fine-tuning. Our approach uses SOM for efficient outlier management in high-dimensional spaces and LSTM-AE for nuanced anomaly detection in time series data. This dual strategy simplifies the detection process and enhances the predictive maintenance framework for wind turbines.

Wind speed in SCADA data is typically measured by an anemometer. Despite its lower accuracy compared to high-resolution sensors, its use is established in deep learning research for wind turbine analysis (see, e.g., [30], [31]). Furthermore, the rationale behind the use of SCADA data with a power rating of 3.5 MW alongside Bladed simulator data rated at a different power (5 MW) is multifaceted. Primarily, tackling wind turbine data with heterogeneous power ratings exemplifies the versatility of the proposed methodology, similarly observed in previous studies [32]. Whereas there might be concerns about prediction errors due to power rating discrepancy, the differential behaviors introduced by these ratings are well captured by the proposed hybrid model. Indeed, the operational configuration may differ depending on the power rating. But we note that the fundamental physics guiding their operations remains invariant, allowing the comprehensive model processing as done in this study.

The rest of this article is organized as follows. Section II discusses the methodology for model processing and fault detection in wind turbines. Section III details the data cleaning process and development of the RNN-LSTM hybrid model. Section IV introduces a fault detection algorithm on generator speed and output power. Numerical experiments and performance evaluation of the hybrid model are presented in Section V. Section VI concludes the article.

II. METHODOLOGY

Fig. 1 depicts the structured methodology that is segmented into the following critical phases.

A. DATA PREPARATION

We first harness historical SCADA data of a 3.5 MW wind turbine taken from a recognized dataset in northwest Turkey [33], applying SOM and LSTM-AE for anomaly and outlier detection. The processed data is then divided into training, validation, and testing segments.

B. PREDICTIVE MODEL DESIGN AND EVALUATION

An advanced RNN-LSTM hybrid model is engineered using the refined SCADA data, focusing on critical parameters like wind speed and power output. The model undergoes comprehensive training and validation to ensure precision and reliability.

C. BLADED SIMULATOR DATA ANALYSIS AND EXTRACTION

Switching to the 5 MW wind turbine data obtained from Bladed simulator, we observe traces of operational discrepancies. This dataset enriches the proposed algorithm's versatility. We extract from this data pivotal metrics including generator speed, wind speed, and power output.

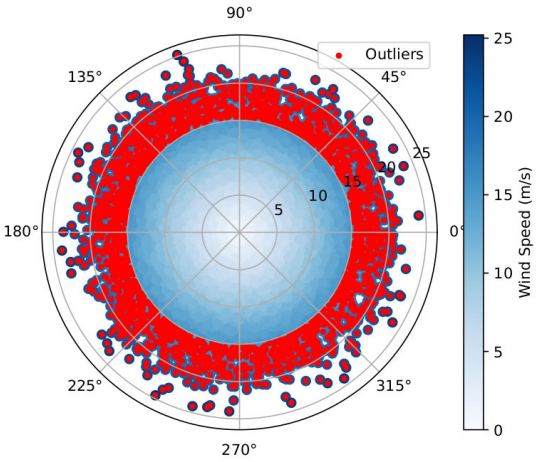


FIGURE 2. Visualized wind conditions in the polar scatter plot.

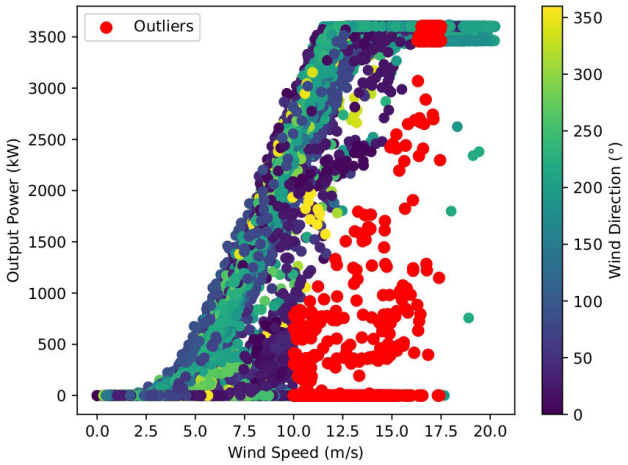


FIGURE 3. Wind turbine power curve before data cleaning.

D. MODEL PREDICTIONS AND PERFORMANCE EVALUATION

The RNN-LSTM model is fine-tuned for fault detection, validated against the Bladed dataset, and evaluated using R-squared (R^2) and Mean Absolute Error (MAE) metrics.

E. FAULT DETECTION AND OPERATIONAL MONITORING

The validation phase utilizes Statistical Process Control (SPC) charts with defined upper control limit (UCL) and lower control limit (LCL) benchmarks. Deviations from these limits trigger fault alarms, enabling proactive measures against potential system anomalies or breakdowns.

III. MODEL PROCESSING

A. SCADA DATA ANALYSIS

Suppose that the wind turbine considered in this study uses a SCADA system to record key operational parameters—output power, wind speed, and generator speed—at a 10-minute sampling frequency wherein only the average values of each parameter are retained to reduce data size and processing time. The wind turbine has rated power of 3.5 MW, rotor diameter of 82 m, and hub height of 80 m. Its cut-in, rated, and cut-out wind speed are 3.5 m/s, 9.2 m/s, and 25 m/s, respectively. Note that the cut-in speed is the minimum for power generation, the rated speed is its optimal operational capacity, and the cut-out speed represents the safety limit to prevent damage in high winds. As these values determine real-time turbine performance, they must be prescribed a priori for proceeding with model derivations.

The data cleaning process, particularly the removal of outliers and anomalies, plays a critical role in our wind turbine modeling study. Wind turbines are subject to a wide range of operational and environmental variables that cause stochastic and systematic errors to SCADA data. These errors may manifest as transient deviations or consistent divergences, potentially leading to inaccurate model predictions and misinterpretations of the turbine's operational status.

In Fig. 2, the polar scatter plot shows the distribution of wind speeds, predominantly below 15 m/s, with some between 15 and 25 m/s. This visualization not just stems from frequency but also incorporates turbine specifications and potential sources of outliers in SCADA data. While Fig. 2 highlights data points with non-uniform distribution, these are not immediately classified as outliers. Instead, they are subject to further domain-specific analysis. Fig. 2 is a preliminary tool for understanding data distribution and patterns, aiding in the identification of outliers. In addition to the polar scatter plot, the power curve of the wind turbine depicted in Fig. 3 is instrumental in detecting outliers and anomalies. This curve illustrates the turbine's nominal behavior and performance [36]. Fig. 3 reveals several points, marked in red, which diverge from the typical output power versus wind speed distribution, indicating potential outliers. These points will be further scrutinized in our data cleaning process.

B. OUTLIER REMOVAL BY SOM

Here, SOM [37] is utilized as a machine learning tool for detecting and removing outliers. To this end, an SOM grid is created and randomly initialized with a suitable number of neurons for the size of the input data. The SOM is then trained using the wind turbine data, where the weight vectors are iteratively updated to minimize the distance between the input data and corresponding neurons. The latter task is done by the update rule

$$w_j(t+1) = w_j(t) + \eta(t)h(j,t)(x(t) - w_j(t)),$$

where $w_j(t)$ is the weight vector for neuron j at time t, $\eta(t)$ is the learning rate at time t, $h(j,t)$ is the neighborhood function that determines the influence of the input vector on the weight vector of neuron j at time t, and $x(t)$ is the wind turbine input data vector at time t, corresponding to the sensor readings of wind speed and power output. $h(j,t)$ is defined in terms of a Gaussian function

$$h(j,t) = \exp\left(-\frac{|r_j - r_i|^2}{2\sigma^2(t)}\right),$$

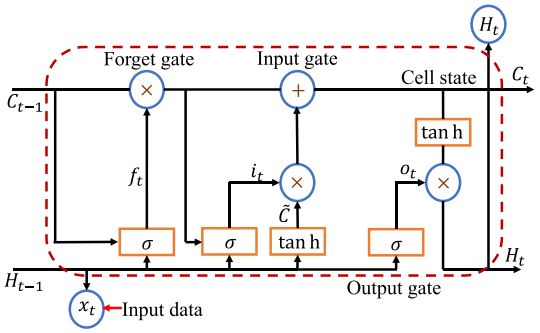


FIGURE 4. Single layer LSTM cell.

where r_j is the location of neuron j in the SOM grid, r_i is the location of the input vector $x(t)$, and $\sigma(t)$ is the neighborhood width at time t. The neighborhood width decreases during the training so as to gradually focus the SOM on the dominant patterns in the input data. Once the SOM has been trained, outliers in the wind turbine data are identified by calculating the distance between each wind turbine data point and its corresponding neuron weight vector described by the distance metric equation

$$d(x, w_j) = |x - w_j|.$$

Data points far away from their corresponding neuron weight vectors are likely to be outliers and are removed from the wind data. The remaining data points will be used for further modeling procedures.

C. ANOMALY DETECTION AND REMOVAL BY LSTM-AE

The LSTM-AE, a neural network architecture combining AE with LSTM networks [38], serves to detect and remove anomalies in the wind turbine SCADA data. Initially, wind data undergoes preprocessing including normalization and division into training and testing sets. Trained on this data, the LSTM-AE generates a reconstructed output. Anomalies are identified by comparing the input with its reconstruction, with significant discrepancies indicating potential anomalies. These anomalies are corrected in the dataset by substituting them with the LSTM-AE's reconstructed values.

Illustrated in Fig. 4, the LSTM unit has three gates—forget, input, and output. First, the forget gate decides which information to forget from the previous cell state by considering both the previous hidden state and new sensor readings. It incorporates weights and biases into the sigmoid function and generates the forget gate value f_t at time t such that

$$f_t = \sigma(W_f[H_{t-1}; x_t] + b_f),$$

where σ is the sigmoid activation function, W_f and b_f are the weight and bias of the forget gate, H_{t-1} is the previous hidden state, and x_t is the current input.

Next, the input gate is responsible for selecting which new information to store in the LSTM network's memory and how much of them to update. This task is important in wind data analysis since it implies which of new wind speed and power measurements are to be selected and stored. It is achieved by generating the candidate memory updated value $\tilde{C}_t$ and the input gate value i_t. $\tilde{C}_t$ has the form of

$$\tilde{C}_t = \tanh(W_c[H_{t-1}; x_t] + b_c),$$

where W_c and b_c are the weight and bias of the input gate. Further, i_t is defined as

$$i_t = \sigma(W_i[H_{t-1}; x_t] + b_i),$$

where W_i and b_i are the weight and bias of the input gate. The cell state C_t is then updated using f_t, $\tilde{C}_t$, and i_t as

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t,$$

where '$\odot$' denotes element-wise multiplication.

Finally, the output gate determines which information from the long-term memory is relevant to the current state of the wind data. The output gate value o_t is given by

$$o_t = \sigma(W_o[H_{t-1}; x_t] + b_o),$$

where W_o and b_o are the weight and bias of the output gate. The new hidden state H_t is induced by applying o_t to a squished version of the cell state C_t as follows.

$$H_t = o_t \odot \tanh(C_t).$$

This process is repeated for all time steps.

An AE includes the input layer, the output layer, and multiple hidden layers. Fig. 5 shows the schematic diagram of the LSTM-AE architecture for anomaly detection and removal. Let us describe each component of the architecture.

1) INPUT SEQUENCE DATA

The wind data is provided as a time sequence $[S_1, S_2, \ldots; S_n]$. Each S_i is a fixed-length time window data $[s_1, s_2, \ldots; s_t]$ where $s_t \in \mathbb{R}^m$ represents m features given at time t. This sequence is reshaped into a two-dimensional (2D) array where each row and column represents a sample and time step, respectively.

2) LSTM ENCODER AND DECODER

The LSTM encoder contains a layer with 10 LSTM cells. We first split the dataset of wind speed and output power into the sequences of 10 samples, with each sequence representing a fixed time window of 100 minutes. These sequences are reshaped into a 2D array having 10 time-steps and two features of the wind turbine. The 2D arrays are then fed to the LSTM encoder. The output from the last LSTM cell will be a 1×16 encoded feature vector containing the information about all the relevant samples in the sequence. The encoded feature vector provided by the LSTM encoder is passed to the LSTM decoder for sequence reconstruction. The decoder contains an LSTM layer with 10 cells. The output from the decoder is reshaped into a 2D array of the same shape as the input to the encoder to obtain the reconstructed sequence.

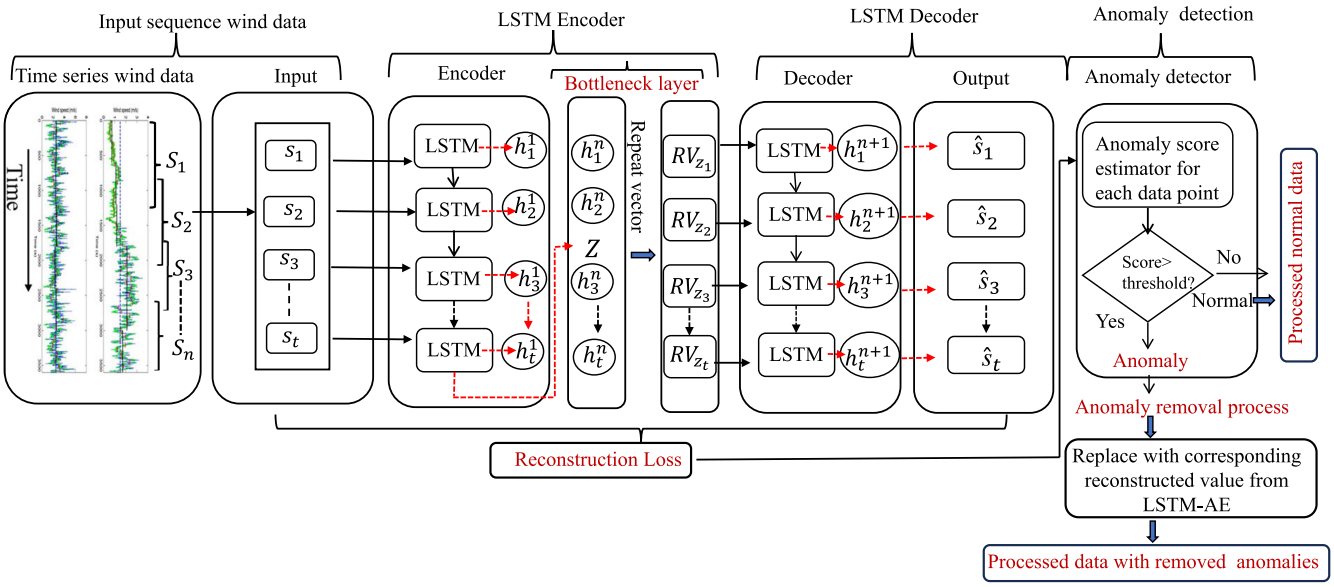


FIGURE 5. LSTM-AE architecture for wind data anomaly detection and removal.

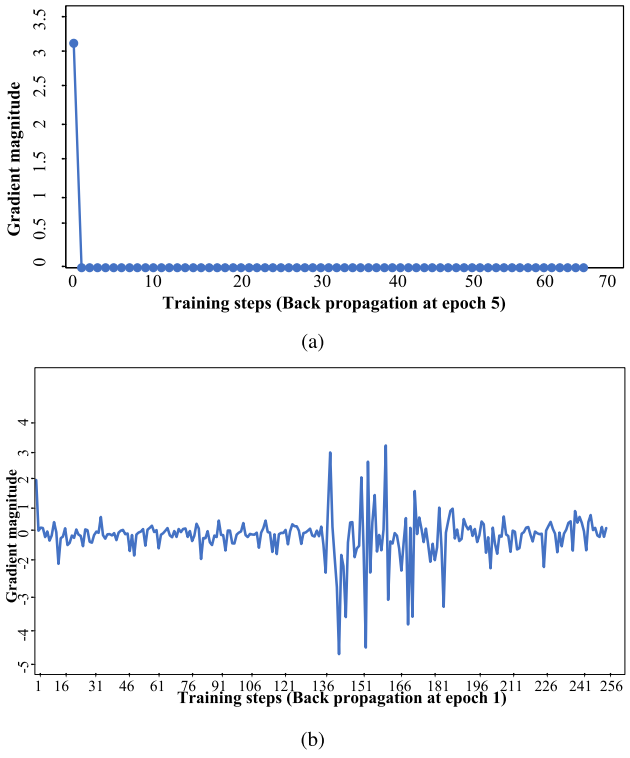


FIGURE 6. Impact of vanishing gradients: (a) stand-alone RNN (epoch 5) and (b) stand-alone LSTM (epoch 1).

3) ANOMALY DETECTION

We then compare the original input sequence s with the reconstructed one $\hat{s}$. If the difference between s and $\hat{s}$, termed the reconstruction loss, exceeds a prescribed threshold, we flag the corresponding time window as anomalous. In this study, the reconstruction loss is defined as the average absolute difference between the corresponding elements of s and $\hat{s}$, i.e., (n is the number of samples)

$$L(s-\hat{s}) = \frac{1}{n}\sum_{t=1}^{n} |\hat{s}_t - s_t|.$$

4) ANOMALY REMOVAL

Data points marked as anomalies due to significant reconstruction loss are replaced with values regenerated by the LSTM-AE. Rather than discarding anomalies, which would potentially cause data gaps or analysis distortions, the LSTM-AE infers normal conditions from its learned data representation, producing fitting substitutes. This precise correction by the LSTM-AE eradicates disruptive effects of anomalies, while enhancing the dataset's relevance to the operational conditions of the turbine.

Remark 1: Note that our outlier and anomaly removal scheme integrating SOM with LSTM-AE overcomes the limitations of the stand-alone LSTM-AE model, which records MAE of 0.0573 in wind speed prediction [34]. It can be said that SOM's strength in dimensionality reduction [35] effectively complements the anomaly detection ability of LSTM-AE. The combined approach significantly improves data processing in the proposed RNN-LSTM model, resulting in a lowered MAE of 0.049, a substantial improvement over the stand-alone SOM and LSTM-AE models.

D. RNN-LSTM HYBRID MODEL

In a stand-alone RNN, the hidden state update equation plays a crucial role in capturing temporal dependencies. This is particularly important when dealing with time series data, e.g., wind turbine data as in this study, where the sequence of past observations influences the future predictions. The

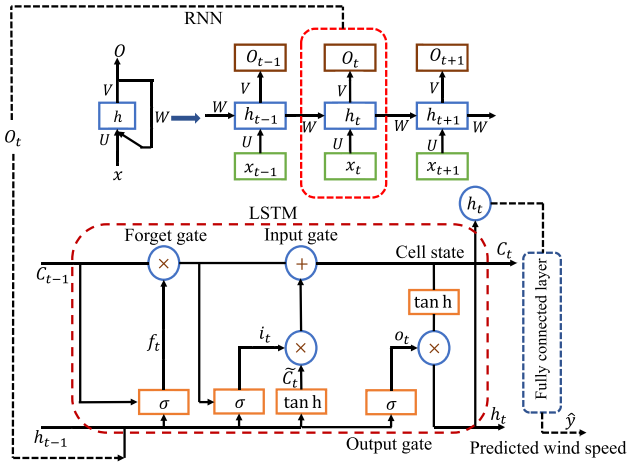


FIGURE 7. Integrated structure of RNN-LSTM.

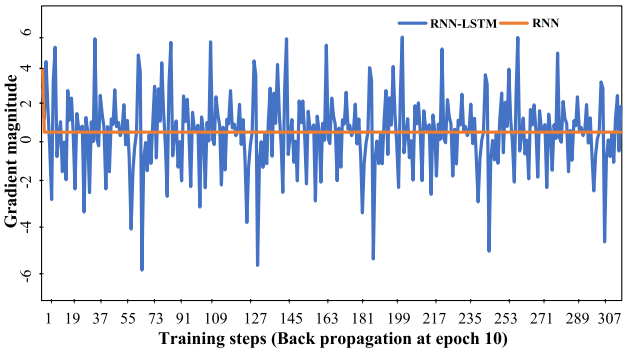


FIGURE 8. Comparison of gradient magnitudes between stand-alone RNN and RNN-LSTM hybrid (epoch 10).

equation for updating the hidden state at time t is given by

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h), \tag{1}$$

where h_t is the hidden state at time t, W_{hh} and W_{xh} are weight matrices, and b_h is the bias vector. By taking into account the previous hidden state h_{t-1} and the current input x_t, the RNN captures the sequential dependencies and generates predictions based on the wind turbine time series data.

Stand-alone RNNs have a drawback termed the vanishing gradient problem [39] in which diminishing gradient flows impede accurate learning performance. The sigmoid activation function with low maximum derivative value 0.25 is known to cause the vanishing gradient problem since it gives a dampening effect on gradients during backpropagation. Stand-alone LSTM models also encounter the issue of vanishing or exploding gradients. This problem invokes instability in gradient updates, hindering learning steps [40].

For the sake of quantitative analysis, we obtain the gradient flow of stand-alone RNN and LSTM by training each model with respect to the wind turbine SCADA data. One can observe from Fig. 6(a) that while the initial gradient magnitude of the stand-alone RNN is relatively high, it exponentially diminishes and declines to zero beyond 65 training steps. On the other hand, the gradient flow of the stand-alone LSTM in Fig. 6(b) shows radically re-surging fluctuations (136 to 181 time steps) after initial diminishing. After the duration, however, it diminishes again like the case of the stand-alone RNN. Fig. 6(b) corroborates that not only stand-alone RNNs but also stand-alone LSTMs suffer the vanishing gradient problem, albeit with lesser degree.

Fig. 7 illustrates the integrated structure of RNN-LSTM. The RNN layer computes the hidden state by (1), and the output of the RNN layer, denoted by O_t, is fed as the input to the LSTM layer. As already mentioned, i_t, f_t, o_t and C_t determine the amount of information to retain or discard from the new input and previous state, as well as the amount of the cell state to the output. The LSTM update equations for these gates are now defined as

$$\begin{aligned} i_t &= \sigma(W_{xi}O_t + W_{hi}h_{t-1} + W_{ci}C_{t-1} + b_i), \\ f_t &= \sigma(W_{xf}O_t + W_{hf}h_{t-1} + W_{cf}C_{t-1} + b_f), \\ o_t &= \sigma(W_{xo}O_t + W_{ho}h_{t-1} + W_{co}C_t + b_o), \end{aligned}$$

where W_{xi}, W_{hi}, W_{ci}, b_i, W_{xf}, W_{hf}, W_{cf}, b_f, W_{xo}, W_{ho}, W_{co}, and b_o are the weight matrices. In association with the above equations, the cell state C_t is updated by

$$C_t = f_t C_{t-1} + i_t \tanh(W_{xc}O_t + W_{hc}h_{t-1} + b_c),$$

and the hidden state h_t is by

$$h_t = o_t \tanh(C_t).$$

Finally, the predicted wind speed is generated by passing h_t through a fully connected layer having a single output node and linear activation function described by

$$\hat{y} = W_{hy}h_T + b_y,$$

where W_{hy} and b_y are the weight matrix and bias vector for the output layer, and $\hat{y}$ is the vector denoting the predicted wind speed and output power.

The proposed RNN-LSTM hybrid model has a significant advantage over stand-alone RNN or LSTM in that it reduces the vanishing gradient problem. In the course of backpropagation, f_t selectively forgets the information delivered from the previous cell state by providing values between 0 and 1. If $f_t \approx 0$, the previous cell state is ignored; if $f_t \approx 1$, it is retained. On the other hand, i_t selectively adds new information to the cell state by providing values between 0 and 1. When $i_t \approx 0$, the new information is ignored; when $i_t \approx 1$, it is retained. Finally, o_t selectively gives the output information from the cell state by yielding values between 0 and 1. With $o_t \approx 0$, the cell state is ignored and with $o_t \approx 0$, the cell state is used to compute the output. In this way, the LSTM layer selectively retains or discards the information, thereby alleviating the vanishing gradient problem.

To validate the performance of reducing vanishing gradients, we plot in Fig. 8 gradient magnitudes of the stand-alone RNN and RNN-LSTM hybrid model with respect to wind turbine SCADA data. In stand-alone RNN, the gradient

magnitude starts high but rapidly declines to almost zero, indicating vanishing gradients. By contrast, the RNN-LSTM hybrid model maintains larger gradient magnitudes with notable peaks and troughs. This confirms better preservation and reduced decay of gradients in the RNN-LSTM model.

IV. FAULT DETECTION

A. EVALUATION METRICS

To develop the fault detection scheme based on the trained RNN-LSTM hybrid model, we utilize R^2 and MAE metrics. R^2 measures the goodness of fit of the model to the data, with values closer to 1 indicating a better fit [41]. Given total number of observation n, R^2 is defined as

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

where $y_i(t)$ and $\hat{y}_i(t)$ are the ith actual and predicted value and $\bar{y}(t)$ is the mean of the actual values. On the other hand, MAE quantifies the average absolute difference between the predicted and actual values, i.e.,

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i(t) - \hat{y}_i(t)\right|.$$

In addition to these metrics, we employ a control chart based on the moving range to assess the deviations between measured and predicted values. The UCL and LCL in the control chart indicate the acceptable boundaries of deviation from the predicted values. Given a value d_i at index i in a sequence and its previous value d_{i-1}, the standard deviation σ and UCL and LCL are

$$\sigma = \frac{|d_i - d_{i-1}|}{1.128}$$
$$\text{UCL} = 3\sigma, \quad \text{LCL} = -3\sigma.$$

B. BLADED DATA AND COMPARISON TARGETS

We utilize Bladed simulator data from a 5 MW wind turbine to validate the fault detection capability of our hybrid model. Although the original Bladed data contains no faults, we introduce synthetic faults between 163 and 210 s in which output power and generator speed surpass their normal values, setting the stage for real-time fault detection assessment. For comparison, we evaluate our RNN-LSTM model against three established methods XGBoost [42], random forest regressor (RFR) [43], and CIL [25] which, enhancing traditional LSTM, incorporates MAE and MSE to capture temporal changes.

The training process of XGBoost involves minimizing an objective function with regularization. The objective function $J(t)$ consists of the loss function and regularization term. The loss function, denoted by $l(y_i(t), \hat{y}_i(t))$, measures the difference between the predicted value and the actual value for each training sample. Let $\Omega(R_k)$ denote the regularization applied to an individual tree where R_k is an individual tree in the ensemble. Then, $J(t)$ is expressed as

$$J(t) = \sum_{i=1}^{n} l(y_i(t), \hat{y}_i(t)) + \sum_{k=1}^{K}\Omega(R_k),$$
$$\Omega(R_k) = \gamma T + 0.5\lambda\sum v^2,$$

where K is the total number of trees in the ensemble, T is the number of leaves in a decision tree, v is the complexity and vector scores of each leaf, γ is the regularization parameter for complexity, and λ is the regularization parameters. The XGBoost model takes wind speed and output power observations as the input pair and passes them through each decision tree in the ensemble. Each decision tree then produces a prediction for wind speed and output power. The final prediction $\tilde{y}_{\text{XGBoost}}(t)$ at time t is obtained by summing the predictions of all the trees as described by

$$\tilde{y}_{\text{XGBoost}}(t) = \sum_{k=1}^{K} G_k(x(t)),$$

where $x(t)$ is the input data including time series information on wind speed and output power observations, and $G_k(x(t))$ denotes the prediction made by the kth tree for $x(t)$.

RFR is an ensemble machine-learning algorithm that combines the predictions of multiple independent decision trees to make accurate predictions. Here, each decision tree in the ensemble is trained on a random subset of wind speed and output power data. The final prediction $\tilde{y}_{\text{RFR}}(t)$ at time t is obtained by averaging the predictions of all the trees, i.e.,

$$\tilde{y}_{\text{RFR}}(t) = \frac{1}{K}\sum_{k=1}^{K} H_k(x(t)),$$

where $H_k(x(t))$ represents the prediction made by the kth tree of RFR for $x(t)$.

The core of CIL's approach is a hybrid function $Z_{\text{CIL}}(t)$ that maximizes temporal accuracy in time series predictions. In particular, it synthesizes MAE and MSE in the form of

$$Z_{\text{CIL}}(t) = \alpha \times \text{MAE} + \beta \times \text{MSE},$$

where α and β are coefficients that determine the balance between the contributions of MAE and MSE. A defining characteristic of CIL is its bespoke transformation mechanism to circumvent challenges caused by the supersaturation zone. Upon data intake, CIL processes temporal metrics, including but not limited to measures like wind speed and power output. This data undergoes interpretation by LSTM layers. The culmination of this process is the prediction represented by

$$\hat{y}_{\text{CIL}}(t) = F_{\text{LSTM}}(x(t), \theta),$$

where $x(t)$ is the specific input metrics, θ is model parameters, and F_{LSTM} is the prediction methodology of CIL.

Remark 2: While the integration of RNN and LSTM for wind turbine fault prediction enhances time series data analysis, it may encounter challenges in terms of great computational demands depending on the data to be

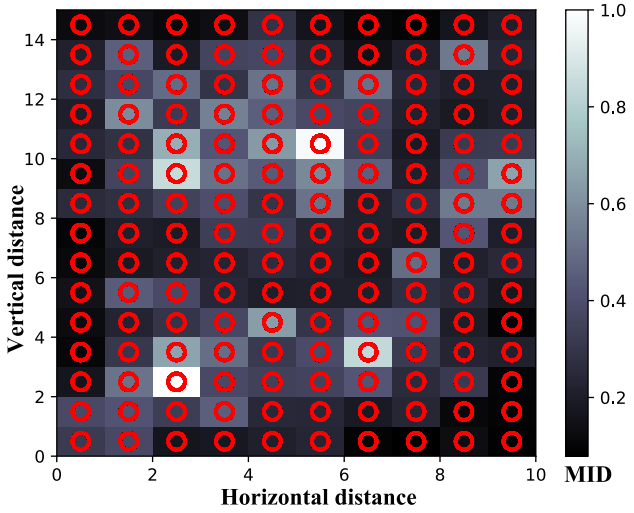


FIGURE 9. SOM on 2D grid.

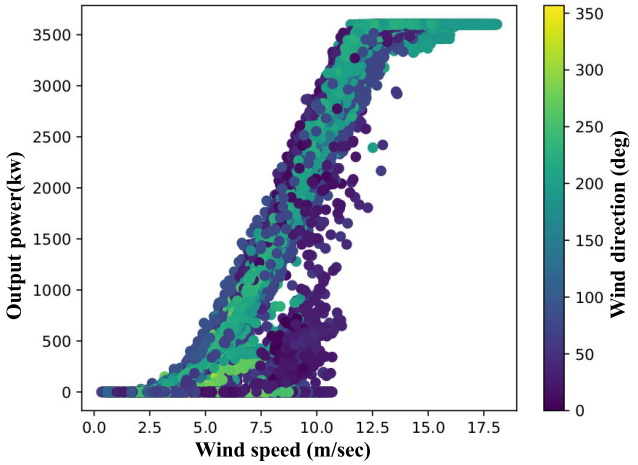


FIGURE 10. Power curve without outliers.

TABLE 1. Summary of the resultant trained data set in LSTM-AE.

Layer Type	Output Shape	Parameters
input_2 (InputLayer)	(None, 1, 1)	0
lstm_4 (LSTM)	(None, 1, 64)	16,896
lstm_5 (LSTM)	(None, 32)	12,416
repeat_vector_1 (RepeatVector)	(None, 1, 32)	0
lstm_6 (LSTM)	(None, 1, 32)	8,320
lstm_7 (LSTM)	(None, 1, 64)	24,832
time_distributed_1 (TimeDistributed)	(None, 1, 1)	65
Total parameters	-	62,529
Trainable parameters	-	62,529
Non-trainable parameters	-	0
Anomalies detected	(504)	-

processed. Nevertheless, the present study is worthwhile and may give a broad impact on fault prediction since it succeeds in balancing between fault detection accuracy and computational efficiency by combining RNN and LSTM.

V. NUMERICAL EXPERIMENTS

A. OUTLIERS AND ANOMALY REMOVAL

We implement an outlier removal module using the MiniSom library in Python 3.8, along with TensorFlow version 2.10.

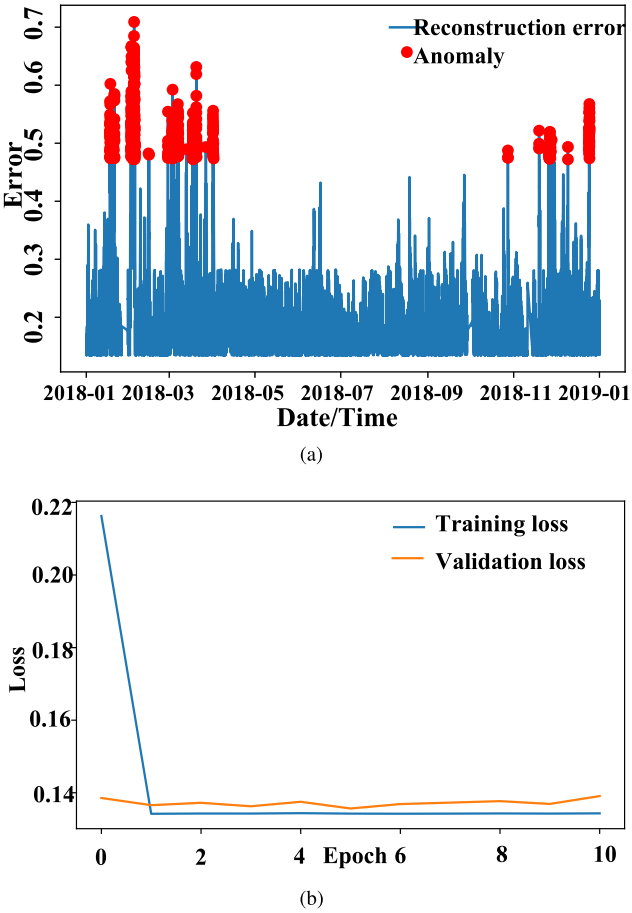


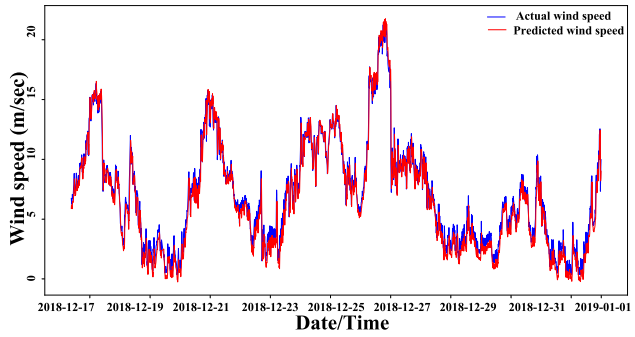
FIGURE 11. (a) Anomalies in wind speed data and reconstruction error and (b) training and validation loss visualization.

TABLE 2. LSTM hidden layer configuration in LSTM-AE.

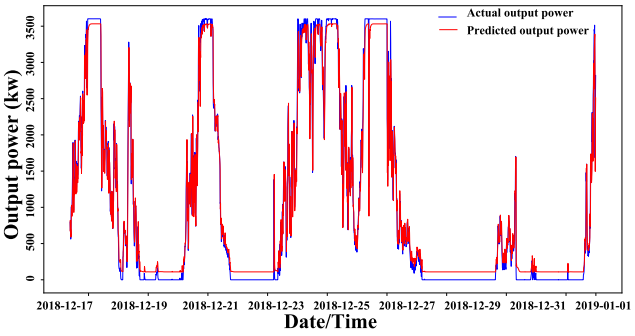
Hyper-Parameters	Used Values
Number of hidden LSTM layers	3
Activation function of output layer	Linear activation function
Loss function	MAE
Optimizer	ADAM
Batch size	32
Epochs	100

The SOM, configured with a 10 × 10 neuron grid, a neighborhood radius of 1.0, and a learning rate of 0.5, is trained over 100 iterations with random input samples. This process involves updating neuron weights iteratively to identify the best matching units (BMUs) based on Euclidean distance. The training results, visualized in Fig. 9, demonstrate the mean interneuron distance (MID) across the SOM grid. Regions with cohesive clusters, indicated by darker colors (smaller MID), confirm successful data organization based on underlying patterns. Conversely, lighter colors signal greater dissimilarity, highlighting potential outliers. From this analysis, we identify and remove 224 outliers.

To assess the impact of outlier removal on the power curve, we compare the original Fig. 3, and refined Fig. 10 power curves. This comparison reveals a significant improvement

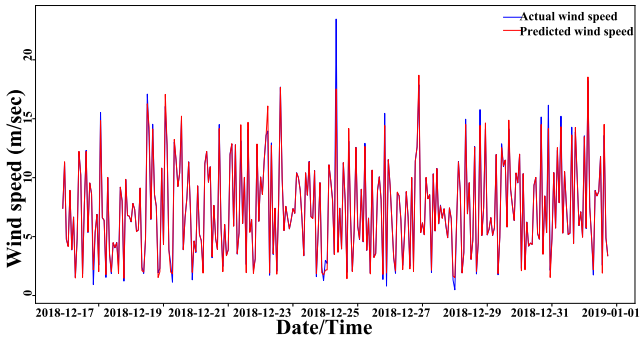


(a)

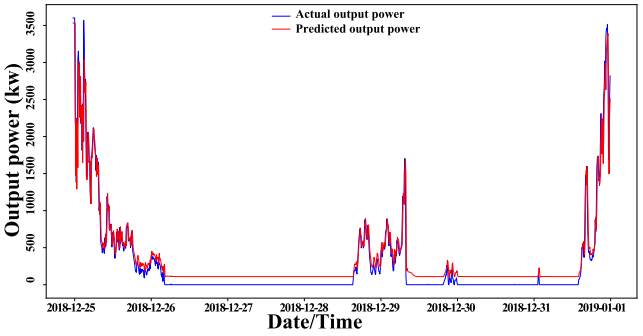


(b)

FIGURE 12. Comparison of actual and predicted wind speed and output power for the SCADA 3.5 MW wind turbine data using RNN-LSTM: (a) actual and predicted wind speed and (b) actual and predicted output power.



(a)



(b)

FIGURE 13. Comparison of actual and predicted wind speed and output power for the SCADA 3.5 MW wind turbine data using XGBoost: (a) actual and predicted wind speed and (b) actual and predicted output power.

in data accuracy, validating the enhanced performance of the RNN-LSTM hybrid model.

Next, we employ LSTM-AE for anomaly detection and removal in the outlier-free wind data. Our LSTM-AE comprises two encoder layers with 64 and 32 units, respectively. The decoder consists of three LSTM layers, with the first layer expanding the compressed data, followed by two layers with 64 and 32 units, and a final dense layer reconstructing wind speed and output power. The AE is trained to minimize the MAE between the input and reconstructed sequences, using the Adam optimizer with a batch size of 32 across 100 epochs. Anomalies in the wind data are identified by comparing reconstruction errors against a threshold set at the 99th percentile of error distribution. Time steps exceeding this threshold are marked as anomalies. Details of the LSTM-AE configuration and trained dataset are summarized in Tables 1 and 2.

Fig. 11(a) displays the detected anomalies in the wind turbine data over time, pinpointing deviations in wind speed and output power from expected trends. The validation plot in Fig. 11(b) affirms the LSTM-AE model's ability to reconstruct input data while avoiding overfitting. It turns out that the LSTM-AE model measures MAE of 0.66 and R^2 of 0.87 in anomaly detection. These performance metrics confirm the model's capability to differentiate between normal and abnormal data patterns, validating its effectiveness in anomaly detection for wind turbine data.

TABLE 3. Learning performances comparison with respect to SCADA 3.5 MW wind turbine data.

Parameter	Model Type	R^2	MAE
Wind speed	RNN-LSTM	0.9889	0.049
	XGBoost	0.94	0.630
	RFR	0.95	0.45
	CIL	0.9667	0.6053
Output power	RNN-LSTM	0.9764	0.538
	XGBoost	0.9212	0.8534
	RFR	0.955	0.96004
	CIL	0.9758	0.654

B. IMPLEMENTATION OF RNN-LSTM HYBRID MODEL

In this experiment, feature scaling via Min-MaxScaler from scikit-learn is used to normalize data for the RNN-LSTM hybrid model. The model comprises four RNN layers and four LSTM layers, each with 50 units and a dropout rate of 0.2, leading to a final output layer with one unit. It is compiled using the Adam optimizer and MSE function, and trained for 100 epochs with a batch size of 32, monitoring MSE loss for training and validation.

The model's performance, depicted in Fig. 12, shows high accuracy in predicting wind speed and output power. For wind speed, it attains R^2 score of 0.9889, and MAE of 0.049, indicating minimal deviation from actual values. Output power predictions yield R^2 score of 0.9764, with a slightly higher MAE of 0.538. These results as summarized in Table 3 underscore the model's proficiency in accurately estimating both wind speed and output power.

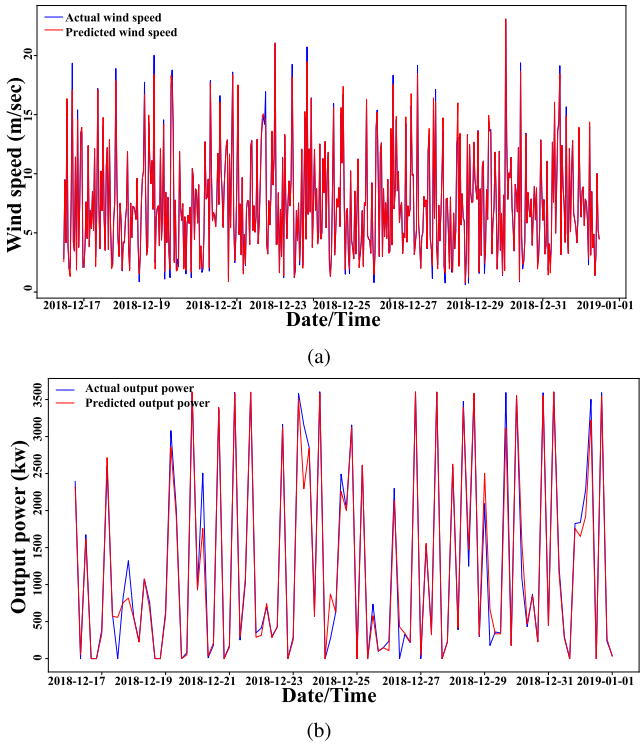


FIGURE 14. Comparison of actual and predicted wind speed and output power for the SCADA 3.5 MW wind turbine data using RFR: (a) actual and predicted wind speed and (b) actual and predicted output power.

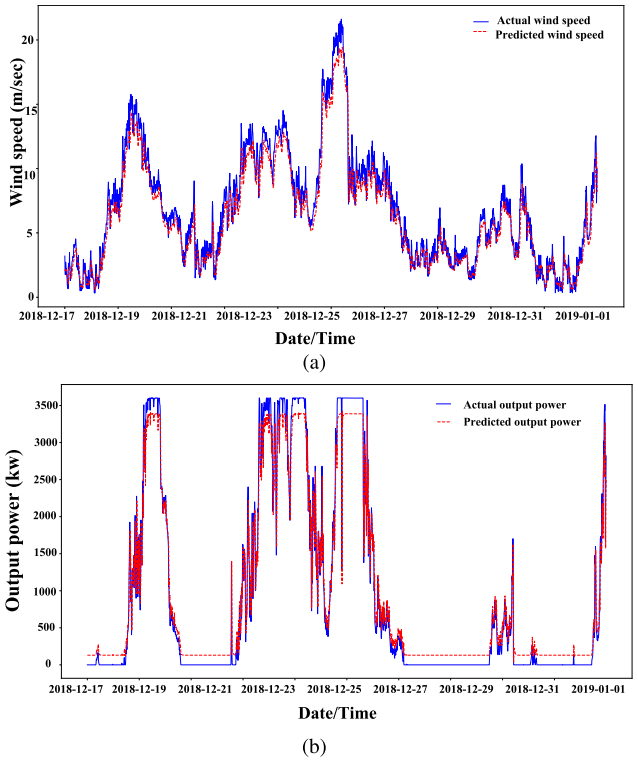


FIGURE 15. Comparison of actual and predicted wind speed and output power for the SCADA 3.5 MW wind turbine data using CIL: (a) actual and predicted wind speed and (b) actual and predicted output power.

C. IMPLEMENTATION OF XGBOOST, RFR, AND CIL

We implement the XGBoost regressor with alpha and lambda regularization parameters on SCADA 3.5 MW wind turbine data, using L1 and L2 regularization to prevent overfitting. The learning performance evaluated by R^2 and MAE is summarized in Table 3 and depicted in Fig. 13. For wind speed, XGBoost achieves $R^2 = 0.94$ and MAE $= 0.630$, while for output power, it shows $R^2 = 0.9212$ with a higher MAE of 0.8534.

The RFR model, employing an ensemble of decision trees, demonstrates a strong fit on the training data. As shown in Table 3 and illustrated in Fig. 14, it achieves $R^2 = 0.95$ and MAE $= 0.45$ for wind speed, and $R^2 = 0.955$ with an MAE of 0.96 for output power.

CIL, which effectively balances MAE and MSE, aligns closely with actual measurements as illustrated in Fig. 15. It exhibits strong predictive capabilities for wind speed ($R^2 = 0.9667$, MAE $= 0.6053$) and output power ($R^2 = 0.9758$, MAE $= 0.654$) as tabulated in Table 3.

It is evident that the RNN-LSTM hybrid model outperforms all of the XGBoost, RFR, and CIL model in terms of prediction accuracy. While the CIL model shows robust predictions with its R^2 and MAE metrics in view of Table 3, it registers a loss value of 0.0274. The RNN-LSTM model, by contrast, demonstrates a remarkable loss of 0.0133 in its initial epoch of training. This substantial difference underscores RNN-LSTM's enhanced prediction accuracy and computational prowess during training phases. Overall, the RNN-LSTM model achieves the highest R^2 and the lowest MAE, indicating a strong correlation between the predicted and actual values of wind speed and output power. These quantitative analyses prove that the RNN-LSTM hybrid model is the most suitable choice among four models for accurate wind speed and output power predictions in wind turbine environments. This finding will precipitate accurate fault detection to wind turbines.

TABLE 4. Training performance comparison with respect to Bladed 5 MW data.

Parameter	Model Type	R^2	MAE
Generator speed	RNN-LSTM	0.96470	0.70542
	XGBoost	0.8235	1.03274
	RFR	0.95434	0.95004
	CIL	0.8048	0.8737
Wind speed	RNN-LSTM	0.92902	0.8485
	XGBoost	0.8757	2.5245
	RFR	0.847	2.01264
	CIL	0.9106	0.8236
Output power	RNN-LSTM	0.95496	0.84791
	XGBoost	0.93405	0.83224
	RFR	0.91312	1.21264
	CIL	0.9708	0.8010

D. COMPARATIVE STUDY WITH BLADED 5 MW WIND TURBINE SIMULATOR

For evaluating the performance of fault detection, we apply the RNN-LSTM hybrid model along with XGBoost, RFR, and CIL to tackling the Bladed simulator 5 MW wind turbine data containing faults. To ensure accurate analysis,

TABLE 5. Evaluating model complexity and computational cost in the proposed integrated RNN-LSTM versus CIL.

Parameter	Model type	3.5 MW SCADA data				5 MW Bladed simulator data			
		Training time (sec)	Inference time (sec)	Number of parameters	Peak memory usage (MiB)	Training time (sec)	Inference time (sec)	Number of parameters	Peak memory usage (MiB)
Wind speed	RNN-LSTM	54.99	0.61	71051	607.2	58.27	1.23	71051	596.89
	CIL	250.27	3.37	461441	645.71	137.73	2.44	461441	639.4
Output power	RNN-LSTM	56.83	0.68	71051	846.02	56.2	1.25	71051	601.9
	CIL	189.54	3.7	461441	647.33	279.85	3.22	461441	644.22

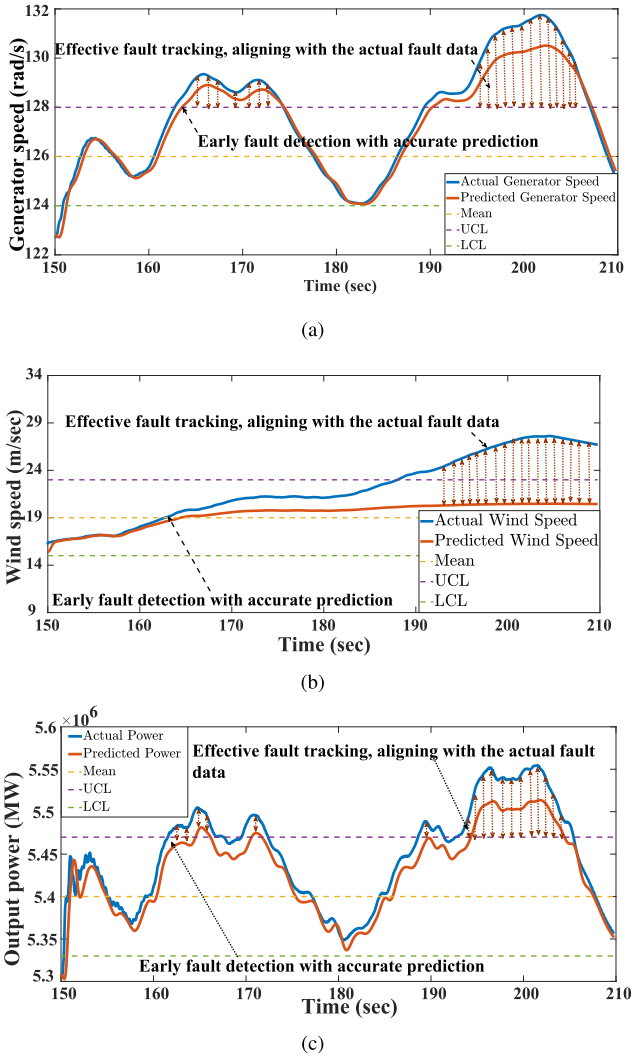


FIGURE 16. Actual and predicted generator speed, wind speed, and output power of the Bladed 5 MW wind turbine simulator and RNN-LSTM hybrid model: (a) generator speed, (b) wind speed, and (c) output power.

we focus our concern on the time period between 10 and 210 s of the simulation in which a deliberate fault is set to occur from 163 to 210 s as mentioned earlier. In real-world scenarios, dummy faults can trigger control system faults, disrupting wind turbine operation, leading to imbalances in power generation and significant fluctuations in generator speed and output power. To facilitate training and testing, we divide the selected dataset into two portions: 70% for training and 30% for testing purposes. Once the training phase is completed, each method is incorporated into the Bladed simulator data, while emulating the wind turbine model with the aforementioned three parameters.

Table 4 elucidates the training performances of four distinct methodologies. A careful inspection reveals the competitive nature of the RNN-LSTM hybrid model. For the generator speed parameter, the RNN-LSTM model shines with an impressive R^2 value of 0.96470, suggesting its capability to elucidate 96.47% of the variability within the dataset. Its MAE of 0.70542 also stands as testimony to its prediction prowess. It is worth noting the commendable performance of the CIL, achieving an R^2 of 0.8048 and MAE of 0.8737, which indicates its effectiveness despite being marginally outperformed by RNN-LSTM. Delving into wind speed, the RNN-LSTM model yet again sets a benchmark with $R^2 = 0.92902$ and MAE $= 0.8485$. As per the output power parameter, on the other hand, the performance of CIL is slightly better than that of RNN-LSTM model as it exhibits the highest R^2 of 0.9708 and lowest MAE of 0.8010. This suggests that, in certain contexts, the CIL may offer incidental advantages over its counterparts.

Table 5 indicates that the RNN-LSTM model offers a more computationally efficient alternative to other complex architectures such as CIL. With reduced training and inference times by up to 75% and 80%, respectively, the RNN-LSTM model demonstrates its potential for rapid deployment and real-time analytical performance. The model's reduced parameter number, a mere 15.4 % that of the CIL model, not only simplifies the computational process but also mitigates the risk of overfitting, thereby enhancing the model's predictive reliability. Moreover, the lower memory footprint, averaging around 10% less across various tests, ensures that the RNN-LSTM model can be implemented within resource-constrained environments.

Fig. 16 shows a visual comparison between the actual and predicted values of generator speed, wind speed, and output power of the Bladed 5 MW wind turbine simulator and the proposed RNN-LSTM hybrid model. At about 163 s, the actual parameters exceed their respective UCL, indicating a potential fault. Remarkably, our prediction algorithm accurately captures this deviation and closely follows the actual values during this period, demonstrating its ability to respond to abrupt changes in the targeted parameters. Moreover, the main impact of the fault is observed between 190 and 210 s, where a significant peak in the actual

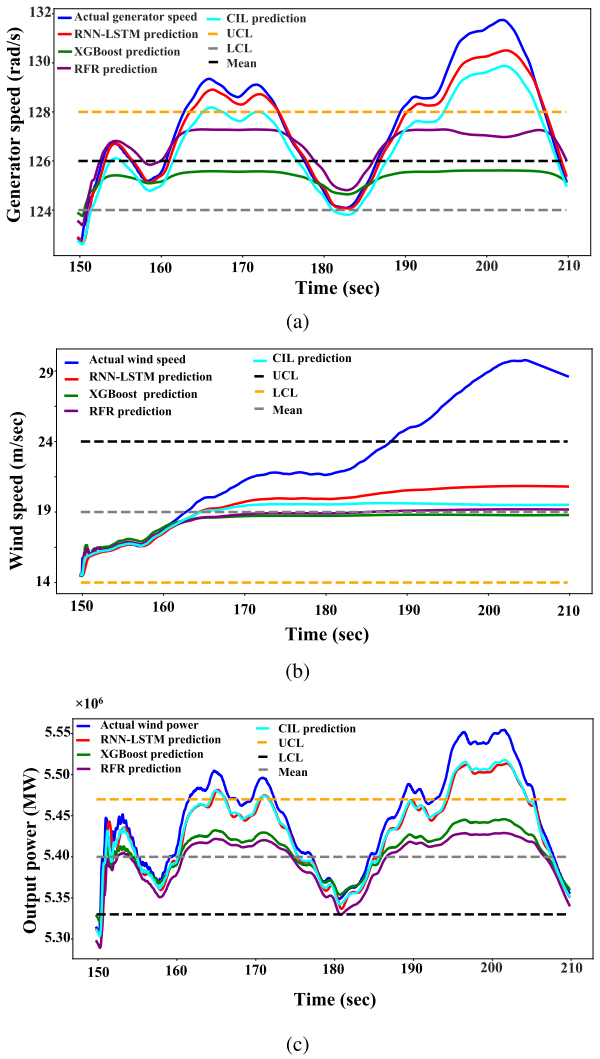


FIGURE 17. Comparison of actual and predicted generator speed, wind speed, and output power between RNN-LSTM, XGBoost, RFR, and CIL: (a) generator speed, (b) wind speed, and (c) output power.

values is evident. Our prediction algorithm also tracks this peak accurately, predicting the main impact of the fault within the short-term time frame. These results highlight the effectiveness of the proposed algorithm in short-term fault prediction for wind turbines. This early detection capability, namely, to accurately capture the fault's main impact at 190 s and to closely track the values shortly before the fault at 163 s, would play a crucial role in preventing significant damages to various wind turbine components, including high-rated generators, gearboxes, and blades.

Fig. 17 provides a comprehensive comparison of the actual and predicted values of generator speed, wind speed, and output power of the Bladed 5 MW wind turbine simulator and the RNN-LSTM, XGBoost, RFR, and CIL models. XGBoost and RFR show variations in their performance, with each model outperforming the other for specific parameters and vice versa. The CIL model acts as a pivotal benchmark, navigating the performance spectrum between the other methods. On the other hand, the RNN-LSTM hybrid model consistently surpasses not only the CIL but the other models in terms of the overall performance, producing accurate estimated parameter trajectories.

VI. CONCLUSION

This study has presented a promising approach for monitoring and detecting faults in wind turbines using SCADA and Bladed simulator data and machine learning algorithms, specifically, the integrated RNN-LSTM hybrid model. The proposed method turns out to outperform XGBoost, RFR, and CIL methodology in short-term fault detection. We have highlighted the usefulness of the proposed scheme in detecting faults in wind turbines having no failure logs, which can be much beneficial to predictive maintenance for preventing catastrophic damages by virtue of early intervention. The proposed scheme can serve as a more dynamic and cost-effective data-driven maintenance strategy in comparison with current static time-based ones.

It is expected that the proposed learning model and fault detection scheme can be extended to other energy systems. As for high-voltage transformers and solar power plants, especially, one can easily attain the models of their key health indicators, such as temperature, electrical load, solar irradiance, and power output. Not only do their operational features bear resemblance to wind power plants, their parameters have similar or identical characteristics to wind turbines. Hence the proposed integrated RNN-LSTM hybrid model will be able to capture these dynamics with little modification. In this regard, the next study will be devoted to applying the proposed maintenance monitoring scheme to these energy conversion systems.

REFERENCES

[1] S. Woo, J. Park, J. Park, and L. Manuel, "Wind field-based short-term turbine response forecasting by stacked dilated convolutional LSTMs," *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2294–2304, Oct. 2020.

[2] M. Shafiee, "Maintenance logistics organization for offshore wind energy: Current progress and future perspectives," *Renew. Energy*, vol. 77, pp. 182–193, May 2015.

[3] Z. Xu, J. Wei, S. Zhang, Z. Liu, X. Chen, Q. Yan, and J. Guo, "A state-of-the-art review of the vibration and noise of wind turbine drivetrains," *Sustain. Energy Technol. Assessments*, vol. 48, Dec. 2021, Art. no. 101629.

[4] R. Fang, Y. Wang, R. Shang, Y. Liang, L. Wang, and C. Peng, "The ultra-short term power prediction of wind farm considering operational condition of wind turbines," *Int. J. Hydrogen Energy*, vol. 41, no. 35, pp. 15733–15739, Sep. 2016.

[5] G. Helbing and M. Ritter, "Deep learning for fault detection in wind turbines," *Renew. Sustain. Energy Rev.*, vol. 98, pp. 189–198, Dec. 2018.

[6] R. Liu, M. Peng, and X. Xiao, "Ultra-short-term wind power prediction based on multivariate phase space reconstruction and multivariate linear regression," *Energies*, vol. 11, no. 10, Oct. 2018, Art. no. 2763.

[7] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018.

[8] J. Carroll, A. McDonald, I. Dinwoodie, D. Mcmillan, M. Revie, and I. Lazakis, "Availability, operation and maintenance costs of offshore wind turbines with different drive train configurations," *Wind Energy*, vol. 20, no. 2, pp. 361–378, Feb. 2017.

[9] (2023). *Wind Turbine Design Software-Bladed.* Accessed: Oct. 2023. [Online]. Available: https://www.dnv.co.kr

[10] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[13] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[14] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[15] D. Huang, W.-A. Zhang, F. Guo, W. Liu, and X. Shi, "Wavelet packet decomposition-based multiscale CNN for fault diagnosis of wind turbine gearbox," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 443–453, Jan. 2023.
[16] H. Wang, Z. Lei, Y. Liu, J. Peng, and J. Liu, "Echo state network based ensemble approach for wind power forecasting," *Energy Convers. Manage.*, vol. 201, Dec. 2019, Art. no. 112188.
[17] C. Zhang, D. Hu, and T. Yang, "Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and XGBoost," *Rel. Eng. Syst. Saf.*, vol. 222, Jun. 2022, Art. no. 108445.
[18] Y. Cheng, H. Zhu, J. Wu, and X. Shao, "Machine health monitoring using adaptive kernel spectral clustering and deep long short-term memory recurrent neural networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 987–997, Feb. 2019.
[19] A. Kouadri, M. Hajji, M.-F. Harkat, K. Abodayeh, M. Mansouri, H. Nounou, and M. Nounou, "Hidden Markov model based principal component analysis for intelligent fault diagnosis of wind energy converter systems," *Renew. Energy*, vol. 150, pp. 598–606, May 2020.
[20] S. Sun, T. Wang, H. Yang, and F. Chu, "Adversarial representation learning for intelligent condition monitoring of complex machinery," *IEEE Trans. Ind. Electron.*, vol. 70, no. 5, pp. 5255–5265, May 2023.
[21] B. Tang, T. Song, F. Li, and L. Deng, "Fault diagnosis for a wind turbine transmission system based on manifold learning and Shannon wavelet support vector machine," *Renew. Energy*, vol. 62, pp. 1–9, Feb. 2014.
[22] L. Xiang, P. Wang, X. Yang, A. Hu, and H. Su, "Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism," *Measurement*, vol. 175, Apr. 2021, Art. no. 109094.
[23] F. Shahid, A. Zameer, and M. Muneeb, "A novel genetic LSTM model for wind power forecast," *Energy*, vol. 223, May 2021, Art. no. 120069.
[24] D. Cabrera, A. Guamán, S. Zhang, M. Cerrada, R.-V. Sánchez, J. Cevallos, J. Long, and C. Li, "Bayesian approach and time series dimensionality reduction to LSTM-based model-building for fault diagnosis of a reciprocating compressor," *Neurocomputing*, vol. 380, no. 7, pp. 51–66, Mar. 2020.
[25] W. Zheng and J. Hu, "Multivariate time series prediction based on temporal change information learning method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7034–7048, Oct. 2023.
[26] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, Jun. 2005.
[27] B. S. Penn, "Using self-organizing maps to visualize high-dimensional data," *Comput. Geosci.*, vol. 31, no. 5, pp. 531–544, Jun. 2005.
[28] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.
[29] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management," *Int. J. Inf. Manage.*, vol. 57, Apr. 2021, Art. no. 102282.
[30] T. R. McVicar, M. L. Roderick, R. J. Donohue, L. T. Li, T. G. Van Niel, A. Thomas, J. Grieser, D. Jhajharia, Y. Himri, N. M. Mahowald, A. V. Mescherskaya, A. C. Kruger, S. Rehman, and Y. Dinpashoh, "Global review and synthesis of trends in observed terrestrial near-surface wind speeds: Implications for evaporation," *J. Hydrol.*, vols. 416–417, pp. 182–205, Jan. 2012.
[31] P. J. Zucatelli, E. G. S. Nascimento, G. Y. R. Aylas, N. B. P. Souza, Y. K. L. Kitagawa, A. A. B. Santos, A. M. G. Arce, and D. M. Moreira, "Short-term wind speed forecasting in Uruguay using computational intelligence," *Heliyon*, vol. 5, no. 5, May 2019, Art. no. e01664.
[32] S. Li, D. C. Wunsch, E. A. O'Hair, and M. G. Giesselmann, "Using neural networks to estimate wind turbine power generation," *IEEE Trans. Energy Convers.*, vol. 16, no. 3, pp. 276–282, Sep. 2001.
[33] B. Erisen. (2018). *Wind Turbine SCADA Dataset*. [Online]. Available: https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset
[34] H. Chen, H. Liu, X. Chu, Q. Liu, and D. Xue, "Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network," *Renew. Energy*, vol. 172, pp. 829–840, Jul. 2021.
[35] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.
[36] M. Schlechtingen, I. F. Santos, and S. Achiche, "Using data-mining approaches for wind turbine power curve monitoring: A comparative study," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 671–679, Jul. 2013.
[37] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.
[38] Y. Wei, J. Jang-Jaccard, W. Xu, F. Sabrina, S. Camtepe, and M. Boulic, "LSTM-autoencoder-based anomaly detection for indoor air quality time-series data," *IEEE Sens. J.*, vol. 23, no. 4, pp. 3787–3800, Feb. 2023.
[39] F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An optical communication's perspective on machine learning and its applications," *J. Lightw. Technol.*, vol. 37, no. 2, pp. 493–516, Jan. 15, 2019.
[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[41] W. Udo and Y. Muhammad, "Data-driven predictive maintenance of wind turbine based on SCADA data," *IEEE Access*, vol. 9, pp. 162370–162388, 2021.
[42] P. Trizoglou, X. Liu, and Z. Lin, "Fault detection by an ensemble framework of extreme gradient boosting (XGBoost) in the operation of offshore wind turbines," *Renew. Energy*, vol. 179, pp. 945–962, Dec. 2021.
[43] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018.

V SIVA BRAHMAIAH RAMA received the M.Tech. degree in power electronics from VIT University, India, in 2011. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering, Kyungpook National University, Republic of Korea. His research interests include machine learning and deep learning applications in wind turbines and rotating machines.

SUNG-HO HUR (Member, IEEE) received the B.Eng. degree (Hons.) in electronics and electrical engineering (EEE) from the University of Glasgow, U.K., in 2004, and the M.Sc. (Hons.) and Ph.D. degrees in EEE from the University of Strathclyde, U.K., in 2005 and 2010, respectively. He is currently an Associate Professor with the School of Electronics Engineering, Kyungpook National University, Republic of Korea. He was with the Wind Energy and Control Group, University of Strathclyde, U.K. His current research interests include control, condition monitoring, and modeling, with a particular interest in wind turbines and farms.

JUNG-MIN YANG received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, in 1993, 1995, and 1999, respectively. Since 2013, he has been with the School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea, where he is currently a Professor. His research interests include control of asynchronous sequential machines, control of complex networks, and wind energy conversion systems.