

ORIGINAL RESEARCH

Open access

IoT-Enabled Predictive Maintenance for Wind Turbines: Leveraging Machine Learning and Wireless Sensor Networks

Lars Andersen^{1*}, Freya Nielsen¹, Mette Sørensen²

Abstract

Wind power (WP) is a Renewable Energy Source (RES) that has experienced remarkable growth as society continues to seek environmentally friendly energy alternatives. This study primarily explores Predictive Maintenance (PM) of Wind Turbines (WT). It evaluates the potential advantages of Wind Energy (WE) when integrated with the Internet of Things (IoT) and Wireless Sensor Networks (WSN). The research proposes an IoT-WSN framework for PM, structured into three key stages: the initial stage involves data acquisition via sensors; the second stage involves data transmission via an internet connection; and the third stage involves data analytics performed in a cloud computing environment. For PM analytics, this work presents a Predictive Maintenance Convolutional Long Short-Term Memory (PM-C-LSTM) model that integrates the spatial feature-extraction strengths of a Convolutional Neural Network with the time-series modeling capabilities of LSTM networks. The PM-C-LSTM model combines a CNN for spatial pattern detection with LSTM networks for sequential data processing, without compromising the precision of WT-PM. Additionally, a Failure Sample Generator model is incorporated to quantify soft failure and hard failure elements, thereby enhancing the predictive performance of the Machine Learning (ML) approach. The model was deployed on a Wind Farm (WF) on the Qinghai-Tibet Plateau, using data collected over 16 months. Comparative evaluations against several conventional models, based on multiple performance metrics, demonstrate that the PM-C-LSTM model delivers superior PM capabilities. The experimental results suggest that the combined use of IoT and ML has strong potential to advance PM strategies for WT, ultimately contributing to greater efficiency and sustainability in WE production.

Keywords Renewable energy, Machine learning, Wireless sensor networks, Internet of things, Long short-term memory, Accuracy

*Correspondence:

Lars Andersen

laa@energy.aau.dk

¹ Department of Power Systems Analytics, Faculty of Engineering, Aalborg University, Aalborg, Denmark

² Department of Smart Energy Optimization, Faculty of Engineering, University of Copenhagen, Copenhagen, Denmark

Introduction

Wind Energy (WE) is a crucial Renewable Energy Source (RES) for nations striving to address climate change and reduce dependence on fossil fuels. It is harnessed via wind turbines in Wind Farms (WF), where turbines convert the kinetic energy of wind-driven blades into electricity [1]. The expanding WE industry demands effective operation and maintenance strategies to guarantee reliability and optimal output. The Internet of Things (IoT) and Wireless Sensor Networks (WSN) represent intelligent technologies that enable enhanced monitoring and upkeep across various domains [2]. IoT + WSN-enabled monitoring of Wind Turbines (WT) employs sensors to continuously measure parameters such as temperature, vibration, and rotational speed, delivering critical insights into the condition and performance of individual WT units through uninterrupted, 24/7 observation [3].

Maintenance of Wind Farms (WF) is essential for sustained energy production, given their often remote placement, intricate and costly components, and the risk of output losses from unexpected breakdowns [4]. The traditional Reactive Maintenance Approach (RMA) addresses issues only after failure occurs, resulting in high costs and prolonged downtime. In contrast, Predictive Maintenance (PM) offers a proactive strategy that leverages IoT data combined with sophisticated Machine Learning (ML) techniques. This approach facilitates early fault identification before failures escalate, thereby minimizing downtime and extending the service life of WT components [5]. By applying ML to sensor data, hidden patterns and predictive indicators can be uncovered, enabling timely interventions, more precise WT maintenance scheduling, improved WF energy generation efficiency, and greater overall sustainability [6].

The rationale for investigating the integration of ML with IoT + WSN in WT systems stems from the pressing need for reliable PM solutions. WT operates under highly variable and harsh environmental conditions, rendering conventional RMA inadequate. This study seeks to harness ML to strengthen PM using real-time data from IoT and WSNs. The primary goal is to develop a robust model capable of accurately forecasting maintenance requirements, thereby boosting operational performance, reducing unplanned outages, and ultimately advancing WE's sustainability by optimizing the efficiency and durability of WT systems.

The integration of IoT and ML in PM encounters several obstacles, including the processing and analysis of massive data volumes produced by IoT devices in WF, achieving precise predictions, managing the inherent complexity of WT systems, ensuring interoperability among diverse technologies and data formats, and coping with the harsh environmental conditions affecting WT systems [7].

This study introduces an IoT + WSN architecture specifically designed for WT-PM, comprising three main layers: a sensor layer for data acquisition, a gateway layer for data transmission, and a cloud layer for advanced analytics.

At the core of the proposed PM system lies the Predictive Maintenance Convolutional Long Short-Term Memory (PM-C-LSTM) network. This hybrid architecture effectively merges the strengths of CNN and LSTM networks. To further enhance the predictive performance of the PM-C-LSTM model, this research incorporates a Failure Sample Generator (FSG) module that calculates Soft Failure (SF) and Hard Failure (HF) indicators. The proposed model was evaluated on a Wind Farm (WF) on the Qinghai-Tibet Plateau, using data collected from three selected turbines over 16 months from June 2021 to September 2022. The model was applied to the gathered dataset, and its effectiveness was benchmarked against other approaches using a variety of performance metrics. Results demonstrated that the proposed model consistently outperformed the competing models across all evaluated metrics.

The fusion of CNN and LSTM within the PM-C-LSTM framework significantly advances WT-PM by facilitating automatic spatial feature extraction (FE), capturing temporal dependencies, and adapting to fluctuating environmental conditions—capabilities that exceed those of conventional models. The inclusion of the FSG module further strengthens the PM-C-LSTM by enabling the generation of additional failure scenarios for training purposes. The FSG produces Soft Failure (SF) and Hard Failure (HF) metrics that cover a broad spectrum of potential anomalies. SF refers to subtle deviations from normal operation, while HF denotes more severe disruptions. By incorporating both SF and HF indicators, the PM-C-LSTM model gains improved capability to identify and forecast failures across multiple degradation pathways. This results in greater robustness and accuracy, enabling the detection of early, nuanced failure signals within WE systems. Deployment of this approach leads to reduced downtime, prolonged WT service life, and more efficient cost management, thereby promoting sustained operational efficiency and economic sustainability.

Literature Review

In recent years, scholarly interest in PM for WT systems has grown substantially, with the primary goals of enhancing system dependability and addressing the intricate maintenance demands of these technologies.

Durlik et al. [8] present a comprehensive methodology for fault diagnosis and prognosis to estimate the remaining useful life of WT, particularly in scenarios with limited degradation data. Their framework integrates domain-specific physical knowledge with Bayesian Framework (BF) statistical modeling, employing wavelet transform techniques for fault detection and a Health Index Algorithm (HIA) for multi-feature integration. Real-world Wind Turbine (WT) data were used to validate the approach, demonstrating superior performance compared to standard data-driven methods. Meanwhile, Wang et al. [9] underscore the value of PM for condition monitoring and anomaly identification in WT, based on an analysis of historical SCADA data. They introduce a hybrid model that combines extreme gradient boosting (XGBoost) with LSTM, augmented by Statistical Process Control techniques.

Arumugham et al. [10] introduce a cloud-based digital twin framework supported by 5G-NG-RAN to forecast Wind Speed (WS) and WP outputs using Deep Learning (DL) methods, including Time Convolution Networks (TCN) and k-nearest Neighbors (kNN) regression. This approach surpasses traditional forecasting techniques and offers valuable guidance for WF operations. Additionally, they present a comprehensive framework for anticipating WT failures and evaluating health conditions, incorporating an adaptive maximum mean discrepancy algorithm alongside a CNN to boost prediction accuracy and minimize false positives [11].

Fahim et al. [12], Abdulkader et al. [13], Karn et al. [14], and Singh et al. [15] advocate an Incremental Learning Approach combined with a Dynamic Down-Sampling Method to enhance the sustained effectiveness of WT monitoring systems based on SCADA data. They stress the critical role of continuously updating condition monitoring models in real time and highlight opportunities for early identification of gearbox faults [15-18]. Furthermore, they investigate the application of Deep Reinforcement Learning to optimize O&M processes in RES systems and show that their WF-O&M optimization strategy outperforms alternative DRL methods [19, 20].

This research emphasizes the integration of failure detection and PM to estimate the remaining useful life of WT, particularly under conditions of limited degradation data [21-25]. It validates the performance of a hybrid methodology that incorporates a BF, wavelet transform, and HIA [26]. Nevertheless, existing literature reveals a notable research gap concerning the scalability and computational efficiency of this approach [27]. This limitation restricts a full appreciation of its feasibility in larger, more intricate WT deployments. The identified gap underscores the need for deeper exploration of the model's adaptability across diverse operating environments and its capacity to efficiently process extensive datasets—factors essential for widespread adoption in varied WE scenarios [28].

Taken together, these investigations illustrate the dynamic advancement of PM-WT systems. They point to the growing complexity of maintenance tasks, the pressing need for sophisticated data analytics, and the promising role of ML techniques in improving reliability and operational efficiency in WE production [29]. Building on these prior contributions, the present study develops an IoT + WSN framework integrated with the novel PM-C-LSTM model to push forward progress in PM-WT applications.

Proposed Methods

Wind turbine monitoring and control system

The proposed architecture (Figure 1) begins with a sensor data acquisition module, achieved by deploying sensors directly on the WT. These sensors capture key parameters, including temperature, vibration, and rotational speed.

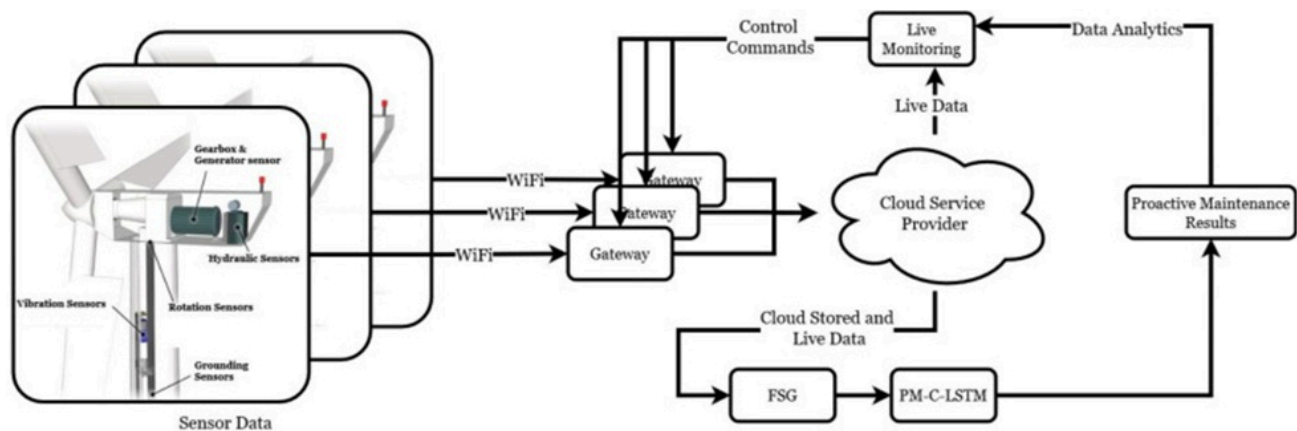


Figure 1. IoT architecture for PM-WT

Temperature and vibration sensors monitor gearbox and generator temperatures and identify structural imbalances. Rotational speed sensors measure the pace of blades and the generator to enhance overall WT system optimization. Central to the data acquisition module is the Micro-Controller Unit (MCU), which consolidates inputs from the various sensors to ensure reliable, precise data transmission for subsequent processing. The architecture next incorporates a gateway component that gathers sensor readings from individual turbines, consolidates them, and performs preliminary processing to prepare the data for transfer [30]. Data transmission to the cloud platform occurs securely and dependably via Wireless Fidelity (Wi-Fi) and Global System for Mobile Communications (GSM) protocols, enabling real-time analytics and storage.

An interoperability protocol efficiently manages and stores incoming data in the cloud, seamlessly linking cloud storage to user interfaces for access. Unlike traditional protocols, this protocol adopts a brokerless design that improves Machine-to-Machine (M2M) interactions and enables multicasting, thereby elevating Quality of Service (QoS) and overall data availability. Moreover, the cloud platform integrates the proposed FSG and PM-C-LSTM models to examine data originating from the target WT. It evaluates a wide array of parameters to identify anomalies and potential faults. Beyond mere historical data review, the model supports real-time oversight, enabling prompt recognition of emerging operational issues. The PM-C-LSTM model effectively identifies potential failures and notifies maintenance personnel during PM efforts. Ultimately, the system helps avoid unplanned outages and maximizes WT operational effectiveness. By incorporating the model's outputs into the broader framework, PM precision is heightened, and maintenance strategies are continually refined.

Failure sample generator for predictive maintenance

The FSG serves as a dedicated PM instrument designed to anticipate failures in WT by simulating operational stresses and possible failure pathways. WT are subjected to diverse environmental and mechanical stresses, including fluctuations in WS, blade and gearbox loads, temperature variations, and icing conditions.

Failures are categorized as follows:

1. **Hard Failures:** Sudden failures triggered by severe stress from extreme wind conditions or mechanical obstructions, requiring urgent repairs and capable of halting operations immediately.
2. **Soft Failures:** Progressive degradation resulting from ongoing operation, gradually impairing performance until eventual breakdown occurs.

The FSG leverages sensor readings from WT components to forecast failures, processing vibration, temperature, and load information to replicate WT states and anticipate issues. This supports proactive maintenance planning by classifying

failure types in advance. Key advantages of this approach include (i) minimizing unanticipated downtime while prolonging turbine lifespan, (ii) averting expensive repairs and yielding long-term cost savings, and (iii) maintaining peak WT efficiency to achieve maximum energy production.

Hard failure (HF)

In WT systems, accurately forecasting HF relies heavily on comprehending multiple stress factors. A compound Poisson process is employed to model these stresses, accounting for their random occurrence times and varying magnitudes; it utilizes a homogeneous Poisson process for the temporal arrival of stresses—such as abrupt wind gusts or sudden load shifts—and applies a normal distribution to capture spatial variations in stress intensity across different WT components. The model incorporates two critical thresholds: Θ and δ . Θ serves to detect overload incidents that exceed the WT's endurance limit, resulting in instantaneous failures. At the same time, δ defines the time interval between consecutive stress events, which may accumulate to trigger HF. Embedding this model within maintenance strategies enables more accurate forecasting and timely corrective actions, thereby enhancing the overall reliability and performance of WT.

- **Wind Turbine Extreme Load Model:** Within WT, extreme loads are stress events that exceed a specified threshold and play a pivotal role in HF prediction. These loads in the original model correspond to operational stresses arising from winds, turbulence, and electromechanical forces—analogueous to extreme impacts yet specific to WT functioning. Θ represents the threshold for extreme loads in WT. Loads exceeding Θ can cause immediate damage to components such as blades or gearboxes. For each extreme load event, Ω_i (the i th event) adheres to a normal distribution characterized by mean μ_Ω (the average intensity of extreme loads) and standard deviation σ (the variability among these loads). The time to failure under cumulative extreme loading is denoted τ_Θ and marks the point at which failure becomes probable under extreme loading. The survival function quantifies the likelihood that the WT endures extreme loads without failing up to time t , adjusted by the threshold Θ (Eq. 1).

$$S_{\tau_\Theta}(t) = P(\tau_\Theta > t) = P\left(\bigcap_i \Omega_i < \Theta\right) = \left(\int_{-\infty}^{\Theta} \frac{1}{\sigma_\Omega \sqrt{2\pi}} e^{-\frac{(t-\mu_\Omega)^2}{2\sigma_\Omega^2}} dt\right)^i \quad (1)$$

In the model, Θ is a critical parameter determined by the WT's structural design constraints and real-world operational records. The mean μ_Ω and standard deviation σ_Ω of the extreme load distribution are derived from historical stress measurements collected from WT. This methodology enables computation of the likelihood that a WT will endure extreme conditions up to a given point in time, thereby informing maintenance decisions and operational strategies to mitigate the risks of intricate failures triggered by severe loads.

- **Rapid Stress Accumulation Model in Wind Turbines:** Within WT maintenance, accurately modeling rapid stress accumulation is essential for forecasting failures induced by closely spaced, high-intensity stresses. A rapid stress event is defined as occurring when consecutive stress episodes (e.g., strong winds or abrupt load variations) happen at intervals shorter than the threshold δ . The occurrence rate of such rapid stress events in WT is modeled as a Poisson process, with the rate parameter equal to the average frequency of these events. The survival probability of the turbine—i.e., the chance of continued operation without failure up to time t —is evaluated using the survival function, which depends on the frequency and temporal spacing of the stress events (Eq. 2).

$$S_{\tau\delta}(t) = P(\tau\delta > t) = P\left(\bigcap_i \text{The interval between events} > \delta\right) = e^{-i\lambda\delta} \quad (2)$$

When combined with the extreme load model, this framework provides a comprehensive assessment of the turbine's vulnerability to both isolated extreme events and rapid-stress sequences. The aggregate survival probability is obtained by multiplying the individual survival probabilities from each respective model.

Soft failure (SF) process

In the PM framework for WT, SF arises from progressive degradation driven by the interplay of steady aging and accelerated wear. One stochastic process simulates the natural aging of blades and gears, while another models environmental and operational irregularities that hasten deterioration. By estimating component lifespans and fine-tuning maintenance interventions to counteract wear and stress-related degradation, this approach maximizes service longevity.

Gradual Aging: Long-term degradation of WT components—such as blade erosion, material fatigue, and insulation breakdown—is characterized as gradual aging. This process is represented using a gamma distribution to model cumulative wear over time. The degradation level $D(t)$ is a time-varying quantity, governed by a gamma distribution with shape parameter α and scale parameter β , which together describe the aging behavior of WT parts. The corresponding probability density function of $D(t)$ is given in [Eq. 3](#),

$$\psi(D(t)) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t}, \quad (3)$$

Where $\Gamma(\alpha)$ denotes the gamma function.

- **Accelerated Damage:** In WT, accelerated damage results from intense stresses—such as overpowering winds or mechanical overloads—that speed up the underlying degradation process. This phenomenon is captured through a compound Poisson process that accounts for the intermittent arrival and magnitude of stress events affecting turbine components. The arrival of operational stress events is modeled as a homogeneous Poisson process with intensity λ , consistent with the hard-failure mechanism. Each stress event A_i (for the i th event) contributes to accelerated degradation and is distributed normally with mean μ_A and standard deviation σ_A . The cumulative accelerated damage up to time t , denoted $\Sigma_A(t)$, is the sum of these individual contributions, and its characteristic function describes the overall impact on turbine degradation ([Eq. 4](#)).

$$\psi_{\Sigma_A(t)}(u) = \exp \left(\lambda t \left[\exp \left(j\mu_A u - \frac{\sigma_A^2 u^2}{2} \right) - 1 \right] \right) \quad (4)$$

SF occurs once the combined degradation—resulting from gradual aging $D(t)$ plus accelerated damage $\Sigma_A(t)$ —surpasses a predefined threshold M . The survival function quantifies the probability that the turbine remains below this critical degradation level up to time t , reflecting the joint influence of natural aging and stress-induced acceleration. This modeling approach estimates the likelihood of SF due to both gradual and accelerated mechanisms, thereby supporting precise maintenance scheduling that accounts for routine wear and stress-driven deterioration.

Proposed machine learning model

The proposed ML model is based on an LSTM architecture, a specialized variant of the Recurrent Neural Network (RNN). The core model employs a standard LSTM architecture to effectively capture temporal sequences and dependencies in WT data, enabling it to recognize emerging patterns indicative of potential failures. This foundational LSTM design serves as the basis for the PM-C-LSTM model, which restructures and enhances the processing and analysis of spatial-temporal features essential to reliable WT operation and maintenance.

Traditional LSTM in predictive maintenance framework

LSTM represents a specialized variant of RNN that incorporates a memory cell along with four essential gates ([Figure 2](#)): the Forget Gate (FG) (f), Input Gate (IG) (i), Update Gate (UG) (c), and Output Gate (OG) (o). This architecture excels at capturing and preserving complex patterns within data, successfully overcoming the vanishing gradient problem and long-term dependency issues commonly encountered in standard RNN models. Variations of LSTM include the conventional

LSTM, Peephole LSTM (P-LSTM), and Convolutional LSTM (C-LSTM), each offering tailored configurations suited to diverse application contexts. Within the PM framework, the traditional LSTM takes as inputs the previous hidden state h_{t-1} , the current input vector x_t , and a bias term b . Through its internal computations, it produces an updated memory cell content c_t and a new hidden state h_t , thereby efficiently regulating the retention and flow of information from the prior cell state c_{t-1} . The fundamental equations governing the traditional LSTM are outlined below:

- Forget Gate (f): Controls the proportion of prior memory information to retain, outputting values between 0 (complete discard) and 1 (full preservation), [Eq. 5](#).

$$f_t = \sigma(w_f \cdot x_t + u_f \cdot h_{t-1} + b_f) \quad (5)$$

- Input Gate (i): Determines the extent to which new information from the current input is incorporated into the memory cell, [Eq. 6](#).

$$i_t = \sigma(w_i \cdot x_t + u_i \cdot h_{t-1} + b_i) \quad (6)$$

- Update Gate (c): Computes the candidate values to potentially add to the cell state, taking into account the outputs of both the Forget Gate and Input Gate, [Eq. 7](#).

$$c_t = f_t \times c_{t-1} + i_t \times \tanh(w_c \cdot x_t + u_c \cdot h_{t-1} + b_c) \quad (7)$$

- Output Gate (o): Governs the information passed to the current output and contributes to updating the hidden state, [Eq. 8](#) and [Eq. 9](#).

$$o_t = \sigma(w_o \cdot x_t + u_o \cdot h_{t-1} + b_o) \quad (8)$$

$$h_t = o_t \times \tanh(c_t) \quad (9)$$

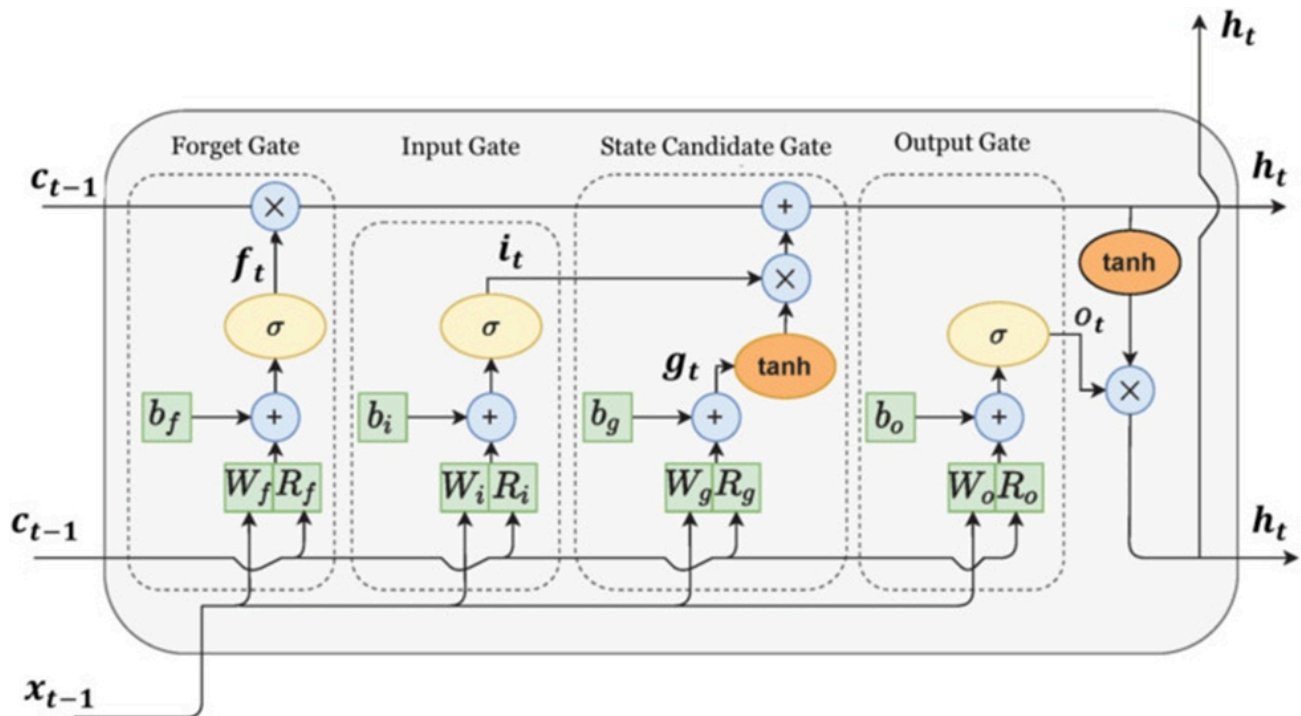


Figure 2. Traditional LSTM

The LSTM network processes FSG data using an input layer, two hidden layers, and a dense output layer. Each of these layers consists of 50 LSTM units, which play a crucial role in detecting and interpreting complex patterns within the data. An attention mechanism is integrated to evaluate the relative importance of individual inputs across layers, thereby assisting in assessing maintenance needs for various WT components. A notable drawback of traditional LSTM is the potential interference from prior memory content in the gating mechanism, which may compromise classification performance and reduce PM accuracy. To mitigate this issue, the present study proposes adopting Convolutional LSTM (C-LSTM) for PM applications. C-LSTM aims to overcome the limitations of both traditional LSTM and standard CNN-LSTMs by providing a more effective and resilient approach to analyzing the intricate, multifaceted data patterns characteristic of WT operations.

CNN in wind turbine predictive maintenance

A CNN is a Deep Learning (DL) algorithm that operates in a manner akin to a conventional Neural Network (NN), with individual neurons receiving targeted inputs. These neurons are adjusted during training through learnable weights and biases that facilitate operations such as the dot product. Originally developed and widely applied in image classification tasks, CNNs have since proven highly effective across a broad range of domains. In the domain of PM for WT, the CNN architecture is trained and evaluated by passing inputs through a sequence of specialized layers: convolutional layers with filters, a pooling layer, Fully Connected (FC) layers, and, ultimately, a SoftMax function for generating final predictions. The key stages of this process are outlined below:

- **Convolution Layer:** This layer extracts meaningful patterns by learning distinctive features from input data, such as WT sensor readings. It performs a dot product between the input and filter weights to produce a feature map, to which a bias term is added to help alleviate the vanishing gradient problem. The layer applies the Rectified Linear Unit (ReLU) activation function to introduce nonlinearity into the network.
- **Pooling Layer:** The rectified feature maps generated by the convolution layer are downsampled within the pooling layer. Through max pooling, the most prominent features are retained while reducing the spatial dimensions of the data, thereby preserving essential information with lower computational complexity.

- **Additional Convolution and Pooling Layers:** Further refinement of feature extraction is achieved by repeating the convolution and pooling operations in subsequent layers.
- **FC:** Following the pooling stages, the resulting feature map is flattened into a one-dimensional vector and passed to the Fully Connected (FC) layer. This layer combines the extracted features to construct a comprehensive representation of the diverse operational characteristics observed in WT data.
- **SoftMax Function:** The final classification stage employs the SoftMax function to produce probability distributions over possible classes. In a PM context, this enables accurate labeling of WT components' operational conditions, such as forecasting impending failures and determining specific maintenance requirements.

By incorporating CNNs in this structured manner, the model can efficiently process and interpret the complex data patterns generated by WT systems, thereby providing actionable insights for PM strategies. This approach enhances the precision and dependability of maintenance planning, ultimately improving WT performance, reliability, and overall service life.

PM-C-LSTM

The proposed PM-C-LSTM model (Figure 3) integrates the complementary strengths of CNN and LSTM. This hybrid design leverages the CNN component's proficiency in extracting concealed spatial features from the input dataset, while harnessing the LSTM component's exceptional ability to model and process sequential and temporal dependencies.

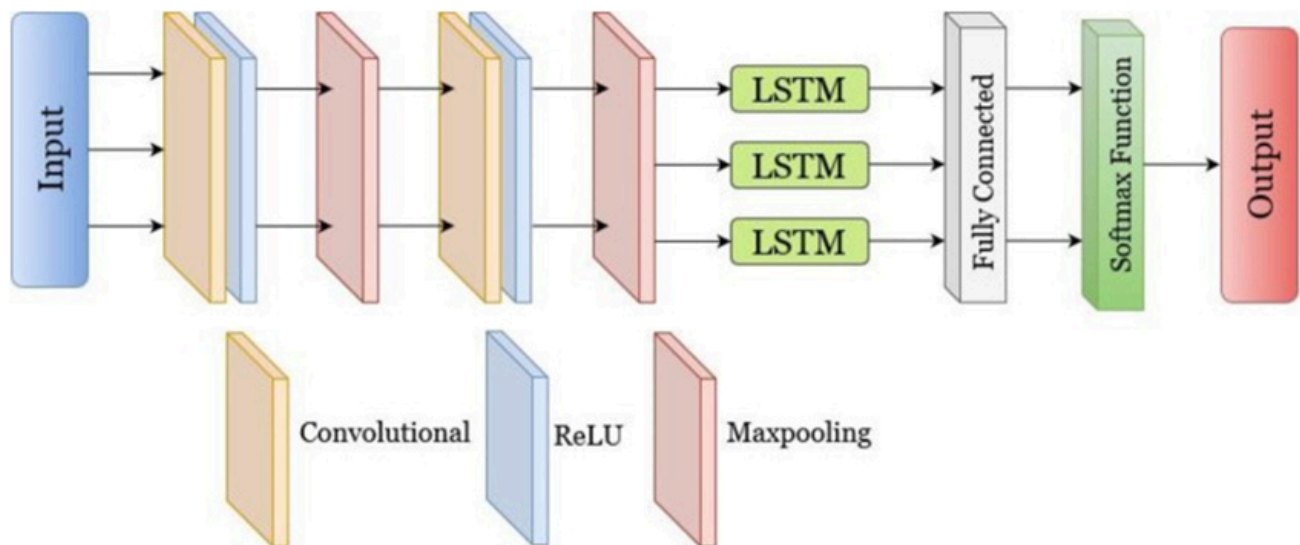


Figure 3. PM-C-LSTM architecture.

This study elaborates on the rationale behind integrating CNN and LSTM within the PM-C-LSTM model. The combination capitalizes on the complementary strengths of CNNs for detecting spatial patterns and LSTM for modeling sequential dependencies. As illustrated in Figure 3, the architecture reveals how the Convolutional Neural Network (CNN) component captures spatial information and identifies intricate patterns, while the LSTM component simultaneously manages temporal relationships across time steps. By leveraging CNN's capacity for automatic hierarchical feature learning and LSTM's proficiency in capturing long-range temporal context, the model effectively evaluates both spatial and time-dependent characteristics critical for PM in WT systems.

As depicted in Figure 3, the PM-C-LSTM architecture comprises three primary modules: an input layer, a series of convolutional layers, and subsequent LSTM layers. The input layer accepts raw data and applies a sliding window technique to prepare sequential inputs for the network. Following this, the convolutional layers extract spatial features

from the input. Each convolutional layer applies a set of convolution kernels to identify relevant spatial patterns. The mathematical operation performed in these layers is expressed as follows,

$$c_i^{(l,j)} = \text{ReLU} \left(\sum_{k=1}^K Z_k^{(l,j)} \cdot x_{i+k-1}^{(l,j)} + b_j^l \right) \quad (10)$$

In Eq.10 above, l indicates the layer index, Z_k^j represents the k th convolution kernel applied to feature map j , x denotes the input from the preceding layer, and b_j^l is the bias term associated with the j th feature map in layer l . In the proposed PM-C-LSTM network, a normalization layer is inserted immediately after each convolution operation to accelerate convergence and stabilize training. This normalization transforms the input vector $x_i^{(1,j)}$ according to:

$$n_i^{(l,j)} = \frac{c_i^l - m_c}{\sqrt{\epsilon^2 + \epsilon}} \quad (11)$$

Where m_c is the mean and ϵ denotes the standard deviation of the input vector. Following normalization, the data proceeds to a pooling layer. The pooling layer reduces the spatial dimensions of the feature maps while preserving the most salient information. In this architecture, MaxPooling is utilized, defined as:

$$p_i^{(l,j)} = \max_{s \in S} \left(n_{i \times T + s}^{(l,j)} \right) \quad (12)$$

Where S is the size of the pooling window, and T is the stride (step size) of the sliding window over the input. This operation effectively compresses the feature map dimensions while retaining key characteristics. The Deep Neural Network (DNN) backbone—comprising convolutional, normalization, and pooling layers—progressively increases in complexity, yielding higher accuracy in analyzing WT sensor data and recognizing PM-relevant patterns.

Once the multi-layer convolutional processing is complete, the extracted features are passed to the LSTM units. These units are tasked with updating and maintaining temporal dependencies within the feature sequences. In the final stage of the PM-C-LSTM model for PM-WT applications, feature sequences from different sensors are fused via the FSG. Unlike the kernel-based feature aggregation in CNN, LSTM relies on memory cells to capture and propagate long-term temporal relationships across the entire sequence. The LSTM incorporates a gating mechanism that selectively regulates the flow of information, controlling the cell state through operations such as writing, reading, and forgetting. These memory cells operate sequentially across the full data sequence and primarily support linear transformations. The input processing within the LSTM layer is formally described as follows.

$$h_t^{(l,j)} = \text{ReLU} \left(\omega_{t,h}^{(l,j)} \cdot x_t^{(l,j)} + h_{t-1}^{(l,j)} \cdot \omega_{t-1,h}^{(l,j)} + b_t^l \right) \quad (13)$$

In Eq.13, b_t^l represents the bias for each unit at time t , x_t^l is the input vector fed to the LSTM unit at time t , and ω denotes the corresponding weight coefficients. This mechanism is essential for the LSTM layer, as it enables the updating of temporal features based on information synthesized from prior layers, thereby improving the model's ability to forecast precise maintenance requirements. A network that processes a single data sequence may struggle to accommodate diverse sensor types and their unique characteristics. To overcome this, the proposed architecture avoids weight sharing across different sensor streams, allowing more effective differentiation of sensor-specific patterns. The PM-C-LSTM module is deliberately configured to handle individual sensor data streams independently; this modular design facilitates

the incorporation of new sensor types, thereby enhancing the model's scalability and adaptability. After processing through the PM-C-LSTM module, the resulting features are fed in parallel into a Fully Connected (FC) layer. The FC layer consolidates the outputs generated from multiple sensor-specific networks into a unified representation.

$$h_t^l = \text{ReLU} \left(h_{t1}^{(l,j)} : h_{t2}^{(l,j)} : \dots : h_{tn}^{(l,j)} \right) \quad (14)$$

In this Eq.14, h_t^l represents the integrated output at time t for layer l , synthesized by aggregating outputs from individual sensors $h_{t1}^{(l,j)}, h_{t2}^{(l,j)}, \dots, h_{tn}^{(l,j)}$. The Fully Connected (FC) layer merges these heterogeneous sensor-derived features into a unified representation suitable for precise predictions. In the concluding phase of this layer configuration, the SoftMax classifier processes and interprets the refined data, assigning it to distinct classes corresponding to the WT's various operational states. The classification is performed according to the following expression:

$$F(\vartheta | h) = \text{argmax} \left(\frac{\exp(h^{(L-1)} \cdot w^L + b^L)}{\sum_{n=1}^N \exp(h^{(L-1)} \cdot w^L)} \right) \quad (15)$$

Where ϑ denotes the predicted output class, N is the total number of possible classes, and L indicates the number of layers in the neural network. This classification mechanism enables accurate determination of the WT's current operational condition and corresponding maintenance needs. Once the data has passed through the successive layers and reached the SoftMax classifier, the model evaluates its performance by calculating the loss function. The loss quantifies the discrepancy between the predicted outputs and the true operational states or conditions. Subsequently, based on this computed loss, the model executes backpropagation to adjust the network weights. This iterative cycle of forward propagation, loss computation, and backpropagation continues until the model converges on an optimal configuration for classifying states, thereby enhancing its accuracy and reliability in fulfilling PM requirements for WT. Network optimization is achieved through the Adam optimizer for weight updates, as described in Eq.16.

$$k_i = \beta_1 \cdot k_{i-1} + (1 - \beta_1) \cdot dx \quad (16)$$

$$v_i = \beta_2 \cdot v_{i-1} + (1 - \beta_2) \cdot dx^2 \quad (17)$$

$$W_i = W_{i-1} - \frac{\alpha \cdot k_i}{\sqrt{v_i}} \quad (18)$$

In Eqs.17 and 18, β_1 and β_2 represent the exponential decay rates for the first- and second-moment estimates, α is the learning rate, and W denotes the weight matrix.

Experiment analysis

The PM-C-LSTM model was evaluated using a real-world dataset collected from a Wind Farm (WF) located on the Qinghai-Tibet Plateau, comprising 124 turbines managed by Mother Earth Energy. The analysis focused on three specific turbines labeled T-001, T-017, and T-033. The dataset covers a wide range of sensor measurements—including gearbox, generator, and transformer conditions—spanning 16 months from June 2021 to September 2022. Measurements were recorded at 15-minute intervals and included generator bearing temperatures, oil temperature, pitch angle, generator rpm, and environmental variables such as WS, wind direction, ambient temperature, atmospheric pressure, humidity, and precipitation. The dataset also includes documented failure events for the gearbox, transformer, and generator

assemblies, along with statistical summaries (minimum, maximum, average, and variance) for key variables. These elements are essential for assessing the PM-C-LSTM model's effectiveness in PM tasks.

Figure 4 illustrates the observed WE output in MWh throughout the experimental period. The total number of detected failures during this timeframe is summarized in Tables 1 and 2, and Figure 5.

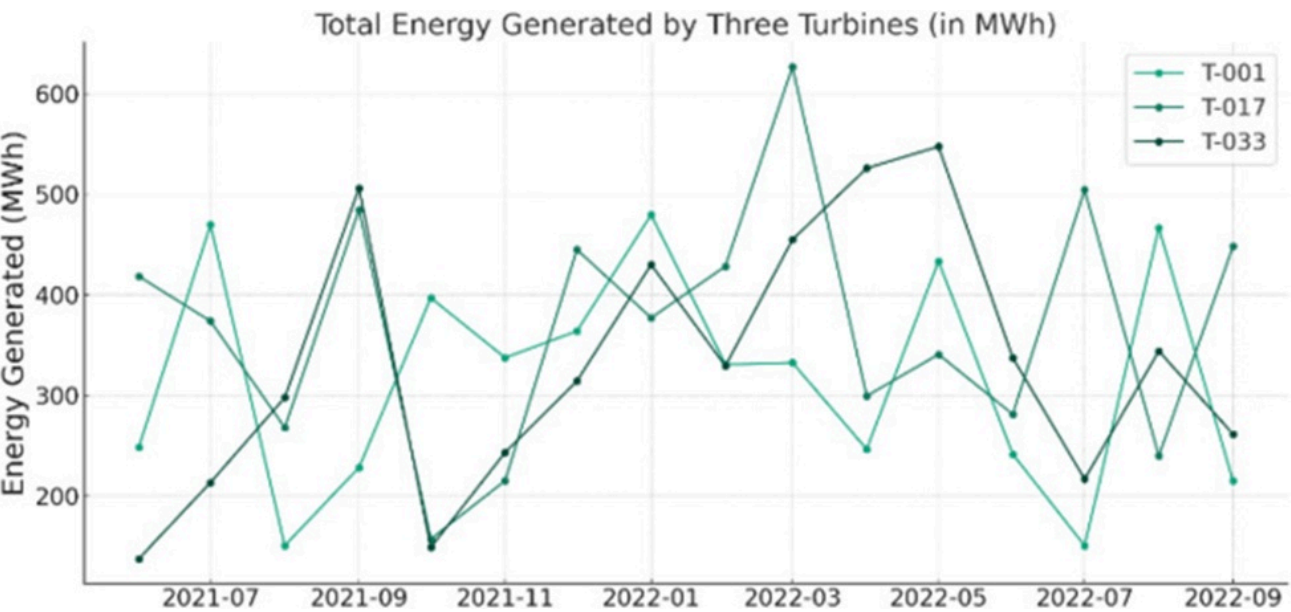


Figure 4. Energy generated during the experiment period.

Table 1. Comprehensive analysis of literature.

Reference	Methodology	Inference	Challenge
Rajagopalan et al. [21]	BF with wavelet transform and HIA	Practical failure detection and prognosis, validated with real-world WT data	Insufficient discussion on scalability and computational efficiency
Neffati et al. [22]	XGBoost, LSTM, and Statistical Process Control (SPC) with historical SCADA data	Effective PM and anomaly detection in WT	Insufficient exploration of model adaptability under varying WT conditions
Zhao et al. [23]	5G-NG-RAN-assisted cloud-based digital twins with DL models (TCN, kNN regression)	Outperformed traditional prediction models for WS and WP generation	Potential challenges in deploying 5G infrastructure across all WF locations
Qin et al. [24]	Adaptive maximum mean discrepancy algorithm and CNN for WT failure prediction	Enhanced failure prediction rates, thorough understanding of degradation trends	Limited discussion on practical implementation challenges
Wu et al. [25]	Incremental Learning Approach (ILA) and Dynamic Down-Sampling Method (DDSM) with SCADA data	Improved long-term performance for gearbox fault detection	Lack of clarity regarding the scalability of the proposed methodology

Xiong et al. [26]	Deep Reinforcement Learning (DRL) with proximal policy optimization, imitation learning, and environmental modeling	Superior WF-O&M optimization performance	Insufficient discussion on potential ethical concerns and real-world applicability
-------------------	---	--	--

Table 2. Details of the failure of a component in each WT.

ID	Component	Timestamp	Observations
T-001	Gearbox	2021-06-15 10:30	Misalignment of the bearing
T-001	Generator	2021-08-21 11:00	Detected overheating
T-001	Transformer	2022-01-30 09:15	Insulation failure
T-001	Gearbox	2022-04-12 14:45	Leakage of oil
T-017	Transformer	2022-05-19 13:30	Imbalance in voltage
T-033	Gearbox	2021-07-22 16:00	Wear in gear teeth
T-033	Generator	2022-03-10 10:15	Imbalance in the rotor
T-033	Transformer	2022-09-05 08:50	Malfunction in the cooling system

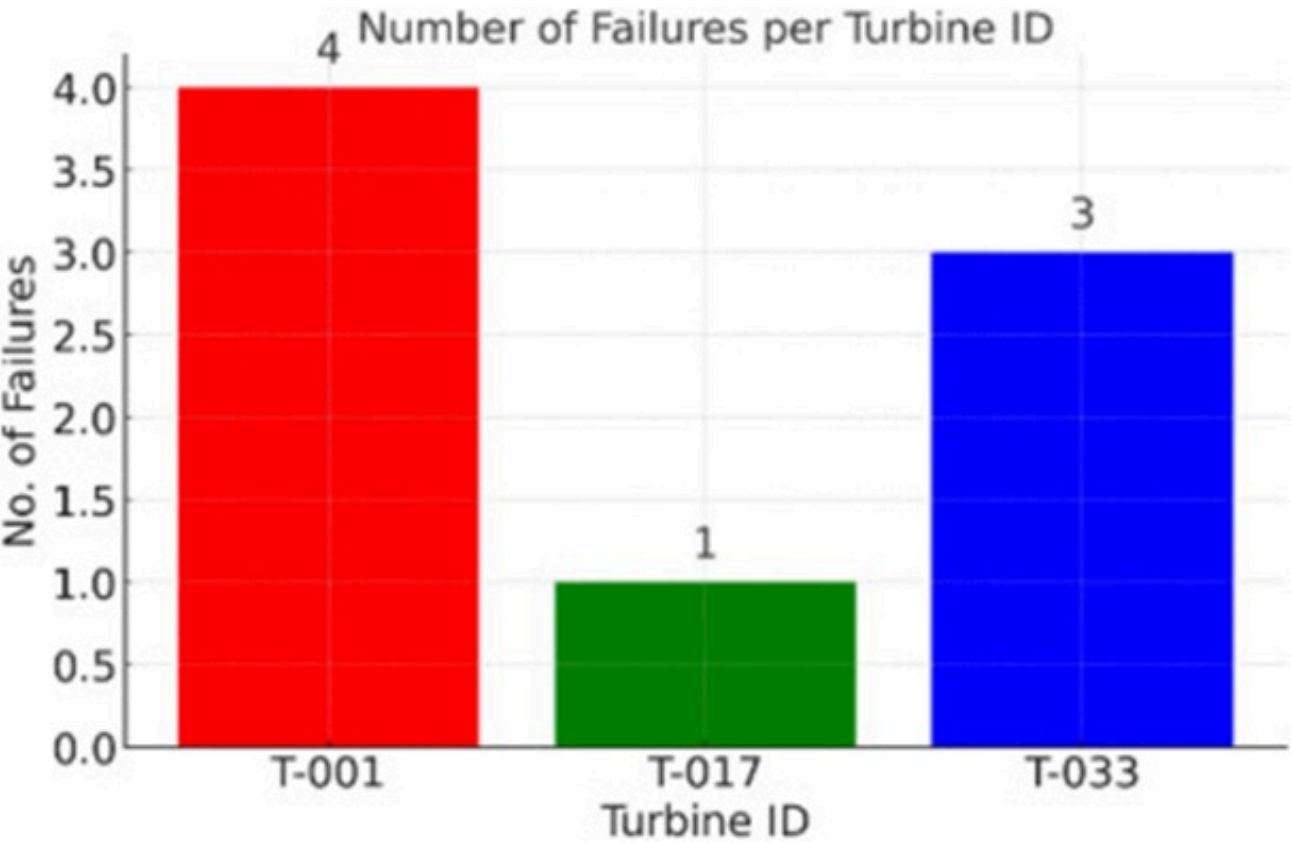


Figure 5. Failure count on each WT.

Data preparation

Data preparation consisted of merging meteorological measurements with turbine sensor readings into a unified dataset. Both data streams were originally collected at 15-minute intervals. Sensor data exhibited missing values in 0.5% of records, while weather-related data was unavailable for 23.6% of the timestamps. To enable integration, missing values were imputed using placeholder techniques. Following consolidation, the resulting dataset contained 86 attributes (Table 3). The FSG was then applied to this combined dataset to generate the corresponding failure-related parameter values.

Table 3. FSG measured the parameter value.

Parameter	Description	Value
Θ	Limit for extreme loads	8.72
δ	Critical interval for stress accumulation	0.0625
$\mu\Omega$	Average of extreme load distribution	2658.03
$\sigma\Omega$	Standard deviation of extreme load distribution	79.47
$\tau\Theta$	Time to potential failure under extreme load	6.25
λ	Frequency of stress events (Poisson process)	0.1
α	Shape parameter of gradual aging (Gamma distribution)	18.0
β	Scale parameter of gradual aging (Gamma distribution)	2880.0
μA	Average damage from each stress event	28.36
σA	Standard deviation of damage from each stress event	0.2537
M	Threshold for total degradation leading to SF	44.53

After defining the failure parameters, the FSG was executed 125 times, with each execution intentionally limited to exactly 15 failure instances to maintain balanced and consistent data quality. This process yielded a comprehensive dataset comprising 874,326 samples. The full dataset was subsequently divided into 75% for training and 25% for validation and testing of the PM model. Before initiating training, selecting an appropriate loss function for the PM-C-LSTM algorithm is essential, as it directly influences the gradient computations required for effective model optimization.

Loss function

For this investigation, the Huber Loss Function (HLF) was selected due to its suitability for regression tasks involving datasets prone to outliers, as defined in Eq.19:

$$L_{\delta}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{If } |y - \hat{y}| \leq \delta \\ \delta(|y - \hat{y}| - \frac{1}{2}\delta) & \text{Otherwise.} \end{cases} \quad (19)$$

Where y denotes the true value and $\hat{y}$ represents the predicted value, while δ is a tunable threshold that determines the transition point between quadratic and linear loss behavior. For deviations smaller than δ , the loss is quadratic, ensuring

sensitivity to minor errors; for larger deviations, it switches to a linear form, thereby reducing the disproportionate influence of outliers.

The proposed PM-C-LSTM model was trained on the dataset using the hyperparameters listed in **Table 4**:

Table 4. Proposed PM-C-LSTM hyperparameter.

Hyperparameter	Value/Range
Number of Convolution Layers	3
Convolution Kernel Size	3 × 3
Number of LSTM Layers	2
LSTM Units	50
Pooling Type	Max pooling
Pooling Window Size	2 × 2
Pooling Step Size	2
Standardization Method	Mean and Standard Deviation
Activation Function	ReLU
Output Activation Function	SoftMax
Loss Function	HLF
Threshold for HFL (δ)	1.0
Optimizer	Adam
Learning Rate (α)	0.001
Beta_1 for Adam Optimizer	0.9
Beta_2 for Adam Optimizer	0.999
Number of Epochs	100

Baselines and metrics

To assess the performance of the PM-C-LSTM model, it was compared against several established baselines: Decision Trees (DT), Random Forest (RF), CNN, XGBoost, LSTM, and CNNLSTM. Evaluation was conducted using the following standard classification metrics: Accuracy, Precision, Recall, and F1-score.

i) Accuracy (Acc): Represents the ratio of correctly classified instances (both true positives and true negatives) to the total number of instances. It provides an overall measure of model effectiveness, calculated as **Eq. 20**:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}, \quad (20)$$

Where TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives, respectively.

ii) Precision: Quantifies the proportion of positive predictions that were actually correct. It reflects the reliability of positive classifications, given by Eq. 21.

$$P = \frac{TP}{TP+FP}, \quad (21)$$

iii) Recall: Measures the proportion of actual positive instances that were successfully identified. It is especially important in applications where failing to detect a positive case incurs high costs, as expressed by Eq. 22.

$$Recall = \frac{TP}{TP+FN}. \quad (22)$$

iv) F1-score: The harmonic mean of Precision and Recall, offering a balanced single metric that accounts for both aspects simultaneously, computed via Eq. 23.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (23)$$

Post-processing

During post-processing, model performance deviations (Δ) are quantified as the difference between observed (measured) and predicted values. This study applies Statistical Process Control using the Cumulative Sum (CUSUM) control chart to monitor these deviations. The CUSUM method relies on two key parameters: i) the Reference Value (k) and ii) the Control Limits (UCL and LCL), which establish the boundaries for detecting statistically significant deviations, as specified in Eqs. 24-27.

$$UCL = k + Allowed\ Deviation \quad (24)$$

$$LCL = k - Allowed\ Deviation \quad (25)$$

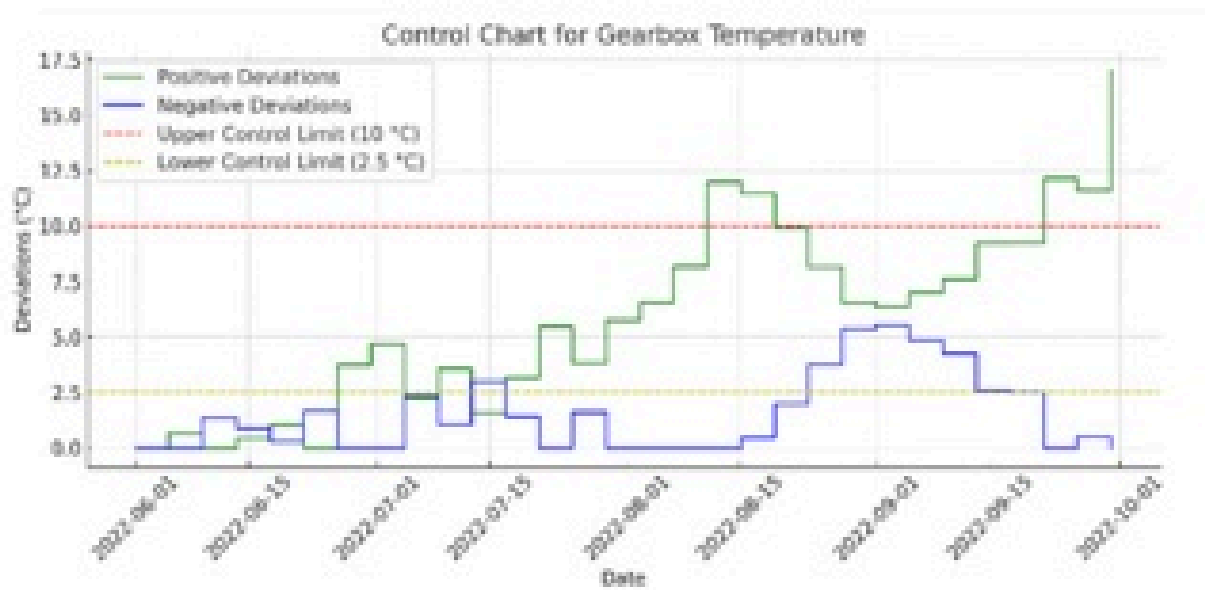
The CUSUM statistic for each observation i is calculated according to Eqs. 26 and 27.

$$S_{high,i} = \text{Max}(0, S_{high,i-1} + (\Delta_i - k)) \quad (26)$$

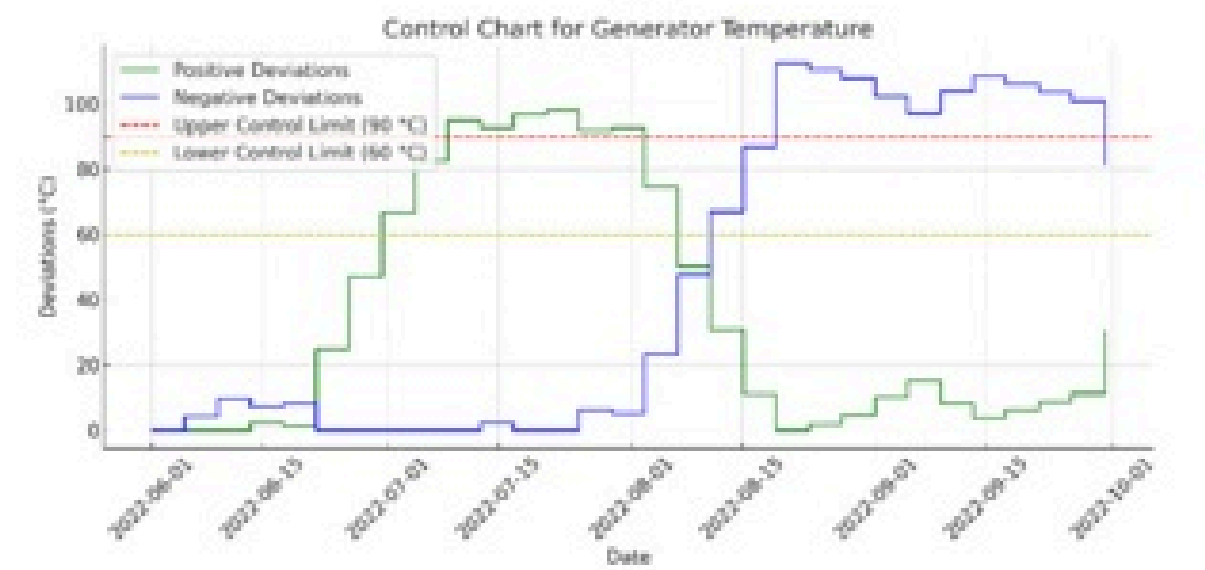
$$S_{low,i} = \text{Max}(0, S_{low,i-1} - (\Delta_i + k)) \quad (27)$$

Data points falling within the control limits are interpreted as consistent with normal operating conditions, indicating close agreement between measured and predicted values. When deviations exceed these limits, they signal anomalous behavior that may warrant further inspection or immediate maintenance intervention. This mechanism supports proactive

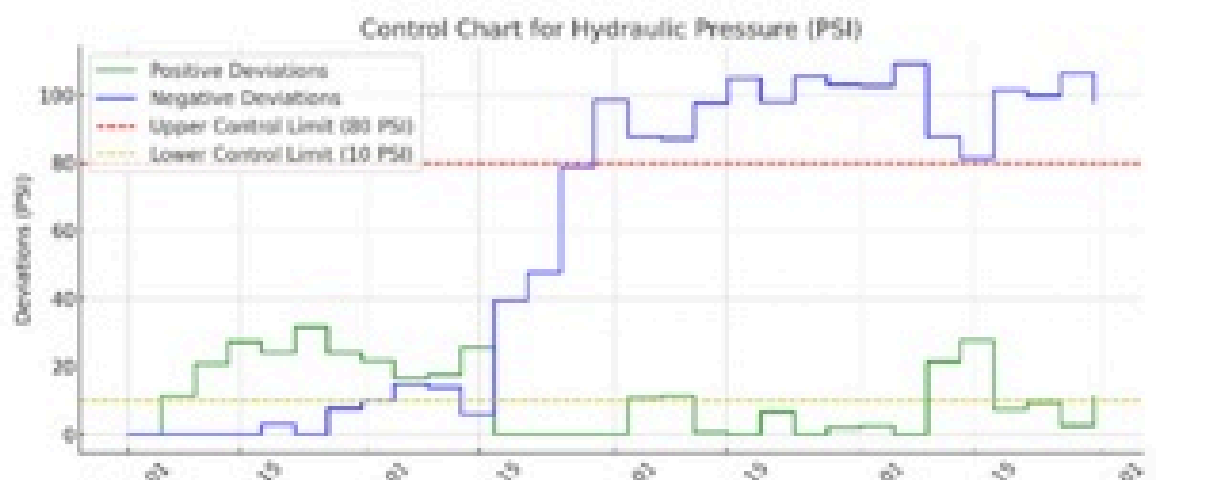
health management of WT components. **Figure 6a** through **Figure 6d** display the CUSUM charts illustrating deviations observed for turbine T-017.



(a)



(b)



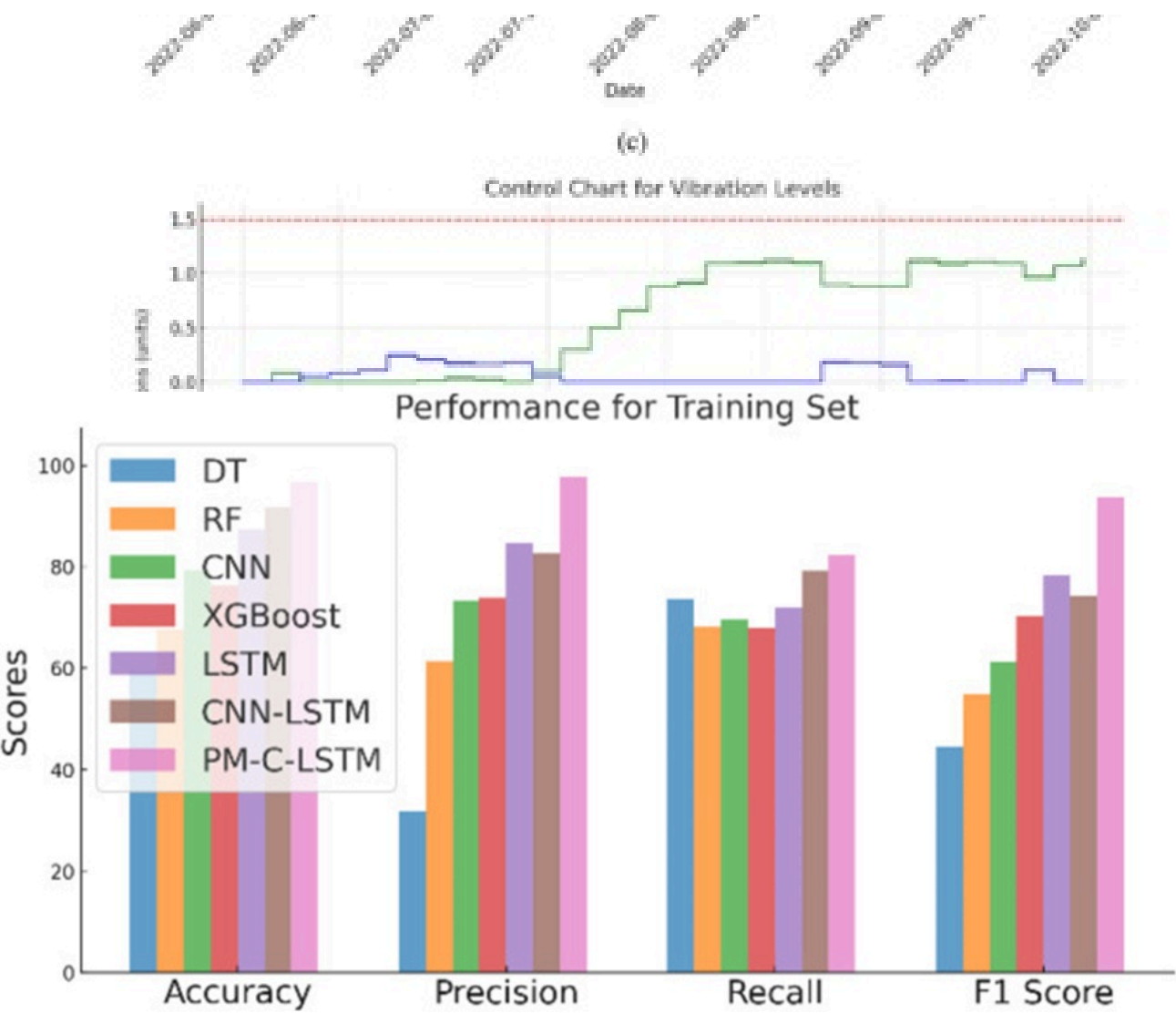


Figure 7. Training set results.

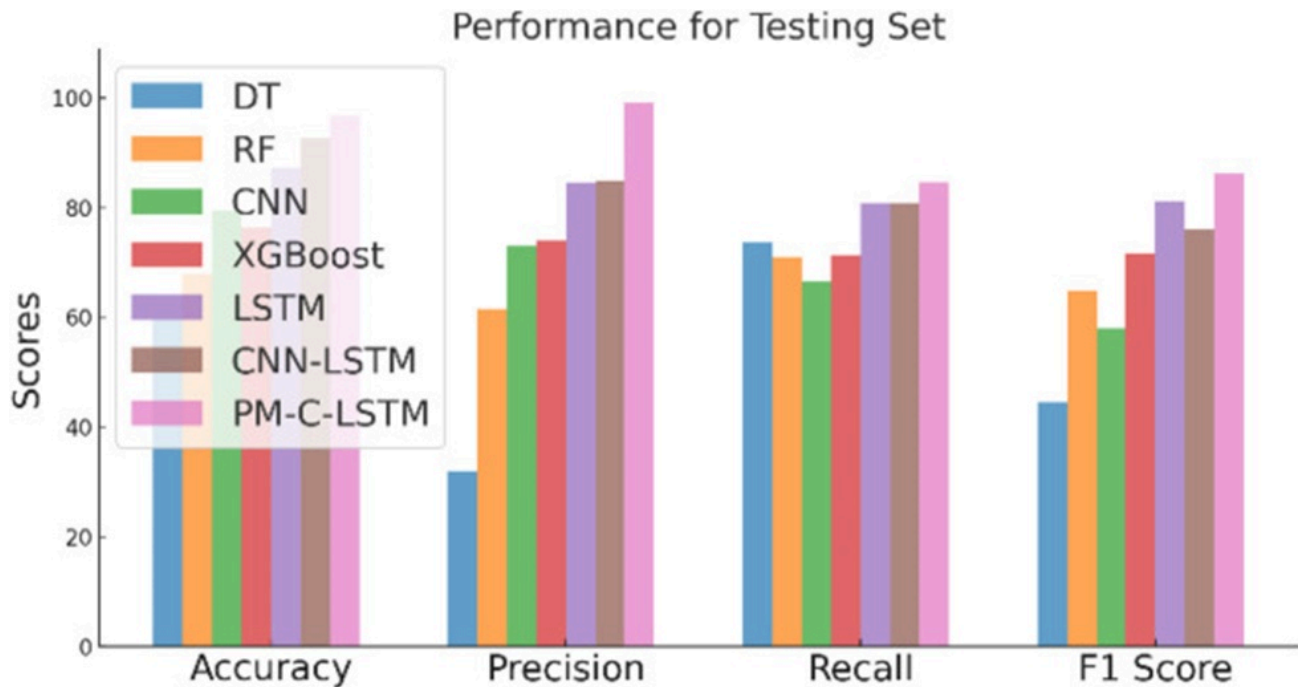


Figure 8. Test set results.

In contrast, conventional models such as DT and RT achieved lower accuracy and F1 scores, whereas RF showed moderate gains over DT. Although CNN outperformed DT and RF, it fell short of the results achieved by ensemble and RNN-based methods. XGBoost demonstrated a strong balance across metrics, and LSTM-based models delivered consistently high scores, especially on the test set. The CNN-LSTM hybrid showed incremental gains, most noticeably in recall and F1-score on the test data.

Analysis of the loss curves presented in [Figure 9](#), which compare HLF across various ML models, further confirms the superiority of PM-C-LSTM. It achieved the lowest loss value, slightly above 6×10^{-3} , following a rapid initial decrease and stabilization after approximately 30 epochs. The CNN and CNN-LSTM models ranked next, with early plateaus and final loss values near 1×10^{-2} , indicating solid learning performance. LSTM performed comparably to the CNN-based approaches. Although RF showed a steady reduction in loss, it ended at a higher value of around 1×10^{-1} . The DT model recorded the highest loss, exceeding 4×10^{-1} , underscoring its limited ability to model complex patterns in the dataset. XGBoost positioned itself between RF and neural network-based models, with a slightly lower loss than RF's.

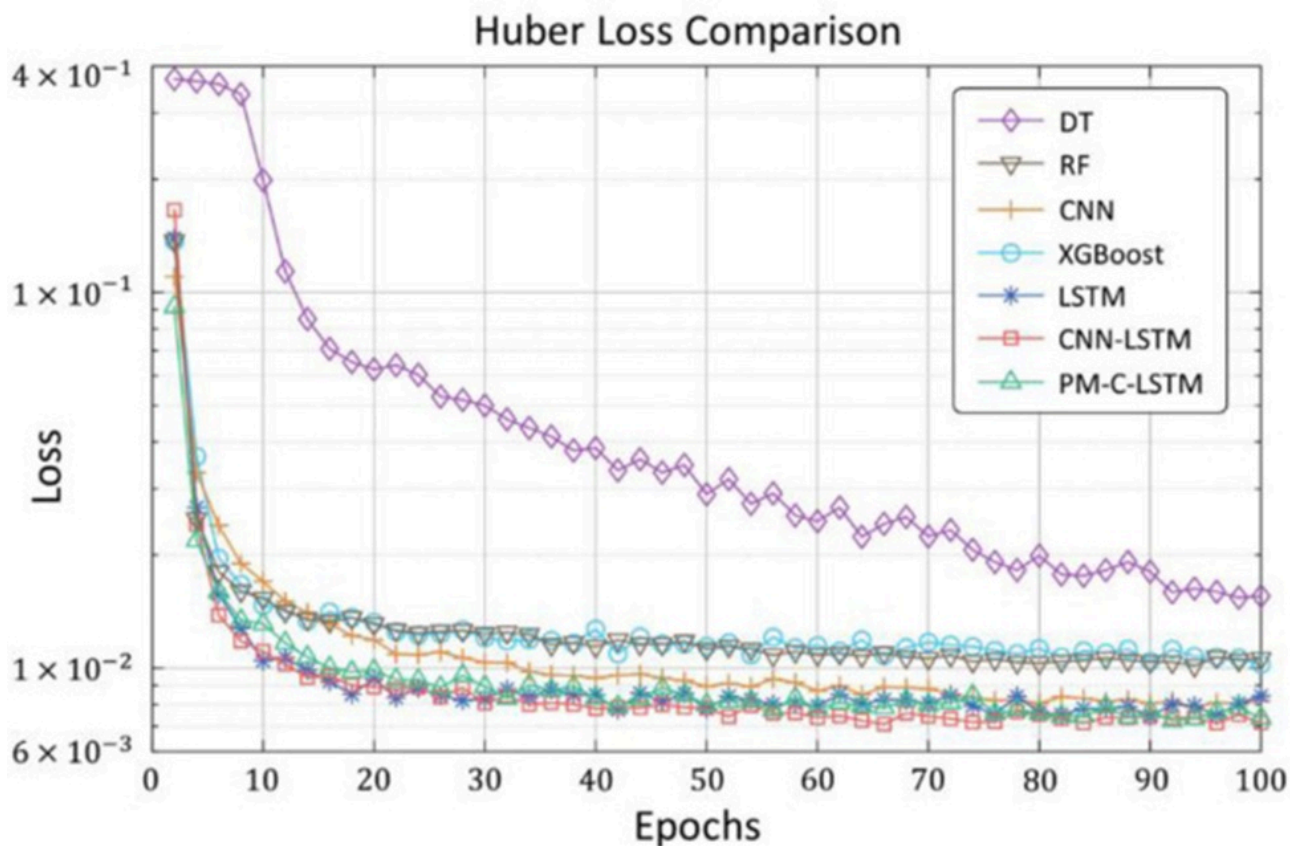


Figure 9. Comparison of prediction losses.

Conclusion

This research effectively demonstrates the value of combining Machine Learning with the Internet of Things and Wireless Sensor Networks to improve Predictive Maintenance for Wind Turbines significantly. The introduced Failure Sample Generator and the Predictive Maintenance Convolutional Long Short-Term Memory (PM-C-LSTM) model together enable more precise forecasting of maintenance requirements for wind turbines. The proposed framework was implemented and tested using data collected over 16 months from three wind turbines at a wind farm located on the Qinghai-Tibet Plateau. Thanks to its synergistic integration of CNN for spatial feature extraction and LSTM for temporal sequence modeling, the recommended model consistently outperformed alternative approaches in accurately anticipating maintenance needs. The synergy of IoT and WSN has transformed real-time data acquisition and continuous monitoring, delivering a thorough understanding of wind turbine behavior. These algorithms successfully detect potential failures within extensive wind farm datasets, paving the way for the adoption of sophisticated ML techniques in the maintenance of wind energy infrastructure.

Future work should explore the proposed model's adaptability across diverse wind farm configurations and assess its robustness under varying environmental conditions. Additional studies are recommended to investigate scalability aspects, compatibility with emerging technologies, and strategies for ongoing refinement in the domain of predictive maintenance for wind turbines.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 01 Apr 2023

Revised: 13 May 2023

Accepted: 22 Jun 2023

Published online: 20 September 2023

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Olabi AG, Obaideen K, Abdelkareem MA, AlMallahi MN, Shehata N, Alami AH, et al. Wind energy contribution to the sustainable development goals: case study on London Array. *Sustainability*. 2023;15(5):4641.
<https://doi.org/10.3390/su15054641>.
- Cheng YL, Lim MH, Hui KH. Impact of Internet of things paradigm towards energy consumption prediction: A systematic literature review. *Sustain Cities Soc*. 2022;78:103624.
- Li S, Patnaik S, Li J. IoT-based technologies for wind energy microgrids management and control. *Electronics*. 2023;12(7):1540.
- Veers P, Bottasso CL, Manuel L, Naughton J, Pao L, Paquette J, et al. Grand challenges in the design, manufacture, and operation of future wind turbine systems. *Wind Energy Sci. Discuss*. 2022;2022:1–102.
- Molęda M, Małysiak-Mrozek B, Ding W, Sunderam V, Mrozek D. From corrective to predictive maintenance – a review of maintenance approaches for the power industry. *Sensors*. 2023;23(13):5970.
<https://doi.org/10.3390/s23135970>.
- Tran MQ, Doan HP, Vu VQ, Vu LT. Machine learning and IoT-based approach for tool condition monitoring: a review and future prospects. *Measurement*. 2023;207:112351.
- Mehbodniya A, Bhatia S, Mashat A, Elangovan M, Sengan S. Proportional fairness based energy efficient routing in wireless sensor network. *Comput Syst Sci Eng*. 2022;41(3):1071–82.
<https://doi.org/10.32604/csse.2022.021529>.
- Durlik I, Miller T, Cembrowska-Lech D, Krzemińska A, Złoczowska E, Nowak A, et al. Navigating the sea of data: a comprehensive review on data analysis in maritime IoT applications. *Appl Sci*. 2023;13(17):9742.

<https://doi.org/10.3390/app13179742>.

Wang J, Liang Y, Zheng Y, Gao RX, Zhang F. An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples. *Renew Energy*. 2020;145:642–50.

Arumugham V, Ghanimi HMA, Pustokhin DA, Pustokhina IV, Ponnamm VS, Alharbi M, et al. An artificial-intelligence-based renewable energy prediction program for demand-side management in smart grids. *Sustainability*. 2023;15(6):5453.
<https://doi.org/10.3390/su15065453>.

Udo W, Muhammad Y. Data-driven predictive maintenance of wind turbine based on SCADA data. *IEEE Access*. 2021;9:162370–88.

Fahim M, Sharma V, Cao TV, Canberk B, Duong TQ. Machine learning-based digital twin for predictive modeling in wind turbines. *IEEE Access*. 2022;10:14184–94.

Abdulkader R, Ghanimi HMA, Dadheech P, Alharbi M, El-Shafai W, Fouda MM, et al. Computing in smart grid with decentralized generation and renewable energy storage system planning. *Energies*. 2023;16(6):2655.

Karn AL, Kondamudi BR, Gupta RK, Pustokhin DA, Pustokhina IV, Alharbi M, et al. An empirical analysis of the effects of energy price shocks for sustainable energy on the macro-economy of South Asian countries. *Energies*. 2023;16(1):363.
<https://doi.org/10.3390/en16010363>.

Singh NP, Suman S, Ramachandran TP, Sharma T, Raja S, Rangasamy R, et al. Investigation on characteristics of Monte Carlo model of single electron transistor using orthodox theory. *Sustain Energy Technol Assess*. 2021;48:101601.
<https://doi.org/10.1016/j.seta.2021.101601>.

Peng J, Kimmig A, Niu Z, Wang J, Liu X, Wang D, et al. Wind turbine failure prediction and health assessment based on adaptive maximum mean discrepancy. *Int J Electr Power Energy Syst*. 2022;134:107391.

Wang Z, Liu C, Yan F. Condition monitoring of wind turbine based on incremental learning and a multivariate state estimation technique. *Renew Energy*. 2022;184:343–60.

Kumar KS, Esther BP, Indragandhi V, Sudhakar S, Ravi L, Subramaniaswamy V. Area based efficient and flexible demand side management to reduce power and energy using evolutionary algorithms. *Malays J Comput Sci*. 2020;33(4):61–77.

Pincirolì L, Baraldi P, Ballabio G, Compare M, Zio E. Optimization of the operation and maintenance of renewable energy systems by deep reinforcement learning. *Renew Energy*. 2022;183:752–63.

Nanthini S, Kalyani SN, Sengan S. Energy efficient clustering protocol to enhance network lifetime in wireless sensor networks. *Comput Mater Continua*. 2021;68(3):3595–614.
<https://doi.org/10.32604/CMC.2021.015038>.

Rajagopalan A, Swaminathan D, Alharbi M, Sengan S, Montoya OD, El-Shafai W, et al. Modernized planning of smart grid based on distributed power generations and energy storage systems using soft computing methods. *Energies*. 2022;15(23):8889.
<https://doi.org/10.3390/en15238889>.

Neffati OS, Sengan S, Thangavelu KD, Kumar SD, Setiawan R, Elangovan M, et al. Migrating from traditional grid to smart grid in smart cities promoted in developing country. *Sustain Energy Technol Assess*. 2021;45:101125.
<https://doi.org/10.1016/j.seta.2021.101125>.

Zhao Y, Liu J, Liu X, Yuan K, Ren K, Yang M. A graph-based deep reinforcement learning framework for autonomous power dispatch on power systems with changing topologies. In: 2022 IEEE Sustainable Power and Energy Conference (SPEC); Perth, Australia. IEEE; 2022. p. 1–5.
<https://doi.org/10.1109/iSPEC54162.2022.10033001>.

Qin L, Yu X, Luo Y, Li S, Zhou Y. Generative adversarial networks-based distribution network operation scenario generation. In: 2022 Power System and Green Energy Conference (PSGEC); Shanghai, China: IEEE; 2022. p. 900–6.
<https://doi.org/10.1109/PSGEC54663.2022.9881069>.

Wu N, Xiao J, Han S, Wu X, Gong W. Research on orderly charging of electric vehicles considering renewable energy consumption. In: 2023 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia); Chongqing, China: IEEE; 2023. p. 743–8.
<https://doi.org/10.1109/ICPSAsia58343.2023.10294925>.

Xiong L, Tang Y, Liu C, Mao S, Meng K, Dong Z. Interpretable deep reinforcement learning for optimizing heterogeneous energy storage systems. *IEEE Trans Circuits Syst I Regul Pap*. 2024;71(2).
<https://doi.org/10.1109/TCSI.2023.3340026>.

Thilagam T, Aruna R. LM-GA: A novel IDS with AES and machine learning architecture for enhanced cloud storage security. *J Mach Comput*. 2023;1(3):69–79.

Guomundsson B, Lorna G. Automated design using machine learning in materials engineering – an explicit forecast. *J Comput Intell Mater Sci*. 2023;1:56–66.

Helgo M. Deep learning and machine learning algorithms for enhanced aircraft maintenance and flight data analysis. *J Robot Spectr*. 2023;1:90–9.

Palanisamy T, Alghazzawi D, Bhatia S, Malibari AA, Dadheech P, Sengan S. Improved energy based multi-sensor object detection in wireless sensor networks. *Intell Autom Soft Comput*. 2022;33(1):227–44.
<https://doi.org/10.32604/iasc.2022.023692>.