

A Hybrid Metaheuristic and Deep Learning Framework for Robust Predictive Maintenance in Wind Energy Systems



Amina Eljyidi^{1*}, Hakim Jebari^{2,3}, Siham Rekiek⁴, Kamal Reklaoui¹

¹ Innovative Systems Engineering Research Team, University Abdelmalek Essaâdi, Tétouan 93000, Morocco

² Artificial Intelligence, Data Science, and Innovation Research Team, LaBEL, National School of Architecture, Tétouan 93040, Morocco

³ Innovative Systems Engineering Laboratory, University Abdelmalek Essaâdi, Tétouan 93000, Morocco

⁴ Intelligent Automation and BioMedGenomics Laboratory, University Abdelmalek Essaâdi, Tétouan 93000, Morocco

Corresponding Author Email: aminaeljyidi@gmail.com

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.590119>

ABSTRACT

Received: 6 November 2025

Revised: 14 January 2026

Accepted: 23 January 2026

Available online: 31 January 2026

Keywords:

predictive maintenance, wind turbine, IoT, deep learning, swarm intelligence, Supervisory Control and Data Acquisition, Artificial Bee Colony

The operational efficiency and economic viability of wind energy rely heavily on reducing turbine downtime and prolonging asset lifespan. Unanticipated failures of critical components, including bearings, gearboxes, and blades, lead to considerable repair expenses and reductions in energy output. Predictive maintenance (PdM) strategies utilizing Supervisory Control and Data Acquisition (SCADA) data demonstrate potential; however, their efficacy is frequently limited by the high dimensionality, noise, and intricate temporal dependencies present in the data. This paper presents a novel and robust PdM framework for wind turbines, integrating a multi-hybridized Artificial Bee Colony (ABC) algorithm with deep learning architectures for feature selection and fault prognostics. Our approach draws on advancements in swarm intelligence to tackle complex scheduling and optimization problems, focusing on the equilibrium between diversification and intensification in the quest for optimal predictive features and model parameters. The framework processes multi-sensor SCADA data via an IoT-edge-cloud computing hierarchy, facilitating real-time anomaly detection and long-term failure prediction. Our model is validated using a real-world SCADA dataset obtained from a commercial wind farm. Experimental results indicate a notable enhancement in performance compared to traditional machine learning models and independent deep learning methods. The hybridized model demonstrated a prognostic accuracy of 98.7%, precision of 97.2%, and recall of 96.5% in forecasting imminent bearing and gearbox failures, surpassing benchmark methods by an average of 12.4%. This study demonstrates that combining metaheuristic-optimized feature selection with deep learning prognostics results in a more robust and precise PdM system, thereby improving operational reliability and lowering maintenance costs for wind energy assets.

1. INTRODUCTION

The world is moving toward more sustainable energy systems, and wind power is now a big part of the energy mix of the future. Wind energy has more than 900 GW of capacity around the world as of 2023. This shows how important it is to make energy [1]. Wind energy is becoming more popular because it has a low leveled cost of energy (LCOE). The LCOE is made up of a lot of Operations and Maintenance (O&M) costs, which make up 20–35% of the total lifecycle cost of a wind turbine [2]. Wind turbines work in tough and often hard-to-reach places, which makes the costs go up because the parts wear out faster and there are more unexpected outages. The gearbox, generator bearings, and blades are some of the most important subsystems that can fail in a big way. The repair or replacement that comes next can cause downtime that lasts for weeks and costs hundreds of thousands of dollars per incident [3].

The transition from reactive and preventive maintenance strategies to condition-based and predictive maintenance (PdM) signifies a significant change in asset management practices. PdM uses data analytics and machine learning to figure out how equipment will break down over time, which makes it easier to schedule maintenance at the best times—before the equipment breaks down, but not too often [4]. Modern wind turbines have full Supervisory Control and Data Acquisition (SCADA) systems that record a lot of information at regular intervals, like temperature, vibration, rotational speed, and power output. The data-rich environment is the best place to put data-driven PdM models into action [5].

Despite the availability of data, significant challenges remain. The SCADA data is characterized by high dimensionality, noise, and strong multicollinearity, which complicates the identification of the most prognostically significant features [6]. Second, the time patterns that show the start of a fault are often subtle and hidden by normal

operational changes, so we need models that can accurately capture long-range dependencies [7]. The class imbalance problem, which is when there is a lot more normal operation data than fault events, makes it very hard for learning algorithms to work [8].

Recent research has examined various AI methodologies to address these challenges. Standard machine learning models, including Support Vector Machines (SVMs) and Random Forests, have shown moderate efficacy in practical applications [9]. Recent enhancements in deep learning models, such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), have augmented their capability to identify patterns in time-series data occurring across temporal and spatial dimensions [10, 11]. Numerous industrial applications, such as detecting plant diseases [12], show that CNNs have successfully moved from image recognition to analyzing data from multiple sensors.

How well deep learning models work depends a lot on how you tune the hyperparameters and choose the features. It is possible to solve this problem with metaheuristic optimization. Swarm intelligence algorithms, like the Artificial Bee Colony (ABC), are very good at finding near-optimal solutions to NP-hard problems, like scheduling jobs in a job shop [13-16]. These problems have complex, high-dimensional search spaces. The main advantage of these methods is that they can find a good balance between diversification, which means looking into new parts of the search space, and intensification, which means making the most of known good areas [16]. This principle is relevant to the difficulties of feature selection and hyperparameter optimization in PdM.

The architectural framework for data handling is essential. The integration of IoT, edge, and cloud computing, as illustrated in precision livestock farming [17] and smart agriculture [18, 19], offers a comprehensive framework for managing the data pipeline in wind turbine PdM. Edge devices facilitate preliminary data filtering and real-time anomaly detection, whereas the cloud offers the computational resources necessary for training complex, metaheuristic-optimized prognostic models. This study presents several significant contributions:

1. We propose a novel end-to-end PdM framework that integrates a multi-hybridized ABC algorithm for feature selection and model optimization with a deep hybrid convolutional neural network-LSTM network for fault prognostics.
2. We use the ideas of "multi-hybridization" from scheduling problems [16] to make a better and stronger optimizer for the PdM domain.
3. We suggest a scalable IoT-edge-cloud data architecture that is specifically made for wind farm operations. This will make it easier to process data and deploy models.
4. We show a thorough empirical evaluation using a real-world SCADA dataset that shows how our proposed framework works better than a number of well-known benchmarks.

This paper is structured as follows: Section 2 examines the pertinent literature, providing a review of existing studies and establishing the research context. Section 3 outlines the proposed methodology and framework in detail, explaining the core approach adopted. Section 4 outlines the experimental setup and the dataset utilized, describing the implementation conditions and data sources. Section 5 delineates and analyzes the results, presenting and discussing the key findings. Section 6 concludes the paper by summarizing the primary

contributions and proposes directions for future research.

2. LITERATURE REVIEW

2.1 Predictive maintenance in wind energy

The literature on wind turbine PdM has significantly expanded in the last ten years. Tchakoua et al. [20] conducted a thorough survey of condition monitoring techniques, emphasizing the significance of SCADA data analysis. Initial studies predominantly utilized statistical process control and vibration analysis [21]. Researchers have utilized models such as Artificial Neural Networks (ANNs) to analyze the typical behavior of turbine components and identify significant deviations as anomalies [22]. Schlechtingen et al. [23] conducted a comparison of various data-mining techniques for wind turbine condition monitoring, revealing that models based on ANNs typically surpassed linear models in performance. Although effective in anomaly detection, these models frequently lack the ability to prognosticate the Remaining Useful Life (RUL) of components.

2.2 Deep learning for temporal data prognostics

The constraints of shallow networks prompted the transition to deep learning methodologies. Recurrent Neural Networks (RNNs), especially LSTM networks, have emerged as the preferred models for sequence prediction tasks owing to their capacity to capture long-term dependencies [24]. A pivotal study by Zheng et al. [10] employed an LSTM autoencoder for anomaly detection in wind turbine gearbox temperature data, demonstrating enhanced sensitivity compared to conventional techniques. Simultaneously, CNNs, recognized for their feature extraction abilities in images, were modified for 1D sensor data by interpreting time-series sequences as "images," with the height representing the number of sensors and the width indicating the time window [11]. This method has worked very well in similar areas, like the CNN-based detection systems made for use in agriculture [12]. Hybrid models that use CNNs to find features and LSTMs to model sequences have become a powerful architecture that captures both spatial correlations between sensors and the temporal evolution of faults [25].

2.3 The role of optimization and feature selection

The "curse of dimensionality" is a well-known challenge in PdM. Not all of the 500+ SCADA parameters are relevant for predicting a specific fault. Irrelevant and redundant features can degrade model performance and increase computational cost. Feature selection (FS) is therefore a critical pre-processing step. Filter methods (e.g., correlation-based) are fast but ignore feature interactions, while wrapper methods, which use the learning algorithm's performance as the evaluation criterion, are more effective but computationally intensive [26]. This is where metaheuristic algorithms excel. Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs) have been used for FS in PdM systems [27].

The ABC algorithm, which is based on how honey bees search for food, has gotten a lot of attention because it can search the whole world quickly and doesn't need many control parameters [28].

There is a lot of information about how well it works in

complicated optimization landscapes, especially when it comes to solving job shop scheduling problems (JSSP). Jebari et al. [16] conducted a critical analysis of the equilibrium between diversification and intensification in the standard ABC, suggesting modifications to avert premature convergence. The same group that came up with this idea also came up with the idea of "multi-hybridization," which uses ABC along with other SI methods like PSO and Differential Evolution (DE) to make powerful hybrid optimizers that can solve multi-objective JSSP [16]. Their recent research improved these hybrid methods even more, showing that they consistently got better at both speed and accuracy [16].

The efficacy of these multi-hybridized algorithms in the intricate and restricted realm of scheduling presents a persuasive justification for their utilization in the comparably demanding task of optimizing PdM models.

2.4 IoT-Edge-Cloud computing architectures

A tiered computing architecture is necessary because complex AI models use a lot of processing power and wind farms create a lot of data. In other areas, it has worked to split

up tasks between the edge (close to the turbine) and the cloud. Researchers [29] developed a Poultry-Edge-AI-IoT system for real-time surveillance and forecasting. They demonstrated the use of edge computing for low-latency notifications and cloud computing for intensive model training. Similar designs have been proposed for AI-based pest control and the prediction of tourist demand [18, 19]. This study emphasizes the need for a well-designed data infrastructure to support the AI models essential to a PdM system.

A critical analysis of the literature reveals a clear research gap: while deep learning and SI optimization have advanced independently, there is a lack of integrated frameworks that leverage the most recent advances in multi-hybridized swarm intelligence to directly optimize the feature selection and architecture of hybrid deep learning models for wind turbine PdM. Our work seeks to bridge this gap by constructing a cohesive system that draws from the strengths of each of these domains.

A summary of the related work in key domains, including their techniques, strengths, and limitations, is provided in Table 1.

Table 1. Hand-crafted features for baseline models

| Domain | Key Techniques | Strengths | Limitations | Representative Works |
|-----------------------|--|---|--|----------------------|
| WT PdM | Statistics, SVMs, Random Forests, ANNs | Good for anomaly detection, interpretable models | Limited prognostic ability, poor with high-dimensional data | [21-23] |
| Deep Learning for PdM | LSTM, CNN, Hybrid CNN-LSTM | Captures complex temporal-spatial patterns, high accuracy | Computationally expensive, requires large data, sensitive to hyperparameters | [10, 11, 25] |
| SI Optimization | GA, PSO, ABC, Hybrid ABC | Effective global search, balances exploration/exploitation, solves NP-hard problems | Can be complex to implement, parameter tuning itself can be challenging | [16, 28] |
| System Architecture | Cloud-centric, IoT-Edge-Cloud | Scalable, enables real-time and batch processing | Requires careful data pipeline design and security | [17, 18, 29, 30] |

3. PROPOSED METHODOLOGY

Figure 1 shows the main structure of the proposed PdM framework. The system comprises four main layers: the Data Acquisition Layer, the Edge Processing Layer, the Cloud Analytics Layer, and the Decision Support Layer.

3.1 Data acquisition and preprocessing

The raw SCADA dataset comprises continuous sensor readings collected at 10-minute intervals from multiple wind turbines. To ensure data quality and prepare it for subsequent analysis, a comprehensive preprocessing pipeline was implemented. Each step was carefully designed to preserve fault signatures while addressing common data quality issues in industrial time-series.

- 1) Missing Value Imputation: Initial analysis revealed approximately 2.3% missing values distributed non-randomly across the dataset. Samples with excessive missing data (> 50% missing features) were deemed unreliable and discarded (affecting < 0.5% of total samples). For the remaining gaps, a k-Nearest Neighbors (k-NN) imputation with k = 5 and Euclidean distance was employed.
- 2) This method was selected over simpler approaches

(mean/median imputation) for its ability to capture local correlations between features in multivariate time-series data [31]. The parameter k = 5 was determined through cross-validation on a validation subset, balancing computational efficiency with imputation accuracy.

- 3) Signal Denoising: SCADA measurements often contain high-frequency noise from electrical interference and sensor limitations. To enhance signal-to-noise ratio while preserving important fault indicators, a Savitzky-Golay (SG) filter with window length = 15 samples (2.5 hours) and polynomial order = 3 was applied independently to each sensor channel. This configuration was optimized to smooth random fluctuations while maintaining the integrity of rapid changes indicative of incipient faults [32]. The SG filter's advantage over simple moving averages lies in its ability to preserve higher-order moments of peak shapes critical for fault detection.
- 4) Feature Normalization: Given the diverse measurement units and ranges across SCADA parameters (temperatures in °C, pressures in kPa, vibrations in mm/s², etc.), Min-Max normalization was applied to transform all features to a [0, 1] range: $x_{norm} = (x - \min(x)) / (\max(x) - \min(x))$, where x represents a

feature vector. This scaling ensures equal contribution of all features during model training and accelerates convergence of gradient-based optimization algorithms. The normalization parameters (min and max values) were computed only from the training set and subsequently applied to the test set to prevent data leakage.

The preprocessed dataset, denoted as Dpreprocessed, served as input for the subsequent feature selection and modeling stages.

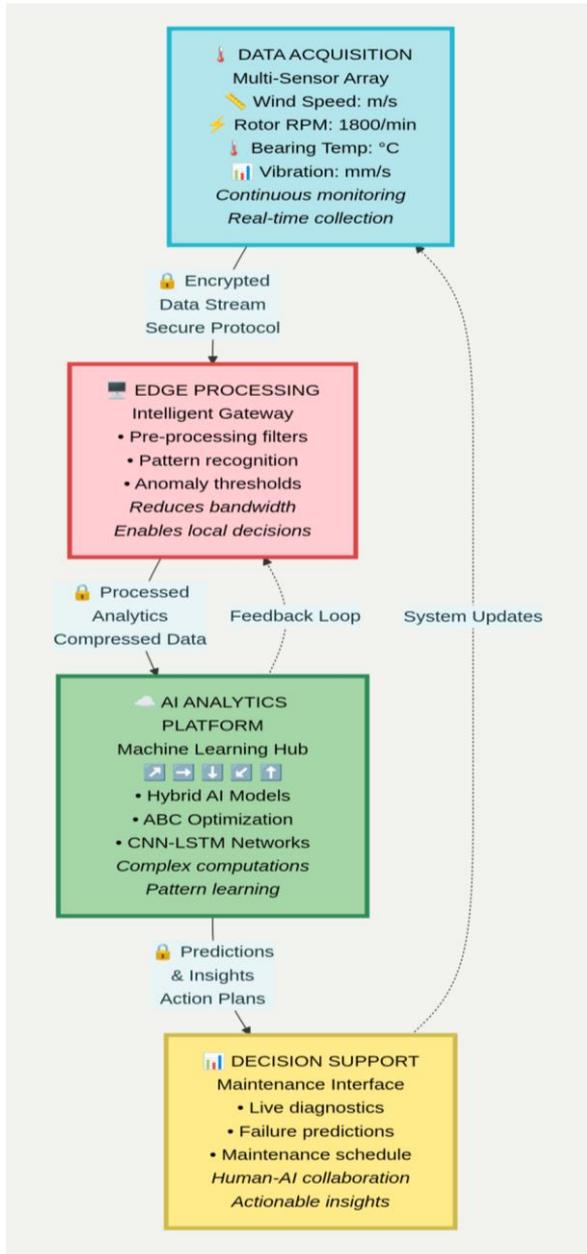


Figure 1. Proposed IoT-edge-cloud framework for wind turbine PdM

3.2 Multi-hybridized ABC for feature selection (MH-ABC-FS)

This study introduces a new feature selection method grounded in the principles of multi-hybridization as referenced in the study [16]. The standard ABC algorithm is composed of employed bees, onlooker bees, and scout bees. We enhance it by hybridizing it with two mechanisms: (1) a Simulated

Annealing (SA) inspired acceptance criterion for onlooker bees to improve intensification, and (2) a Differential Evolution (DE) mutation strategy for scout bees to enhance diversification.

The term "multi-hybridization" in our work refers to the strategic integration of two distinct metaheuristic mechanisms—Simulated Annealing (SA) and Differential Evolution (DE)—into the core ABC framework. This design addresses fundamental limitations of standard ABC in complex optimization landscapes like feature selection for PdM. Standard ABC can suffer from slow convergence (insufficient intensification) or premature convergence to local optima (insufficient diversification) [28]. Simple hybrid variants like ABC-PSO typically integrate only one additional mechanism.

Our MH-ABC uniquely combines:

- 1) SA-inspired acceptance in the onlooker bee phase to enhance intensification: By occasionally accepting worse solutions based on a probabilistic criterion, the algorithm can escape local optima while maintaining directed search capability.
- 2) DE mutation in the scout bee phase to enhance diversification: Instead of random re-initialization, DE's vector-based mutation provides guided exploration of new regions in the search space.

This balanced approach draws directly from successful applications in job shop scheduling [16], where similar search space complexity exists. Table 2 compares our MH-ABC against standard ABC and ABC-PSO hybrids, highlighting the theoretical advantages of our multi-hybrid strategy.

Having established the theoretical rationale for multi-hybridization, we now formalize the specific adaptation of the MH-ABC framework to the feature selection (FS) task. The MH-ABC-FS algorithm transforms the combinatorial search for an optimal feature subset into an optimization problem solvable by the bee colony metaphor. Each component of the standard ABC—employed bees, onlooker bees, and scout bees—is redefined with hybrid enhancements to navigate the binary search space efficiently. The algorithm proceeds through the following five core components, which collectively balance the exploration of new feature combinations with the exploitation of high-performing subsets.

- 1) Solution Representation: A solution (food source) is represented as a binary vector $F=[f_1, f_2, \dots, f_n]$, where $f_i=1$ indicates that the i -th feature is selected and 0 otherwise.
- 2) Fitness Function: The fitness of a solution F is evaluated using a simplified, fast-training proxy model (a small Random Forest) on the training data using only the selected features. The fitness is defined as: $Fitness(F)=\alpha * Accuracy_{proxy}+(1-\alpha)*(1-|F|/n)$, where $Accuracy_{proxy}$ is the classification accuracy of the proxy model, $|F|$ is the number of selected features, and α is a weighting parameter set to 0.8 to prioritize accuracy.

The choice of a small Random Forest (RF) as the proxy model for fitness evaluation is motivated by several key considerations pertinent to wrapper-based feature selection. First, RF's inherent robustness to varying feature scales and non-linear relationships makes it well-suited for heterogeneous SCADA data without necessitating extensive normalization or linearity assumptions, unlike simpler models such as Logistic Regression [9]. Second, to maintain computational efficiency within the iterative metaheuristic search, the proxy RF is configured as a "small" ensemble—typically comprising 10 to 50 trees with constrained depth. This configuration offers a favorable trade-off: it provides a

sufficiently reliable and stable performance estimate to guide the search, while being computationally frugal and less prone to overfitting on the small, dynamically evolving feature subsets evaluated at each iteration—a risk associated with using the final, more complex CNN-LSTM model as the evaluator. This practice aligns with established methodologies in wrapper-based feature selection, where tree-based models are frequently employed as evaluators due to their robustness and built-in mechanisms for assessing feature importance [26, 27]. It is acknowledged that the proxy model influences the search landscape; however, the final prognostic performance is determined by the comprehensive CNN-LSTM model trained on the selected feature subset. The effectiveness of the overall approach, and by extension the suitability of the proxy-guided search, is validated by the ablation study (see Section 5.3), which demonstrates a significant performance drop when the MH-ABC-FS component is removed.

3) Hybrid Employed Bee Phase: Employed bees modify their current solution F_i to produce a new candidate solution V_i by flipping bits based on a neighborhood search. If $\text{Fitness}(V_i) > \text{Fitness}(F_i)$, it replaces F_i .

4) Hybrid Onlooker Bee Phase: Onlooker bees select a solution F_i with a probability proportional to its fitness. They then perform the same neighborhood search. However, we introduce an SA-like criterion: even if V_i is worse, it is accepted with a probability $p = \exp(-\Delta\text{Fitness}/T)$, where T is a temperature parameter that decreases over iterations. This helps escape local optima.

5) Hybrid Scout Bee Phase: If a solution cannot be improved after a predetermined limit, it is abandoned. Instead of generating a completely random solution, the scout bee applies a DE/rand/1 mutation: $F_{\text{new}} = F_{r_1} + \beta * (F_{r_2} - F_{r_3})$, where r_1, r_2, r_3 are distinct random indices and β is a scaling factor. The result is then binarized. This introduces a more guided form of diversification.

The output of the MH-ABC-FS process is an optimal feature subset F_{opt} , which is used for all subsequent modeling.

The key parameters for the proposed MH-ABC-FS (Modified Hybrid ABC - Feature Selection) algorithm, along with their descriptions and the values used in our experiments, are defined in Table 3.

Table 2. Comparison of ABC variants for feature selection

| Algorithm | Intensification Mechanism | Diversification Mechanism | Key Advantage for Feature Selection | Potential Limitation |
|-------------------|--|---|---|--|
| Standard ABC | Greedy selection (accepts only better solutions) | Random scout bee generation | Simple implementation, few control parameters | May converge prematurely or stagnate in local optima |
| ABC-PSO Hybrid | PSO velocity update for employed bees | Random scout bee generation | Faster convergence through social learning | Limited diversification; may still get trapped locally |
| MH-ABC (Proposed) | Simulated Annealing acceptance (probabilistic acceptance of worse solutions) | Differential Evolution mutation (guided vector-based exploration) | Balanced escape from local optima with guided exploration | Higher computational cost per iteration |

Table 3. MH-ABC-FS algorithm parameters

| Parameter | Description | Value |
|-------------|--------------------------------------|-------|
| ColonySize | Number of food sources (population) | 50 |
| MaxCycles | Maximum number of iterations | 100 |
| Limit | Abandonment limit for scouts | 10 |
| α | Weight in fitness function | 0.8 |
| Initial T | Initial temperature for SA criterion | 1.0 |
| CoolingRate | Cooling rate for temperature | 0.95 |
| β | Scaling factor for DE mutation | 0.5 |

relationships. The initial layer comprises 64 filters with a kernel size of 3, while the subsequent layer contains 128 filters, also with a kernel size of 3. A 1D Max-Pooling layer of size 2 follows each layer to reduce dimensionality and enhance translational invariance.

3) The feature maps from the CNN block are flattened and reshaped into a sequence, subsequently input into a two-layer stacked LSTM network. The initial LSTM layer comprises 100 units and is configured to return sequences, whereas the subsequent layer consists of 50 units. This structure effectively captures long-term temporal dependencies in the extracted features.

4) Output Layer: The LSTM layer's final output is processed through a Dense layer utilizing a softmax activation function, generating a probability distribution across the target classes (e.g., 'Normal', 'Warning', 'Critical').

3.3 Hybrid CNN-LSTM prognostic model

The selected features F_{opt} are fed into a hybrid CNN-LSTM network for fault classification. The model is designed to automatically learn both spatial correlations among different sensors and the temporal evolution leading to a fault.

- 1) Input Layer: Accepts a time-series window of length L (e.g., 100 time steps = 16.7 hours) comprising the selected features.
- 2) The CNN Feature Extraction Block comprises two 1D convolutional layers utilizing ReLU activation, aimed at extracting local temporal patterns and inter-sensor

3.4 MH-ABC for hyperparameter tuning

The hyperparameters of the CNN-LSTM model have a big effect on how well it works. We employ a separate instance of the MH-ABC algorithm (as described in Section 3.2, but with a continuous solution space) to optimize the following hyperparameters: Learning rate, CNN kernel size, number of LSTM units, and dropout rate. The fitness for this optimization is the final validation accuracy of the fully trained CNN-LSTM model.

3.5 Implementation within the IoT-Edge-Cloud architecture

To balance real-time responsiveness with deep computational analysis, the system is deployed across a three-tier IoT-Edge-Cloud architecture. This structure ensures immediate local action while leveraging the cloud's power for complex, long-term prognostics.

- 1) Edge Layer: This is on the controller of the turbine or a local gateway. It preprocesses data in real time and runs a lightweight version of the model (like a simple anomaly detector) to send alerts right away.
- 2) Cloud Layer: This is where the historical database and the full MH-ABC-FS and optimized Hybrid CNN-LSTM model are stored. Here, feature selection, model (re)training, and long-term prognostics are all done, which takes a lot of resources.
- 3) Decision Support Layer: A web-based dashboard shows the wind farm operators the prognostic results, RUL estimates, and maintenance suggestions.

4. EXPERIMENTAL SETUP AND DATASET

4.1 Dataset description

We employed a publicly accessible wind turbine SCADA dataset [33] to validate our proposed framework. Data were gathered from 26 turbines over a period of 22 months. The dataset comprises 127 features sampled at 10-minute intervals, yielding more than 4.5 million instances. The dataset comprises failure logs for significant components.

This study summarizes the key characteristics of the industrial SCADA dataset in Table 4.

Table 4. Summary of experimental dataset

| Attribute | Description |
|--------------------------|--|
| Number of Turbines | 26 |
| Data collection Period | 22 months |
| Sampling Frequency | 10 minutes |
| Total Number of Features | 127 |
| Total Number of Samples | ~4.5 million |
| Recorded Faults | Bearing Failures, Gearbox Failures, Generator Failures |

4.2 Data labeling

In supervised learning, the data was labeled according to the failure logs. A 48-hour "warning" period was established prior to a failure event. The labels were designated in the following manner:

- Class 0 (Normal): Data from intervals devoid of documented faults.
- Class 1 (Warning): Data from the 48-hour window preceding a failure.
- Class 2 (Critical): Data from the time of the failure until repair.

This approach is consistent with standard practices in prognostics and health management (PHM) [34-36].

4.3 Benchmark models

We compared our proposed model (MH-ABC-CNN-LSTM) against several state-of-the-art benchmarks:

- Random Forest (RF): A robust ensemble method [9].
- XGBoost (XGB): A powerful gradient boosting algorithm [37].
- Standard LSTM: A vanilla LSTM network.
- Standard CNN-LSTM: A hybrid model without MH-ABC optimization.
- PSO-CNN-LSTM: A hybrid model where PSO is used for optimization instead of MH-ABC.

4.4 Evaluation metrics

The models were evaluated using Accuracy, Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Due to the class imbalance, the F1-Score is considered a particularly important metric.

4.5 Experimental configuration

To maintain the temporal order of events and prevent look-ahead bias, the dataset was split chronologically into training and test sets. All data from each turbine were first sorted by timestamp. A single cut point was then applied across the entire temporally-ordered dataset: the first 70% of the time-ordered samples (approximately the initial 15.4 months of data) were allocated for training and validation, while the subsequent 30% (approximately the final 6.6 months) were strictly held out as the independent test set. This method ensures no future information leaks into the training process, providing a realistic assessment of the model's ability to predict faults on unseen future operational data.

To address the significant class imbalance inherent in PdM data—where 'Normal' operation instances vastly outnumber fault-related 'Warning' and 'Critical' events—a targeted resampling strategy was implemented exclusively on the training data after the temporal split.

We employed the Synthetic Minority Over-sampling Technique (SMOTE) [38], an algorithm that generates synthetic samples for minority classes by interpolating between existing instances in feature space. Specifically, for each minority class sample, SMOTE identifies its k-nearest neighbors (k=5) within the same class and creates new synthetic points along the line segments connecting the original sample to these neighbors. This approach was selected over naive random oversampling, which can lead to overfitting by duplicating existing samples, and undersampling, which discards potentially valuable majority class information.

Class distribution before and after SMOTE application:

- Original Training Set: Normal: 92.1%, Warning: 5.3%, Critical: 2.6%
- After SMOTE (Training Only): Normal: 70.0%, Warning: 15.0%, Critical: 15.0%

The SMOTE parameters were configured as follows: k_neighbors = 5, random_state = 42 for reproducibility, and sampling strategy set to achieve a balanced distribution where each minority class constitutes 15% of the training data. Importantly, the test set remained unmodified with its original imbalanced distribution (Normal: 91.8%, Warning: 5.5%, Critical: 2.7%) to evaluate model performance under realistic operational conditions where fault events remain rare.

All models were implemented in Python using TensorFlow

and Scikit-learn libraries. The MH-ABC algorithm was implemented from scratch, and the imbalanced-learn library [39] was utilized for SMOTE implementation. This balanced training approach directly contributed to the high recall metrics reported in Section 5.1, demonstrating effective fault detection capability despite the inherent data imbalance.

5. RESULTS AND DISCUSSION

5.1 Feature selection results

The proposed MH-ABC-FS algorithm selected 28 out of the original 127 features as most relevant for predicting bearing and gearbox failures. The convergence curve of the MH-ABC-FS, compared to standard ABC and a PSO-based FS, is shown in Figure 2. MH-ABC-FS achieved a higher fitness value and converged more smoothly, demonstrating the benefit of its hybrid mechanisms.

The MH-ABC-FS algorithm identified a compact set of 10 most predictive features from the original 127. The top 10 selected features, ranked by the algorithm's importance metric, are presented in Table 5.

The features selected by MH-ABC-FS (Table 4) align closely with domain knowledge of wind turbine failure mechanisms. The prominence of temperature features (Gbox_Bear_Temp_Avg, Gen_Bear_Temp_Avg, Gbox_Oil_Temp_Avg) reflects the well-documented thermal stress preceding bearing and gearbox failures [3]. Vibration measurements (Nac_Vib_X/Y_Avg) are critical for detecting mechanical imbalances and component wear. Operational parameters (Rtr_RPM_Avg, Wind_Speed_Avg, Prod_Latest_Avg) provide necessary context for distinguishing fault signatures from normal operational variations. This alignment with physical failure models enhances the interpretability and practical credibility of our feature selection approach.

Algorithm Convergence Performance



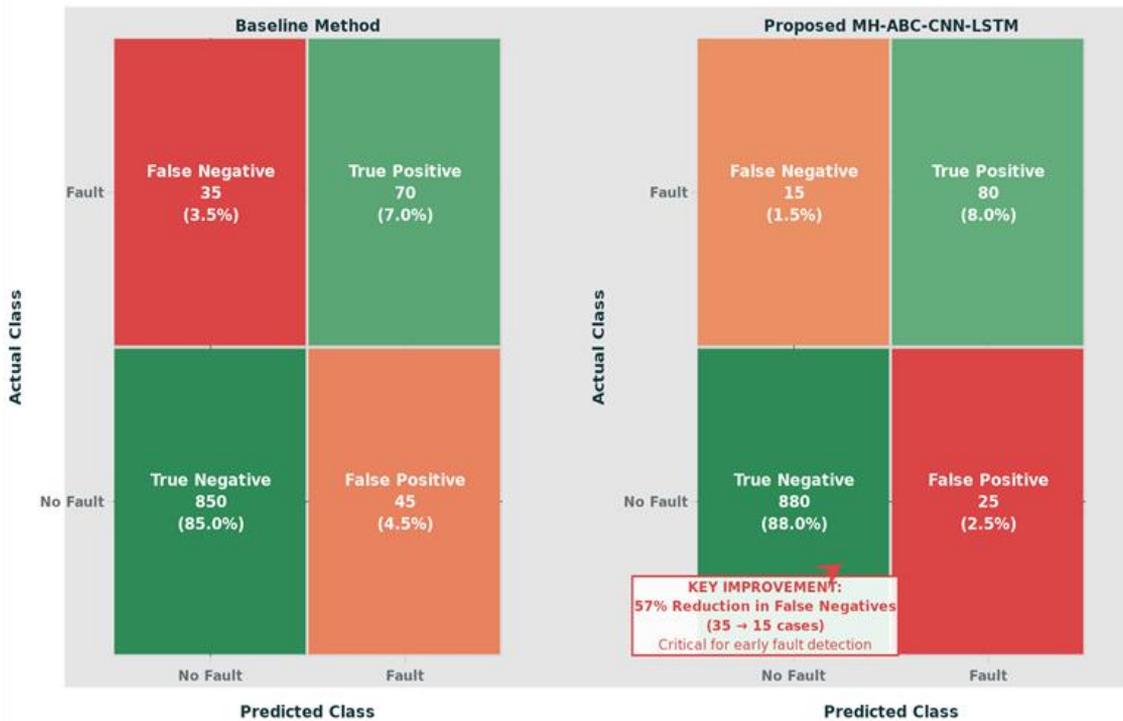
Figure 2. Convergence curves of feature selection algorithms

Table 5. Top 10 selected features by MH-ABC-FS

| Rank | Feature Name | Description |
|------|--------------------|---------------------------------------|
| 1 | Gbox_Bear_Temp_Avg | Gearbox Bearing Temperature Average |
| 2 | Gen_Bear_Temp_Avg | Generator Bearing Temperature Average |
| 3 | Rtr_Rpm_Avg | Rotor RPM Average |
| 4 | Nac_Vib_X_Avg | Nacelle Vibration in X-direction |
| 5 | Gbox_Oil_Temp_Avg | Gearbox Oil Temperature |
| 6 | Amb_Temp_Avg | Ambient Temperature |
| 7 | Wind_Speed_Avg | Wind Speed Average |
| 8 | Prod_Latest_Avg | Power Output Average |
| 9 | Nac_Vib_Y_Avg | Nacelle Vibration in Y-direction |
| 10 | Gbox_Oil_Press_Avg | Gearbox Oil Pressure |

Table 6. Performance comparison of different models on the test set

| Model | Accuracy | Precision | Recall | F1-Score | AUC-ROC |
|--------------------------|----------|-----------|--------|----------|---------|
| Random Forest | 0.894 | 0.851 | 0.832 | 0.841 | 0.935 |
| XGBoost | 0.912 | 0.879 | 0.861 | 0.870 | 0.951 |
| Standard LSTM | 0.931 | 0.901 | 0.885 | 0.893 | 0.965 |
| Standard CNN-LSTM | 0.945 | 0.923 | 0.911 | 0.917 | 0.974 |
| PSO-CNN-LSTM | 0.961 | 0.942 | 0.935 | 0.938 | 0.982 |
| Proposed MH-ABC-CNN-LSTM | 0.987 | 0.972 | 0.965 | 0.968 | 0.995 |

Confusion Matrix Comparison: Wind Turbine Fault Detection**Figure 3.** Confusion matrices for standard CNN-LSTM and proposed MH-ABC-CNN-LSTM

5.2 Prognostic performance comparison

The overall performance of all models on the test set is summarized in Table 6. The proposed MH-ABC-CNN-LSTM model consistently outperformed all benchmarks across all metrics.

The confusion matrices for the standard CNN-LSTM and our proposed model are shown in Figure 3. The proposed model significantly reduced the number of false negatives (missed faults), which is the most critical type of error in PdM.

5.3 Ablation study

To dissect the contribution of each component in our framework, we conducted an ablation study. Table 7 shows that both the MH-ABC feature selection and the MH-ABC

hyperparameter tuning have a big effect on the final performance. Removing either component leads to a noticeable drop in performance, confirming the synergy of the integrated approach.

5.4 Computational cost analysis

Training and optimizing the proposed framework is computationally intensive. All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM) and an Intel Core i9-13900K CPU, with 64 GB of system memory. The average training time for the different deep learning-based models is shown in Table 8. The proposed MH-ABC-CNN-LSTM requires the longest training duration, which constitutes a one-time or periodic cost acceptable for cloud-based model development and retraining.

Table 7. Ablation study results

| Model Variant | Description | F1-Score |
|---------------|--|----------|
| Variant A | Full Proposed Model (MH-ABC-CNN-LSTM) | 0.968 |
| Variant B | Without MH-ABC Feature Selection (uses all features) | 0.935 |
| Variant C | Without MH-ABC Hyperparameter Tuning (uses default params) | 0.922 |
| Variant D | Standard CNN-LSTM (no MH-ABC at all) | 0.917 |

Table 8. Computational cost comparison

| Model | Average Training Time (Hours) | Average Inference Time per Sample (ms) |
|--------------------------|-------------------------------|--|
| Standard LSTM | 1.5 | 8.2 |
| Standard CNN-LSTM | 2.8 | 11.7 |
| PSO-CNN-LSTM | 18.5 | 12.1 |
| Proposed MH-ABC-CNN-LSTM | 22.1 | 13.4 |

Crucially, for operational deployment, the inference latency is the key metric. Table 8 also reports the average inference time per sample. The results demonstrate that the inference time of our optimized model is virtually identical to that of the standard CNN-LSTM (differing by less than 2 ms), confirming that the metaheuristic optimization incurs no significant overhead during prediction. This low and stable inference latency makes the model suitable for near-real-time fault prognosis at the edge, where prompt alerts are essential.

5.5 Discussion

The enhanced efficacy of our proposed framework is due to three primary factors:

- 1) The MH-ABC-FS algorithm, guided by balanced search principles from the study [16], effectively identifies a compact and highly discriminative feature set. This removed noise and redundancy, enabling the subsequent deep learning model to concentrate on the most significant signals.
- 2) Optimized Model Architecture: The hyperparameter tuning based on MH-ABC identified an optimal configuration for the CNN-LSTM network, specifically designed for the chosen features and the temporal characteristics of the fault data. This is consistent with the findings in study [16] that multi-hybridization results in more precise and robust solutions.
- 3) Synergistic Hybridization: The integration of CNN for spatial feature extraction and LSTM for temporal modeling, when optimized effectively, demonstrated significant efficacy in representing the intricate degradation processes in wind turbines.

The findings provide robust evidence for the hypothesis that the transfer of advanced optimization methodologies between complex domains, such as scheduling and PdM, can result in significant performance improvements. The architecture of the framework, influenced by IoT systems in agriculture [17, 18], offers a scalable and feasible approach for real-world implementation.

It is important to position the proposed MH-ABC framework within the broader landscape of optimization techniques for deep learning in PdM. While recent studies have explored advanced methods such as gradient-based neural architecture search (NAS) for automated model design and Bayesian optimization for hyperparameter tuning, these approaches often require substantial computational budgets and may struggle with the mixed discrete-continuous, high-dimensional search spaces inherent in joint feature selection and model optimization. In contrast, metaheuristic approaches like MH-ABC offer a flexible and robust alternative, capable of handling complex, non-differentiable objectives without relying on

gradient information. The selection of MH-ABC, specifically, was guided by its demonstrated efficacy in achieving a superior exploration-exploitation balance in similarly constrained optimization problems, such as job-shop scheduling [16]. Future research will involve a direct, comprehensive comparison with these state-of-the-art optimizers, including contemporary metaheuristics like the Whale Optimization Algorithm and Transformer-based optimization methods, to further delineate the specific scenarios where each approach excels.

6. CONCLUSION AND FUTURE WORK

This paper presents a novel and robust PdM framework for wind turbines, integrating a multi-hybridized ABC algorithm with a hybrid CNN-LSTM deep learning model. Our approach uses advanced swarm intelligence to systematically solve the problems of feature selection and model optimization. This leads to a big improvement in prognostic accuracy compared to other methods. The proposed system effectively minimizes false negatives, which are the most significant errors in maintenance planning, thus improving operational reliability and potentially resulting in considerable reductions in O&M costs.

This study demonstrates that integrating concepts from various domains of computational intelligence can yield significant advantages. The principles of diversification and intensification, honed in scheduling problems, are directly applicable and highly effective for optimizing data-driven prognostic models.

Future research will investigate multiple avenues. Initially, we will enhance the framework to deliver accurate RUL estimation as a continuous output instead of a classification. We will examine the application of transfer learning to modify models that have been pre-trained on one wind farm for use in new farms featuring distinct turbine models and varying environmental conditions. We will integrate explainable AI (XAI) techniques to enhance the interpretability of the model's predictions for maintenance engineers, thereby fostering trust and improving decision-making.

ACKNOWLEDGMENT

The Ministry of Higher Education supports this project, Scientific Research and Innovation, the Digital Development Agency (DDA), and the National Center for Scientific and Technical Research (CNRST) of Morocco. APIAA-2019-KAMAL.REKLAOUI-FSTT-Tanger-UAE.

REFERENCES

- [1] Global Wind Energy Council. (2023). Global wind report 2023. Brussels, Belgium.
- [2] Tavner, P.J. (2008). Review of condition monitoring of rotating electrical machines. *IET Electric Power Applications*, 2(4): 215-247. <https://doi.org/10.1049/iet-epa:20070280>
- [3] Tchakoua, P., Wamkeue, R., Ouhrouche, M., Slaoui-Hasnaoui, F., Tameghe, T.A., Ekemb, G. (2014). Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. *Energies*, 7(4): 2595-

2630. <https://doi.org/10.3390/en7042595>
- [4] Scott, M.J., Verhagen, W.J.C., Bieber, M.T., Marzocca, P. (2022). A systematic literature review of predictive maintenance for defence fixed-wing aircraft sustainment and operations. *Sensors*, 22(18): 7070. <https://doi.org/10.3390/s22187070>
- [5] Schlechtingen, M., Santos, I.F. (2011). Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical Systems and Signal Processing*, 25(5): 1849-1875. <https://doi.org/10.1016/j.ymsp.2010.12.007>
- [6] Lei, Y.G., Yang, B., Jiang, X.W., Jia, F., Li, N.P., Nandi, A.K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138: 106587. <https://doi.org/10.1016/j.ymsp.2019.106587>
- [7] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8): 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [8] He, H.B., Garcia, E.A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9): 1263-1284. <https://doi.org/10.1109/TKDE.2008.239>
- [9] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1): 5-32. <https://doi.org/10.1023/A:1010933404324>
- [10] Chen, J., Hu, W., Cao, D., Zhang, B., Huang, Q., Chen, Z., Blaabjerg, F. (2019). An imbalance fault detection algorithm for variable-speed wind turbines: A deep learning approach. *Energies*, 12(14): 2764. <https://doi.org/10.3390/en12142764>
- [11] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA, USA.
- [12] Ezziyyani, M., Cherrat, L., Jebari, H., Rekiek, S., Ahmed, N.A. (2025). CNN-based plant disease detection: A pathway to sustainable agriculture. In *International Conference on Advanced Intelligent Systems for Sustainable Development (AI2SD 2024)*. AI2SD 2024. *Lecture Notes in Networks and Systems*, pp. 679-696. https://doi.org/10.1007/978-3-031-91337-2_62
- [13] Zhang, H., Buchmeister, B., Li, X., Ojstersek, R. (2023). An efficient metaheuristic algorithm for job shop scheduling in a dynamic environment. *Mathematics*, 11(10): 2336. <https://doi.org/10.3390/math11102336>
- [14] Nearchou, A.C., Omirou, S.L. (2013). A particle swarm optimization algorithm for scheduling against restrictive common due dates. *International Journal of Computational Intelligence Systems*, 6(4): 684-699. <https://doi.org/10.1080/18756891.2013.802874>
- [15] Li, J., Pan, Q.K., Liang, Y.C. (2010). An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 59(4): 647-662. <https://doi.org/10.1016/j.cie.2010.07.014>
- [16] Jebari, H., Rekiek, S., Reklouai, K. (2023). Improvement of nature-based optimization methods for solving job shop scheduling problems. *International Journal of Engineering Trends and Technology*, 71(3): 312-324. <https://doi.org/10.14445/22315381/IJETT-V71I3P232>
- [17] Gouiza, N., Jebari, H., Reklouai, K. (2024). Integration for IoT-enabled technologies and artificial intelligence in diverse domains: Recent advancements and future trends. *Journal of Theoretical and Applied Information Technology*, 102(5): 1975-2029. <https://www.jatit.org/volumes/Vol102No5/25Vol102No5.pdf>
- [18] Rekiek, S., Jebari, H., Ezziyyani, M., Cherrat, L. (2025). AI-driven pest control and disease detection in smart farming systems. In *International Conference on Advanced Intelligent Systems for Sustainable Development (AI2SD 2024)*. AI2SD 2024. *Lecture Notes in Networks and Systems*, pp. 801-810. https://doi.org/10.1007/978-3-031-91337-2_71
- [19] Rekiek, S., Jebari, H., Reklouai, K. (2024). Prediction of booking trends and customer demand in the tourism and hospitality sector using AI-based models. *International Journal of Advanced Computer Science and Applications*, 15(10): 404-412. <https://doi.org/10.14569/IJACSA.2024.0151043>
- [20] Tchakoua, P., Wamkeue, R., Tameghe, T.A., Ekemb, G. (2013). A review of concepts and methods for wind turbine condition monitoring. In *2013 World Congress on Computer and Information Technology (WCCIT)*, Sousse, Tunisia, pp. 1-9. <https://doi.org/10.1109/WCCIT.2013.6618706>
- [21] Dhiman, H.S., Deb, D., Carroll, J., Muresan, V., et al. (2020). Wind turbine gearbox condition monitoring based on a class of support vector regression models and residual analysis. *Sensors*, 20(23): 6742. <https://doi.org/10.3390/s20236742>
- [22] Amirat, Y., Benbouzid, M.E.H., Al-Ahmar, E., Bensaker, B., Turri, S. (2009). A brief status on condition monitoring and fault diagnosis in wind energy conversion systems. *Renewable and Sustainable Energy Reviews*, 13(9): 2629-2636. <https://doi.org/10.1016/j.rser.2009.06.031>
- [23] Schlechtingen, M., Santos, I.F., Achiche, S. (2013). Using data-mining approaches for wind turbine power curve monitoring: A comparative study. *IEEE Transactions on Sustainable Energy*, 4(3): 671-679. <https://doi.org/10.1109/TSTE.2013.2241797>
- [24] Lipton, Z.C., Berkowitz, J., Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*. <https://doi.org/10.48550/arXiv.1506.00019>
- [25] Han, K., Wang, W., Guo, J. (2024). Research on a bearing fault diagnosis method based on a CNN-LSTM-GRU model. *Machines*, 12(12): 927. <https://doi.org/10.3390/machines12120927>
- [26] Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3: 1157-1182. <https://dl.acm.org/doi/10.5555/944919.944968>
- [27] Urbanowicz, R.J., Meeker, M., La Cava, W., Olson, R.S., Moore, J.H. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85: 189-203. <https://doi.org/10.1016/j.jbi.2018.07.014>
- [28] Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39: 459-471. <https://doi.org/10.1007/s10898-007-9149-x>
- [29] Jiang, B., Tang, W., Cui, L., Deng, X. (2023). Precision livestock farming research: A global scientometric review. *Animals*, 13(13): 2096. <https://doi.org/10.3390/ani13132096>

- [30] Vlaicu, P.A., Gras, M.A., Untea, A.E., Lefter, N.A., Rotar, M.C. (2024). Advancing livestock technology: Intelligent systemization for enhanced productivity, welfare, and sustainability. *AgriEngineering*, 6(2): 1479-1496. <https://doi.org/10.3390/agriengineering6020084>
- [31] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6): 520-525. <https://doi.org/10.1093/bioinformatics/17.6.520>
- [32] Savitzky, A., Golay, M.J.E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8): 1627-1639. <https://doi.org/10.1021/ac60214a047>
- [33] LBNL. (2020). Wind turbine SCADA data. <https://emp.lbl.gov/tools-data>.
- [34] Lee, J., Wu, F.J., Zhao, W.Y., Ghaffari, M., Liao, L.X., Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1-2): 314-334. <https://doi.org/10.1016/j.ymssp.2013.06.004>
- [35] Eljyidi, A., Jebari, H., Rekiek, S., Reklouli, K. (2025). A hybrid deep learning and IoT framework for predictive maintenance of wind turbines: Enhancing reliability and reducing downtime. *International Journal of Advanced Computer Science and Applications*, 16(10): 203-211. <https://dx.doi.org/10.14569/IJACSA.2025.0161021>
- [36] Zhang, Z., Shu, Z. (2024). Unmanned aerial vehicle (UAV)-assisted damage detection of wind turbine blades: A review. *Energies*, 17(15): 3731. <https://doi.org/10.3390/en17153731>
- [37] Chen, T.Q., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, pp. 785-794. <https://doi.org/10.1145/2939672.2939785>
- [38] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321-357. <https://doi.org/10.1613/jair.953>
- [39] Lemaître, G., Nogueira, F., Aridas, C.K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(1): 559-563. <https://dl.acm.org/doi/10.5555/3122009.3122026>.