Machine Fault Diagnosis Using Hybrid CNN–LSTM Deep Learning: A Detailed Examination

Satar Jabbar Kadhim
Missan Oil Training Institute, Ministry of Oil, Iraq

Received: 19 Jan 2026 | Received Revised Version: 24 Feb 2026 | Accepted: 11 Mar 2026 | Published: 2 Apr 2026

Volume 08 Issue 04 2026 | Crossref DOI: 10.37547/tajet/Volume08Issue04-02

Abstract

Industry 4.0 has increased the demand for intelligent predictive maintenance systems capable of supporting real-time monitoring, early fault detection, and efficient decision-making in industrial environments. In this context, accurate prediction of machine failures has become essential for minimizing downtime, reducing maintenance costs, and improving operational reliability. This study employs the Predictive Maintenance Dataset from the UCI repository to develop and evaluate data-driven models for machine failure prediction and classification. The research pursues two primary objectives: first, to compare the performance of several machine learning algorithms in classifying machine failures, and second, to assess the effectiveness of deep learning approaches in achieving higher predictive accuracy. Among the machine learning models examined, the XGBoost classifier demonstrates the strongest performance. To further enhance prediction capability, this study adopts a hybrid Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM) model, which integrates CNN's strength in automatic feature extraction with LSTM's ability to learn temporal dependencies from sequential data. Experimental results show that the proposed CNN–LSTM model outperforms traditional machine learning models as well as Artificial Neural Networks (ANN) in predicting machine failures. The main contribution of this study lies in the comparative evaluation of machine learning and hybrid deep learning techniques on an imbalanced predictive maintenance dataset. The findings confirm the potential of hybrid deep learning models for predictive maintenance applications and highlight their practical value in enabling proactive maintenance strategies, optimizing resource allocation, and enhancing the reliability of smart industrial systems.

Keywords: *Deep Learning, Industry 4.0, Machine Fault Diagnosis, Predictive maintenance, CNN, and LSTM*

Cite This Article: Kadhim, S. J. (2026). Machine Fault Diagnosis Using Hybrid CNN–LSTM Deep Learning: A Detailed Examination. The American Journal of Engineering and Technology, 8(4), 12–25. https://doi.org/10.37547/tajet/Volume08Issue04-02

1. Introduction

In recent years, the ever-increasing demand for safe, efficient and cost-effective operation of various industrial processes has led to the development of increasingly complex and technologically advanced machinery. With such an emphasis placed upon the ability of modern machinery to execute a wide variety of tasks, there exists a corresponding risk that such machinery will experience malfunctions and/or fail [1], [2]. In addition to the direct impacts of such malfunctions and failures (e.g., loss of productivity, compromised safety), it also directly impacts the overall system performance. Consequently, identifying faults in a timely and accurate manner becomes essential for preventing the aforementioned issues and performing proactive maintenance to minimize time lost due to equipment failure or malfunction and associated costs [3].

Traditionally, fault diagnosis methods have employed a combination of knowledge-based systems, rule-based systems and signal processing techniques. Although such approaches have been successful in certain applications, they possess inherent limitations when attempting to diagnose faults that are complex, noisy, small in size, subject to changing operating conditions, and evolving in terms of machine characteristics [4],[5]. Artificial Intelligence (AI) and Machine Learning (ML) have recently opened new avenues for developing fault diagnosis solutions that utilize data to identify complex relationships and make predictive decisions [6].

The field of machine fault diagnosis has seen a steady evolution of methodologies and techniques used to diagnose faults in machinery. Such methodologies and techniques were developed through the creativity and imagination of researchers and engineers. Methods and techniques for diagnosing faults in machinery have generally been categorized into four broad categories including: (1) physically-model based methods; (2) signal processing-based methods; (3) Machine Learning (ML)-based methods; and (4) hybrid-methodologies [7]. Physically-model based methodologies rely heavily on a detailed understanding of the internal workings of the machinery being diagnosed. Creating accurate physically-based models for modern complex machinery, especially in dynamic and noisy environments, is a major challenge. In addition, physically-based models are typically difficult to update using real-time monitoring data [6]. Signal processing-based methodologies focus on enhancing the quality of the information related to the diagnostic characteristics of faults by applying advanced noise-reduction filtering techniques. The primary limitation of this methodology is that it requires an excellent understanding of the frequency characteristics of the relevant equipment. Furthermore, establishing a robust theoretical foundation and mathematical framework for the representation of faults is critical to the successful application of this methodology [8]. ML-based methodologies represent data-driven approaches that have gained tremendous popularity in the modern industrial environment. Classical Machine Learning models, which include Support Vector Machines (SVMs) and K-Nearest Neighbors (KNN), have achieved remarkable results. Nevertheless, both types of classical models are limited by their inability to address many of the requirements of modern industry. For example, the requirement to manually extract/ select features from data hampers the efficacy of both classical models in analyzing large amounts of complex data. Moreover, their shallow structures limit their ability to thoroughly search for and identify high dimensional features [10]. The separation of feature-mining and decision-making processes result in inefficient optimization leading to suboptimal performance. With increasing numbers of sensors and increasing complexity of machines, traditional algorithms cannot produce satisfactory results due to increasing dimensions of data and dynamics. Hybrid methodologies seek to combine the best of multiple methodologies (knowledge-based methodologies, such as physical models, and data-based methodologies, such as Machine Learning). Combining the strengths of these methodologies and incorporating the analytical capabilities of signal processing provides the most promising approach to resolving the challenging problems of machine fault diagnosis in modern industrial environments.

Deep learning, a rapidly developing area of Machine Learning, has shown immense expansion and success in many areas such as Image Recognition [11] and Language Processing [12]. These developments are due to a combination of several aspects. For example, Deep Learning's intrinsic ability to learn from large datasets and discover the most complex features, along with new and innovative architectures have contributed to an increasing rate of use of this method. The explosion of Industrial Big Data, improvements in Hardware, Internet of Things (IoT) [13], and the need for Intelligent Solutions in all sectors, have accelerated the development of Deep Learning [14]. As a result of these advances, the area of machine fault diagnosis has been impacted as well. Over the last five years, Deep Learning has revolutionized the field of Fault Diagnosis [6] and led to the creation of a variety of models that include; Deep Auto Encoders, Deep Belief Networks (DBN), Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM).

This study utilizes the Predictive Maintenance Dataset from the UCI repository to test and evaluate data-based models for classifying and predicting machine failure. The overall goals of the research are to comparatively examine how several Machine Learning algorithms can classify machine failure, and to determine if Deep Learning methods can achieve better predictive capabilities than the Machine Learning algorithms. Of all the Machine Learning models tested, the XGBoost classifier exhibited the best performance. Additionally, this study utilizes a hybrid Convolutional Neural Network- Long Short-Term Memory (CNN-LSTM) model, to provide greater predictive capability by combining the strengths of CNN's automated feature extraction, and LSTM's ability to learn sequential data. Results indicate that the proposed CNN-LSTM model will be superior to both Machine Learning models and Artificial Neural Networks (ANN), in terms of predictive capability, when it comes to machine failure. The major contribution of this study is to comparatively evaluate Machine Learning and Hybrid Deep Learning Methods using an unbalanced Predictive Maintenance Dataset. The results of this study also demonstrate the feasibility of Hybrid Deep Learning Models for Predictive Maintenance, and their applicability in implementing proactive maintenance strategies, in optimizing resource allocations, and in increasing the reliability of Smart Industrial Systems.

The remainder of this paper is organized as follows: Section 2 explains the proposed research methodology, Section 3 presents the results and discussion, and Section 4 provides the conclusion and outlines future directions related to this work.

2. The Proposed Method

This research introduces a predictive maintenance model for forecasting the occurrence of failures in machines by applying a secondary manufacturing data set received from the University of California at Irvine (UCI) repository. The primary goal of this research was to classify machine states into two types: failure and no-failure; and to evaluate the predictive abilities of traditional machine learning algorithms versus those using Deep Learning architectures that are typically used when dealing with datasets where one class has significantly more samples than another. The research applied the CNN-LSTM hybrid architecture as a means of enhancing the ability to extract features and perform sequential representation learning without changing the remaining parts of the original method. Figure 1 depicts the proposed CNN-LSTM research model.

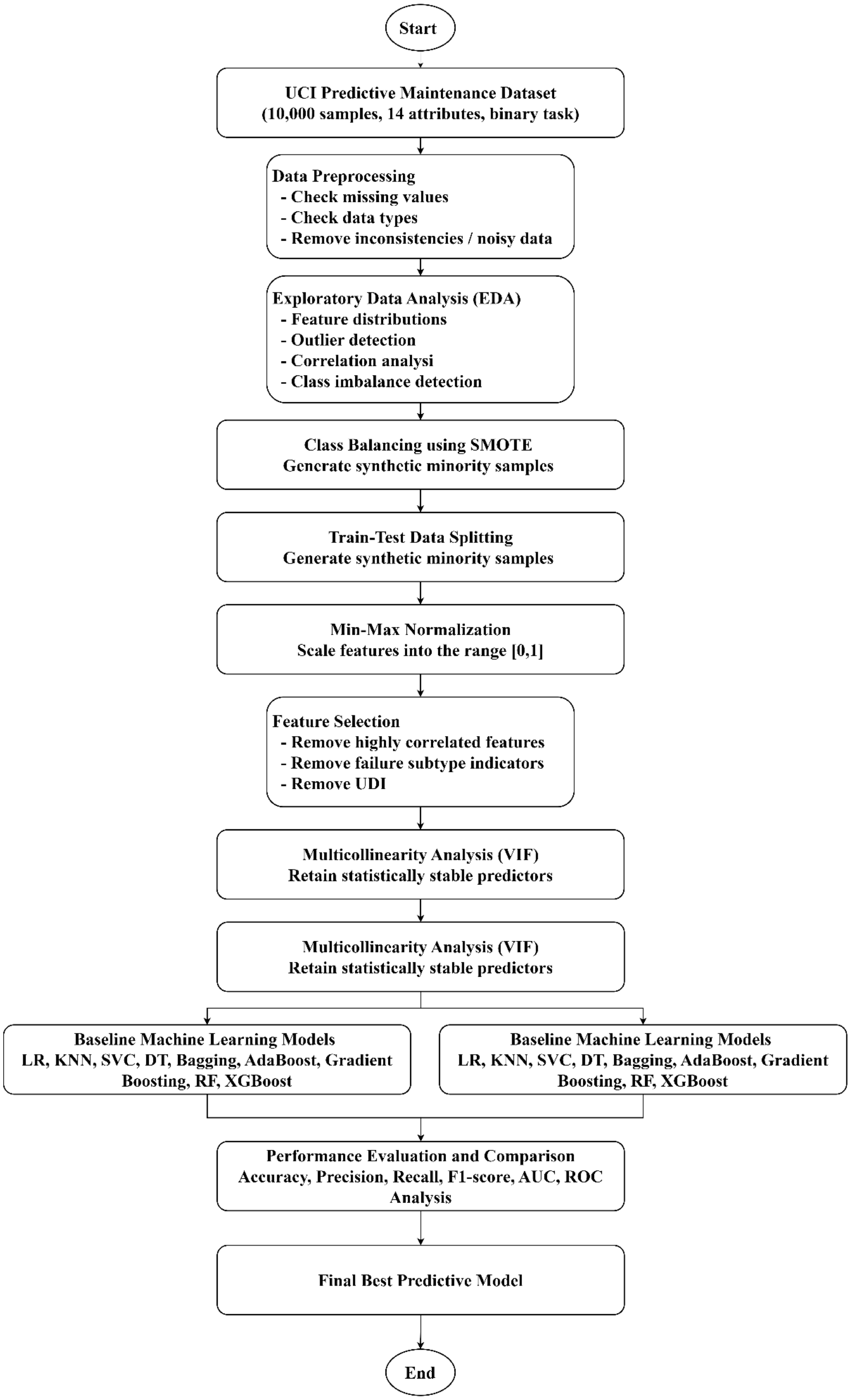

Figure 1. The proposed CNN-LSTM research model.

2.1. Dataset description and problem formulation

This data set contains 10000 records with 14 attributes for the machine operation conditions, quality type, thermal properties, rotation properties, torque, wear condition and also the label of machine-failure. In fact, the authors explain that they created this data-set to be an imitation of predictive maintenance data in a production facility. It includes such fields as UDI [15], Product Type, Air Temp., Process Temp., Rot. Speed., Torque, Tool Wear, Machine Failure and Five sub-type failure indicator fields: TWF, HDF, PWF, OSF, and RNF. Let the dataset be denoted as

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \tag{1}$$

where $x_i \in \mathbb{R}^d$represents the feature vector of the i-th observation, $y_i \in \{0,1\}$is the class label, $N = 10000$, and $d = 14$before feature reduction.

The task is formulated as a binary classification problem, since the response variable has only two outcomes: machine failure or machine non-failure. This is consistent with the source paper, which explicitly states that supervised learning was chosen because the output variable is categorical and the problem is therefore a binary classifier. The target label is written as

$$y_i = \begin{cases} 1, & \text{if the machine fails} \\ 0, & \text{if the machine does not fail} \end{cases} \tag{2}$$

This formulation provides the foundation for all later modelling stages. In practical terms, the classifier learns a mapping from measured machine-condition variables to a failure decision, allowing the system to provide an early warning signal before a breakdown occurs.

2.2. Data preprocessing and exploratory data analysis

The proposed study emphasizes that unprocessed data is rarely usable as it is for use in machine learning and deep learning applications. The data can contain a variety of inconsistencies, noise, or inappropriate data types, as well as unnecessary information in the form of redundant or irrelevant patterns which can severely impact how well a model performs. Therefore, prior to training, the raw data undergoes some form of processing, specifically referred to as preprocessing. This preprocessing phase includes data cleansing (i.e., removing unwanted or invalid data), determining whether any values are missing, identifying and/or replacing noisy data, validating whether the appropriate type of data has been collected for each variable, and ultimately prepares the data for both statistical and predictive analysis.

The exploratory analysis reported in the paper shows that the dataset has shape $(10000, 14)$, contains no missing values, and includes one object-type feature for machine type, several numerical predictors, and a binary target variable. The paper further identifies five continuous variables, namely air temperature, process temperature, rotational speed, torque, and tool wear. Outlier inspection shows that the most noticeable outliers appear in rotational speed and torque, whereas the remaining variables are relatively stable. The target count plot also reveals severe class imbalance, since machine-failure cases are much fewer than non-failure cases.

rom a statistical perspective, outlier detection may be described using the interquartile range. If $Q_1$and Q_3denote the first and third quartiles, then

$$IQR = Q_3 - Q_1 \tag{3}$$

and the lower and upper bounds are

$$LB = Q_1 - 1.5(IQR) \tag{4}$$
$$UB = Q_3 + 1.5(IQR) \tag{5}$$

Any value outside $[LB, UB]$may be treated as an outlier candidate. The paper notes that outliers were addressed through this general logic during preprocessing. The purpose of EDA here is not only descriptive. It also guides later modelling decisions. For example, identifying strong imbalance motivates resampling; identifying correlated features motivates feature selection; and identifying outliers motivates normalization and robustness checks. In this sense, EDA functions as both a diagnostic and a design step in the proposed predictive pipeline.

2.3 Class imbalance treatment using SMOTE

A central challenge in this dataset is that the minority class, namely machine failure, is strongly underrepresented. The source paper explains that machine learning algorithms often perform poorly or yield misleadingly optimistic results when trained on highly imbalanced datasets because they tend to favor the majority class. To address this issue, the study considers three data-sampling strategies: up-sampling, down-sampling, and SMOTE [16], and it concludes from the literature review and modelling framework that SMOTE is the preferred method for this dataset.

SMOTE, or Synthetic Minority Over-sampling Technique, creates artificial minority samples instead of merely duplicating existing ones. This helps the classifier observe a broader and smoother representation of the minority region in the feature space. A synthetic sample is generated by interpolation between an original minority sample and one of its nearest minority neighbors:

$$X_{new} = \mathrm{xi} + \lambda(X_{zi} - X_i), \qquad 0 \leq \lambda \leq 1 \tag{6}$$

where $x_i$is a minority-class sample, $x_{zi}$is one of its nearest minority neighbors, and $\lambda$is a random scalar that controls the interpolation location.

The practical value of SMOTE in this study is that it reduces model bias toward the majority class and gives later classifiers a better chance of learning the boundary that separates failure from non-failure. Since predictive maintenance is especially sensitive to missed failures, balancing the dataset is not merely a technical option; it is a necessary step for meaningful classification performance.

2.4 Data splitting strategy

After balancing and basic data preparation, the dataset is partitioned into training and testing subsets. This paper uses a 4:1 split ratio, resulting in a training set of size (8000,13) and a testing set of size (2000,13) before the final reduced-feature stage. The study also states that only the training portion is used to construct and optimize the learning models.

Mathematically, the partition can be written as

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test} \tag{7}$$

subject to

$$\mathcal{D}_{train} \cap \mathcal{D}_{test} = \emptyset \tag{8}$$

and

$$\frac{|\mathcal{D}_{train}|}{|\mathcal{D}_{test}|} = \frac{4}{1} \tag{9}$$

This split ensures that the model is trained on a sufficiently large sample while still reserving unseen data for objective evaluation. In predictive maintenance studies, this separation is important because performance assessed only on training data can overestimate the true utility of the classifier in real industrial settings.

2.5 Feature normalization

The source paper states that each feature has a distinct numerical range and therefore adopts min–max normalization to rescale the data before training. This is a standard and necessary step because variables such as temperature, speed, torque, and wear are measured in different units and magnitudes. Without normalization, features with larger numeric ranges could dominate the learning process, especially in distance-based and gradient-based algorithms

The normalization formula used in this study is [17]

$$X_n = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{10}$$

where $X_n$is the normalized value, $X$is the original value, and $X_{\min}$and $X_{\max}$are the minimum and maximum values of that feature.

This transformation maps each variable to the interval $[0, 1]$, making the input space more numerically consistent. In the modified CNN-LSTM version, normalization is even more important because neural networks are generally more stable and easier to optimize when the input features lie within comparable ranges.

2.6 Feature selection based on correlation structure

After normalization, the next step is featuring selection. The proposed study performs this step using a correlation heatmap in order to identify highly correlated predictors and remove redundant information. This is important because redundant variables may not improve predictive power but can increase complexity, noise, and instability. In this study, strong positive correlation exists between air temperature and process temperature, while a strong inverse correlation exists between torque and rotational speed. Therefore, one feature from each highly correlated pair is removed. In addition, the five failure subtype variables are removed because they directly reveal the cause of failure and would therefore leak target information into the predictor set. The UDI column is also removed because it does not contribute useful discriminatory information. After this process, the training data are reduced to five main predictive features.

The Pearson correlation coefficient between two variables $x$and y is

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \quad (11)$$

where $r_{xy} \in [-1,1]$. Values close to 1indicate strong positive correlation, values close to -1indicate strong negative correlation, and values near 0suggest weak linear association.

The conceptual value of this step is that it simplifies the learning problem. Instead of forcing the model to process a mixture of informative and redundant variables, the selected feature set retains only the most useful attributes for classification. This often improves interpretability and can also improve generalization performance.

2.7 Multicollinearity analysis using VIF

After correlation-based filtering, the source paper further checks the retained variables for multicollinearity using the Variance Inflation Factor (VIF). This is a useful diagnostic because even if a pairwise heatmap appears acceptable, a predictor may still be strongly explained by a combination of other predictors. The paper states that multicollinearity may require further investigation if VIF > 4 or if tolerance is below 0.25, and that serious multicollinearity is present when VIF > 10 or tolerance falls below 0.1. The study reports that the selected predictors do not show serious multicollinearity; therefore, principal component analysis is not applied.

he VIF for predictor $j$is defined as

$$VIF_j = \frac{1}{1-R_j^2} \quad (12)$$

where $R_j^2$is the coefficient of determination obtained by regressing the j-th predictor on all other predictors?

The corresponding tolerance is

$$\text{Tolerance}_j = 1 - R_j^2 = \frac{1}{VIF_j} \quad (13)$$

This step strengthens the methodological rigor of the paper. It confirms that the retained predictors are not merely fewer in number, but also statistically appropriate for downstream modelling. In other words, feature selection reduces redundancy, while VIF confirms that the reduced set is stable enough for predictive modelling.

2.8 Baseline machine learning stage

Once preprocessing, balancing, normalization, and feature engineering are completed, the study trains a group of supervised machine learning classifiers. The source paper explains that this stage is intended to compare the performance of different algorithms on the same predictive maintenance task and to identify the strongest conventional baseline before deep learning comparison. It further notes that the AUC score is used as the primary criterion for selecting the optimal model, followed by hyperparameter adjustment.

In general form, a classifier learns a mapping

$$\hat{y} = f(x;\theta) \quad (14)$$

where $f(\cdot)$is the model, θdenotes its learned parameters, and $\hat{y}$is the predicted class?

If the classifier outputs a probability score rather than a hard class decision, then

$$\hat{p}(y = 1 \mid x) = f(x;\theta) \quad (15)$$

and the final class label is obtained as

$$\hat{y} = \begin{cases} 1, & \hat{p} \geq \tau \\ 0, & \hat{p} < \tau \end{cases} \quad (16)$$

where $\tau$is a decision threshold?

This stage serves two purposes. First, it provides a practical benchmark against which the deep model can be evaluated. Second, it reveals which conventional classifier is naturally most compatible with the structure of the predictive maintenance data. According to the source paper, this process ultimately identifies XGBoost as the strongest traditional model.

2.9 Hyperparameter tuning

The paper gives separate emphasis to hyperparameter tuning and discusses Grid Search and Random Search as common tuning techniques. It also states that the tuned model is selected after comparing performance, and that Random Search CV is regarded as superior among commonly employed methods in the study context. Hyperparameter tuning is especially important because the performance of many machine learning algorithms depends strongly on their configuration rather than only on the algorithm family itself.

The optimization goal can be written as

$$\theta^* = \arg\max_{\theta \in \Theta} \mathcal{M}(f_\theta, \mathcal{D}_{val}) \tag{17}$$

where $\Theta$is the hyperparameter search space and $\mathcal{M}$is a validation metric such as AUC.

Conceptually, this means the model is not accepted in its default form. Instead, multiple candidate settings are explored to determine which combination yields the best classification performance on validation data. In predictive maintenance, this step is crucial because small improvements in recall, AUC, or F1-score can translate into meaningful improvements in early-failure detection.

2.10 Proposed CNN-LSTM

This study compares the best machine learning model with ANN and CNN- LSTM, and reports that CNN-LSTM performs better than ANN and the conventional models on the unbalanced predictive maintenance problem.

2.10.1 Convolutional feature extraction

The first stage of the hybrid model is a one-dimensional convolutional layer. Its function is to automatically extract local and informative patterns from the normalized input. Rather than manually designing interaction terms between temperature, torque, speed, and wear, the convolution layer learns these patterns directly from data.

For filter k, the convolution output at position $t$is

$$z_t^{(k)} = \sum_{j=0}^{m-1} w_j^{(k)} x_{t+j} + b^{(k)} \tag{18}$$

where $m$is the kernel size, $w_j^{(k)}$are the filter weights, and $b^{(k)}$is the bias.

After applying an activation function, the feature map becomes

$$a_t^{(k)} = \phi\left(z_t^{(k)}\right) \tag{19}$$

where $\phi(\cdot)$may be a nonlinear activation such as ReLU.

This stage allows the model to detect meaningful local structures, such as coupled variations among neighboring input components. In your predictive maintenance context, this can help the model learn compact patterns of abnormal operating behavior before they are passed to the recurrent stage.

2.10.2 Pooling and dimensionality reduction

After convolution, pooling is used to compress the learned feature maps and retain the most salient signals. This reduces computational cost and also suppresses noise.

A max-pooling operation can be written as

$$p_t^{(k)} = \max_{j \in \mathcal{W}_t} a_j^{(k)} \tag{20}$$

where $\mathcal{W}_t$is the pooling window?

Pooling acts as a local summarization operator. Instead of passing every small fluctuation to the next stage, the model forwards only the most informative responses. This usually improves robustness and reduces overfitting.

2.10.3 LSTM sequence modelling

The pooled feature representations are then fed into an LSTM layer. The LSTM component is designed to preserve useful information over longer dependencies and regulate information flow through gating operations. This is especially valuable when the learned patterns have conditional relationships that cannot be captured by a purely feedforward network.

The LSTM equations are

$$\begin{aligned}
f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned} \tag{21}$$

where f_t, i_t, and $o_t$are the forget, input, and output gates, $C_t$is the cell state, $h_t$is the hidden state, $\sigma(\cdot)$is the sigmoid function, and $\odot$denotes element-wise multiplication.

In practical terms, the LSTM stage helps the model decide what information should be retained, updated, or discarded as the learned representation moves through the sequence. This makes the hybrid model more expressive than ANN alone and potentially more adaptive than a plain CNN.

3.10.4 Dense classification output

The last stage of the network converts the deep representation into a binary failure probability:

$$\hat{y} = \sigma(W_d h_t + b_d) \quad (22)$$

where W_d and b_d are the dense-layer weights and bias, respectively.

Because the final task is binary classification, the output is naturally interpreted as the probability that a machine is in the failure class.

2.11 Loss function and optimization

Since the output is binary, a suitable training objective for the proposed CNN-LSTM is the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (23)$$

This loss penalizes incorrect classification with stronger emphasis when the model assigns high confidence to the wrong class. In industrial classification problems, such a loss is appropriate because the model must learn not only which class is correct, but also how confidently it should separate failure from non-failure.

The generic gradient-based parameter update is

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L} \quad (24)$$

where η is the learning rate?

This optimization framework aligns naturally with the paper's emphasis on comparing machine learning and deep learning under a common predictive maintenance setting, even though the exact CNN-LSTM equations are part of your modified methodology rather than the source paper itself.

2.12 Performance evaluation criteria

The paper explicitly reports that model performance is evaluated using accuracy, precision, recall, F1-score, and AUC-score, and that AUC is used as the main criterion for selecting the best conventional model. The paper also discusses ROC analysis for XGBoost and later compares ANN and LSTM performance. Your modified paper should retain the same evaluation scheme so that the proposed CNN-LSTM remains directly comparable with the other methods.

Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives. Then [18] , [19] , [20].

$$\begin{aligned}
&\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \\
&\text{Precision} = \frac{TP}{TP+FP} \\
&\text{Recall} = \frac{TP}{TP+FN} \\
&F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}
\end{aligned} \quad (24)$$

For ROC analysis, [21]

$$TPR = \frac{TP}{TP+FN} \quad (25)$$

$$FPR = \frac{FP}{FP+TN} \quad (26)$$

and the area under the ROC curve is

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \quad (27)$$

Each metric has a specific role. Accuracy gives an overall correctness rate, but it may be misleading under class imbalance. Precision indicates how reliable positive predictions are, recall measures the ability to detect actual failures, and F1-score balances these two concerns. AUC is especially useful because it evaluates class discrimination across thresholds rather than at a single decision point. For predictive maintenance, recall and AUC [22] are often especially important because missing a true failure can be more costly than issuing a false alert.

3. Results and Discussions

The main goal of developing the classification model is to determine if a piece of equipment will have failed by a certain date. While regression models can be employed to establish a predicted useful life for a machine; however, the predictions from these models will generally vary greatly as the machine degrades over time. In addition, when monitoring multiple machines simultaneously, keeping track of each individual machine becomes impractical. Consequently, an early warning system utilizing a classification model has been developed to produce timely and accurate warnings before a predetermined time frame. The performance metrics for all of the machine learning models that were examined are outlined in Figure 2.

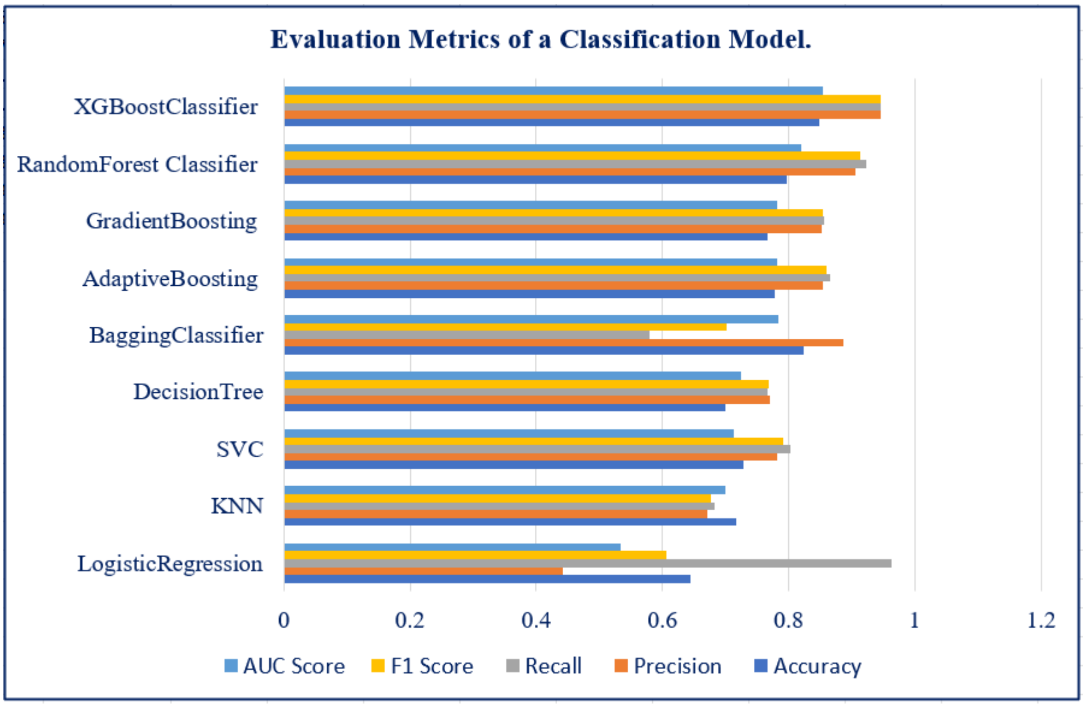

Figure 2. The evaluation metrics of a classification model.

Following the development of the initial results for the numerous machine learning models, hyperparameter tuning was applied to enhance the predictive capabilities of the models and increase their overall performance. The effectiveness of hyperparameter tuning with respect to the performance of the machine learning models is outlined in Figure 3.

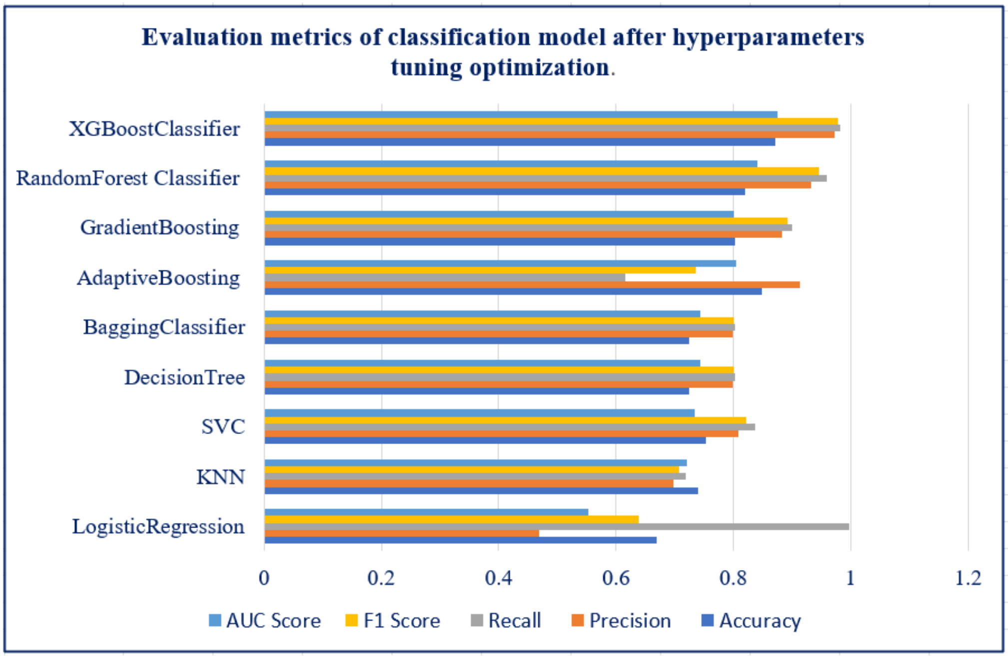

Figure 3. The evaluation metrics of classification model after hyperparameters tuning optimization

According to the results presented in Figure 3, the XGBoost classifier achieved the best overall performance among the evaluated machine learning models, primarily due to its highest Area Under the Curve (AUC) score and superior classification accuracy. The ROC curve of the XGBoost model demonstrates its ability to discriminate effectively between positive and negative classes across different threshold values, making it an important indicator of classification performance. The ROC curve corresponding to the XGBoost classifier is illustrated in Figure 4, while the confusion matrix obtained on the validation dataset is presented in Table 1.

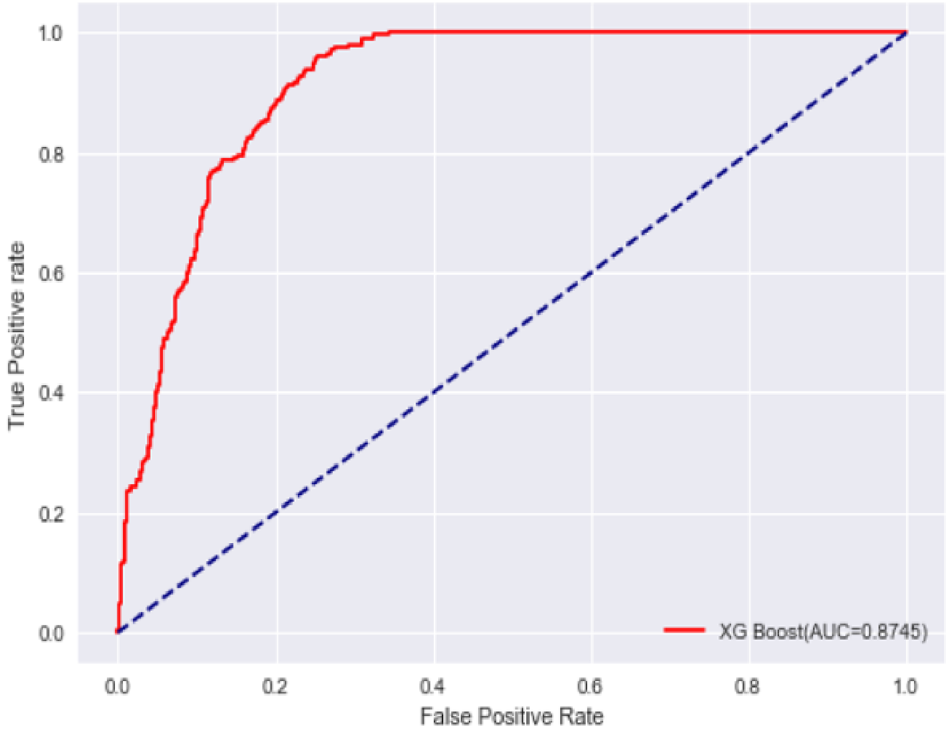

Figure 4. The ROC curve corresponding to the XGBoost classifier

Table: 1 The Confusion Matrix of the proposed study

CONFUSION MATRIX		Actual	
		Positive	Negative
Predicted	Positive	1339	41
	Negative	28	56

Following the evaluation of the conventional machine learning models, the proposed hybrid deep learning model based on Convolutional Neural Networks and Long Short-Term Memory networks (CNN–LSTM) was developed and trained for machine failure prediction. The performance of the CNN–LSTM model is presented in the corresponding results tables and compared with that of the traditional machine learning approaches (See Figure 5). By integrating CNN for automatic feature extraction and LSTM for temporal dependency learning, the hybrid model provides a more comprehensive representation of machine condition patterns and failure-related sequences.

This section evaluated the effectiveness of several machine learning methods in comparison with the proposed hybrid deep learning framework. Among the machine learning models, XGBoost produced the best performance on the imbalanced dataset. In contrast, the CNN–LSTM model demonstrated the strength of hybrid deep learning in predictive maintenance by capturing both local discriminative features and sequential information from the input data. These findings suggest that the CNN–LSTM architecture constitutes a more advanced and promising framework for machine fault diagnosis, with the potential to improve predictive performance beyond that of conventional standalone models.

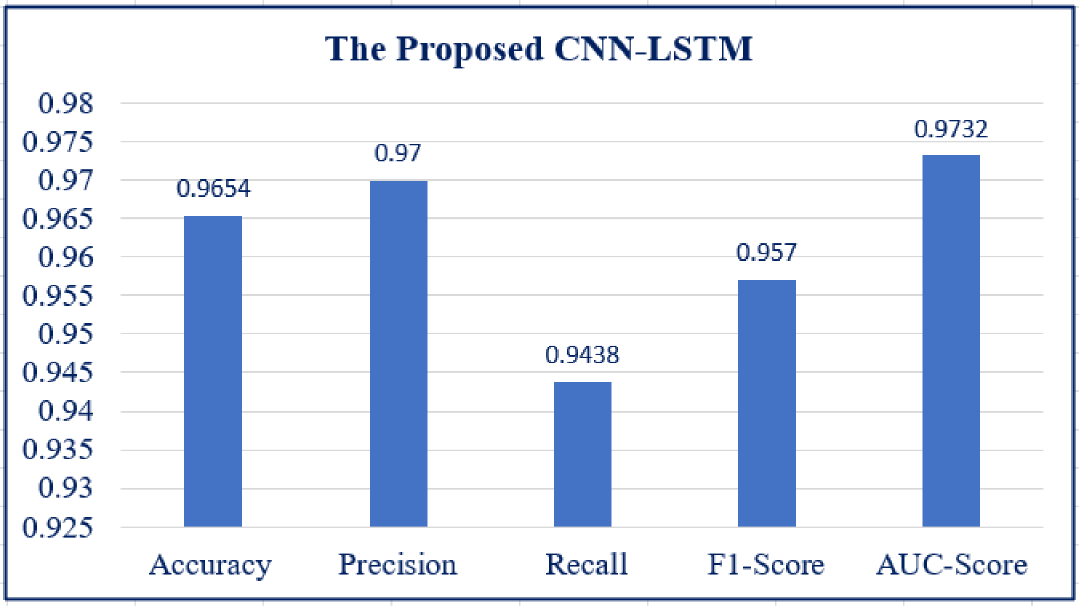

Figure 5: The performance measures of the proposed CNN- LSTM model.

4. Conclusions and Future Directions

Predictive maintenance has been identified as a key element of Industry 4.0 and supports early fault detection, optimal planning of maintenance activities and reduction of unscheduled downtime in industrial applications. Therefore, the development of robust classification models for predicting machine failures is currently one of the central objectives of research and practice. The results of this study show that data-driven predictive models are able to provide a suitable support for condition monitoring and early identification of possible equipment failures. However, there are numerous obstacles that limit the effective introduction of predictive maintenance approaches. Class imbalance in industrial datasets and the need for systematic hyperparameter optimization to achieve robust predictive performance are two major obstacles.

This paper compares well-established machine learning and deep learning models for predicting machine failures based on an imbalanced dataset. In order to establish a fair comparison environment, the SMOTE technique was used to counteract class imbalance and to make the comparison of the models more reliable. The results show that the model XGBoost performed best among the conventional machine learning models. In the area of deep learning, the use of a hybrid CNN-LSTM approach provides a higher level of predictiveness than the individual components, because it combines the ability of Convolutional Neural Networks (CNN) to extract features from data and the ability of Long Short-Term Memory (LSTM) networks to model sequential dependencies. Therefore, it provides a promising approach for more accurate fault diagnoses and predictive maintenance. This shows that the improvement of predictive accuracy is closely related to balancing of the data, feature representation and model optimization.

In addition, this study shows the important role of high-quality data for the development of reliable predictive maintenance systems. Reliable predictive maintenance is highly dependent on continuous data collection via IoT infrastructures and sensor-based monitoring platforms. Despite these opportunities, many industrial environments lack sufficient investments in data acquisition technologies. As a consequence, unanticipated failures and production stops continue to pose serious operational risks. Although predictive maintenance offers strong strategic and economic benefits, the large-scale application of predictive maintenance is still limited by the high costs of advanced sensing, monitoring and computing infrastructure.

Therefore, future research should follow several paths. Firstly, the focus should be placed on the development of low-cost and energy-efficient sensing technologies for the promotion of predictive maintenance in resource-poor industrial environments. Secondly, further investigations will be required to improve the architectural design and hyperparameter configuration of hybrid CNN-LSTM models for better performance regarding generalization, computational resources and scalability. Thirdly, future studies can integrate the CNN-LSTM framework with additional advanced methods, such as attention mechanisms, transfer learning and Explainable Artificial Intelligence (XAI), to enhance the predictive power and model transparency. Fourthly, further evaluations of the hybrid deep learning models based on larger, more heterogeneous and real-world industrial datasets will demonstrate stronger empirical evidence for the practical applicability and robustness of the proposed approaches across various predictive maintenance applications.

References

1. Ciaburro, G. (2022). Machine fault detection methods based on machine learning algorithms: A review. *Mathematical Biosciences and Engineering*, *19*(11), 11453-11490.
2. Souza, R. M., Nascimento, E. G., Miranda, U. A., Silva, W. J., & Lepikson, H. A. (2021). Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers & Industrial Engineering*, *153*, 107060.
3. Belagoune, S., Bali, N., Bakdi, A., Baadji, B., & Atif, K. (2021). Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. *Measurement*, *177*, 109330.
4. Saberironaghi, A., Ren, J., & El-Gindy, M. (2023). Defect detection methods for industrial products using deep learning techniques: A review. *Algorithms*, *16*(2), 95.
5. Rezazadeh, N., Perfetto, D., de Oliveira, M., De Luca, A., & Lamanna, G. (2026). A fine-tuning deep learning framework to palliate data distribution shift effects in rotary machine fault detection. *Structural Health Monitoring*, *25*(2), 661-683.
6. Li, Z., Jiang, H., & Wang, X. (2025). A novel reinforcement learning agent for rotating machinery fault diagnosis with data augmentation. *Reliability Engineering & System Safety*, *253*, 110570.
7. Li, X., Yu, S., Lei, Y., Li, N., & Yang, B. (2024). Dynamic vision-based machinery fault diagnosis with cross-modality feature alignment. *IEEE/CAA Journal of Automatica Sinica*, *11*(10), 2068-2081.
8. Yan, X., Jiang, D., Xiang, L., Xu, Y., & Wang, Y. (2024). CDTFAFN: A novel coarse-to-fine dual-scale time-frequency attention fusion network for machinery vibro-acoustic fault diagnosis. *Information Fusion*, *112*, 102554.
9. Romeser, M., & von Schwerin, R. (2026). Data-Driven Serial Testing of Axial Piston Units: Integrating Signal Processing and Machine Learning for Roller Bearing Fault Detection. *International Journal of Fluid Power*, 175-224.
10. Amiri, A. F., Oudira, H., Chouder, A., & Kichou, S. (2024). Faults detection and diagnosis of PV systems based on machine learning approach using random forest classifier. *Energy Conversion and Management*, *301*, 118076.
11. Bilal, H., Obaidat, M. S., Aslam, M. S., Zhang, J., Yin, B., & Mahmood, K. (2024). Online fault diagnosis of industrial robot using IoRT and hybrid deep learning techniques: An experimental approach. *IEEE Internet of Things Journal*, *11*(19), 31422-31437.
12. Hanoon, F. S., Çevik, M., & Taha, M. S. (2024, May). A fault classification for defective solar cells in electroluminescence imagery based on deep learning approach. In *AIP Conference Proceedings* (Vol. 3097, No. 1, p. 050006). AIP Publishing LLC.
13. Abdulghani, S. F., Shtayt, B. A., Taha, M. S., & Hashim, M. M. (2025, September). Effective Knowledge Graph Representation for Cybersecurity Using AI-Based X Data and Named Entity Relation Technique. In *International Conference on Cybersecurity and Artificial Intelligence Strategies* (pp. 71-89). Cham: Springer Nature Switzerland.
14. Jabbar, N. K., Naderan, M., & Taha, M. S. (2025). HybridIoMT: A Dual-Phase Machine Learning Framework for Robust Cybersecurity in Internet of Medical Things. *International Journal of Intelligent Engineering & Systems*, *18*(4).
15. Ayankoso, S., Dutta, A., He, Y., Gu, F., Ball, A., & Pal, S. K. (2026). Performance of vibration and current signals in the fault diagnosis of induction

motors using deep learning and machine learning techniques. *Structural Health Monitoring*, *25*(1), 196-212.

16. Mei, Y., Nie, L., Liao, K., Xu, Y., Lin, C., & Xiao, B. (2026). UniStitch: Unifying Semantic and Geometric Features for Image Stitching. *arXiv preprint arXiv:2603.10568*.

17. Ismael, B. M., Ngadi, A. B., Taha, M. S., & Sharif, J. B. M. (2026, February). Non-dominated sorting genetic algorithm for channel assignment in multiple radio interfaces with multiple channels. In *AIP Conference Proceedings* (Vol. 3393, No. 1, p. 060041). AIP Publishing LLC.

18. Schnawa, S. A., Rafie, M., & Taha, M. S. (2023, December). DAE-DBN: An effective lung cancer detection model based on hybrid deep learning approaches. In *International Conference of Reliable Information and Communication Technology* (pp. 108-118). Cham: Springer Nature Switzerland.

19. Yaseen, N. A., Jasim, M. H., Abdulghani, S. F., & Taha, M. S. (2026). ENHANCING ROAD SAFETY THROUGH ADVANCED MACHINE LEARNING: A COMPARATIVE STUDY OF RANDOM FOREST AND NEURAL NETWORKS FOR MULTI-CLASS ACCIDENT SEVERITY PREDICTION. *Power System Protection and Control*, *54*(1), 324-340.

20. Ahmed, A. S., Abood, I. N., & Taha, M. S. (2026). Adaptive Multi-objective Optimization for Obstacle-aware Wireless Sensor Network Deployment: A Comparative Analysis of State-of-the-Art Algorithms. *International Journal of Intelligent Engineering & Systems*, *19*(2).

21. Sideek, S. M. M., Ali, N. S., Abed, W. Y., & Taha, M. S. (2026). An Improved Anomaly-based Intrusion Detection System for IoT Applications using Machine Learning Methods. *Pertanika Journal of Science & Technology*, *34*(1).

22. Asraa, S. A., Haddad, A. A. A., Hameed, R. S., & Taha, M. S. (2025). An Accurate Model for Text Document Classification Using Machine Learning Techniques. *Ingenierie des Systemes d'Information*, *30*(4), 913.

23. Ismael, B. M., Ngadi, M. A., Sharif, J. B. M., & Taha, M. S. (2025). Multi-Agent Reinforcement Learning for User-Router Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks. *International Journal of Intelligent Engineering & Systems*, *18*(8).