SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Engineering Data Systems For Intelligent Fault Detection And Predictive Maintenance In Industrial Equipment Monitoring

Dr. Sarita Mohapatra[1], Dr. Sweta Kumari Barnwal[2], Vijay Kumar[3], Amol Bhilare[4], Gendal Lal[5], Suganya S[6], Arif Ali[7], Manjula R[8]

[1] Assistant Professor, Department of Computer Applications, Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India. Email: saritamahapatra@soa.ac.in ORCID: 0009-0004-6374-8005
[2] Assistant Professor, Department of Computer Science & IT, Arka Jain University, Jamshedpur, Jharkhand, India. Email: sweta.b@arkajainuniversity.ac.in ORCID: 0000-0003-2116-930X
[3] Assistant Professor, Department of School of Engineering & Technology, Noida International University, Greater Noida, Uttar Pradesh, India. Email: vijay.kumar@niu.edu.in
[4] Assistant Professor, Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India. Email: amol.bhilare@vit.edu
[5] Assistant Professor, Department of Computer Science & Engineering, Vivekananda Global University, Jaipur, Rajasthan, India. Email: gendal.lal@vgu.ac.in ORCID: 0009-0002-3753-5781
[6] Assistant Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: ssuganyamba@maher.ac.in
[7] Assistant Professor, Department of Computer Application, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.arif@dbuu.ac.in ORCID: 0009-0000-9025-7368
[8] Assistant Professor, Department of Commerce, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: manjular@maher.ac.in

Abstract
Advanced manufacturing systems produce Internet of Things (IoT)sensor data in huge volumes, which provides intelligent decision support for fault detection and Predictive Maintenance (PdM). However, current techniques mainly depend on labeled data, and it is difficult for them to handle heterogeneous, noisy, and non-stationary industrial data. Additionally, their lack of interpretability and poor generalization ability pose problems in practical applications. Thus, this research introduces an intelligent Decision Support System (DSS) using Elephant Clan Optimized-Convolutional Recurrent Neural Network (ECO-CRNN). The proposed approach is experimented with by applying an industrial sensor dataset, which involves vibration, temperature, and current signals under two classes, namely normal and faulty states of machines. Firstly, Isolation Forest (IF) is used as the data preprocessing method to eliminate outlier data. Feature extraction is executed using the Discrete Wavelet Transform (DWT), which extracts time-frequency features from non-stationary sensor data for efficient fault representation. The findings from experiments reveal that the technique has achieved MAE 0.050, MSE 0.010, and R^2 0.960 when compared with the conventional Deep Learning (DL) and Machine Learning (ML) algorithms, and the system uses an implementation model based on Python. In conclusion, this research proposes an effective combination of DL and data engineering techniques for providing intelligent decision-making capabilities for fault detection and PdM in Industrial environments.

Keywords: Engineering Data Systems, Intelligent Decision Support, Predictive Maintenance (PdM), Fault Detection.

1. Introduction

Most industrial processes are usually rigid and expensive, hence making it difficult to scale operations and carry out efficient maintenance practices. The incorporation of into Artificial Intelligence (AI) and the Internet of Things (IoT) sensor network, coupled with cloud computing, can result in improved decision-making and increase operational efficiency [1]. The Predictive Maintenance (PdM) utilizes the technologies of IoT and Machine Learning (ML) for monitoring of equipment in industries that results in early detection of problems. It ensures less downtime, more reliability, and changes the maintenance process from reactive to proactive. Although it offers all these benefits, handling big data from sensors nevertheless poses many challenges [2]. PdM increases efficiency, reduces costs, and improves the reliability of the system by utilizing the power of sensors, ML, and Digital Twin (DT).Nevertheless, scalability, interpretability, and deployment remain challenging issues, emphasizing the necessity of industrial-level solutions [3]. DT is a fundamental component of Industry 4.0, facilitating PdM, increasing efficiency in operations, reducing downtime, improving quality, and ensuring real-time monitoring and decision-making, all of which help in optimizing industrial processes [4]. PdM along with the IoT sensors, Supervisory Control and Data Acquisition (SCADA), and Enterprise Resource Planning (ERP) systems, can help in monitoring machines, minimizing downtime, improving efficiencies, and dealing with reliability limitations and decision-making [5].With regard to sheet metal stamping operations in Industry 4.0, digital technologies and ML techniques are used for predicting the shape, detecting any problems that could arise at an early stage, and increasing productivity [6].

Research Aim: The purpose of this research is to design an intelligent decision-making tool for fault detection and PdM based on industrial sensor information (vibration, temperature, and current). In this research, the ECO-CRNNmodel is used to improve the accuracy of fault detection in noisy and imbalanced environments.

Research Organization: Part 1 explains the problem and motivation for applying the concept of PdM to industries. Part 2 discusses the literature review, which considers existing studies related to the fault prediction and PdM. Part 3 explains the methodology of the research. Evaluation results are provided in part 4. The conclusion and future scope is offered in part 5.

2. Related Work

PdM in manufacturing industries for better fault prediction in the research [7] uses Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) models. The method results in high accuracy, but their diverse dataset constraints mean the results might not generalize well. A Hybrid Deep Learning(DL)model was developed to predict faulty machines in the industrial sewage treatment plants, enhancing key parameter forecasts by using Multi-Head Attention, Autoencoder–Self-Organizing Map (AE-SOM)[8] with Convolutional Neural Network–Bidirectional Long Short-Term Memory (CN–BiLSTM), and Variational Autoencoders (VAEs), resulting in a high coefficient of determination (R^2) value, but scaling up to real-world use is tough.In the PdM system [9]for knitting machines to catch faults early Adaptive Boosting (AdaBoost) is used on IoT data about speed and steps, and the model hits 92% accuracy. The DT model [10]was developed for a squirrel cage induction motor to predict maintenance needs, find the faults early, and estimate the motor life. It achieves up to 90.9% accuracy with Support Vector Machines (SVM), but the method can't yet be easily applied to other systems. DT-based PdM system for Remote Terminal Units (RTUs)[11] in power generation to spot operation issues early by using ML and DL with sensor data. Long Short-Term Memory Autoencoder (LSTM-AE)performs best here, but its broad use in different settings is nevertheless restricted. The research developed a Convolutional Neural Network–Recurrent Neural Network (CNN-RNN) [12] model for predicting maintenance needs in Direct Current (DC) motor drives of industrial robots and early fault detection using temperature and rotational speed data. The model outperforms a Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) with better accuracy, but it encounters integration issues when applied in different industrial settings. The PdM method introduced for the Photovoltaic systems (PV) to detect the anomalies and faults by using LSTM-AE[13] for analyzing irradiance and Alternating Current(AC) output data. The results show low reconstruction errors, Mean Absolute Error (MAE) 2.76, Root Mean Squared Error (RMSE) 3.30, and Mean Squared Error (MSE) 10.95. But still, it faces challenges due to its inability to cope with unusual cases as it depends on previous data patterns. The PdM technique [14], driven by AI and designed for gas turbines, was developed to improve thermal condition monitoring using XGBoost on thermocouple-related features and reaches high accuracy levels. Validation of this technique was impeded due to narrowness of the used dataset, while further application requires more investigation. Research [15] developed a PdM technique based on privacy protection and interpretation for Industry 4.0 with the aim of minimizing machine downtime and increasing transparency. Using XAI in conjunction with FL, an XFL model was designed. The accuracy level of the model is high, but it can require further investigation for extremely diverse industrial environments. Decision Support System (DSS) [16] was developed for PdM in industries, trying to beat labeled-data obstacles.

By using feature extraction and ML on condition-monitoring data, the research ensure an MAE of 0.089, MSE of 0.018, and R^2 of 0.868, but it needs to be tested more extensively for optimal functioning regardless of changing environmental conditions.

3. Methodology

The suggested method shown in Figure 1 involves using the Kaggle PdM dataset with data about vibrations, temperatures, and currents of industrial machines. The Isolation Forest (IF) algorithm is used for data preprocessing, and to extract the time-frequency characteristics of the non-stationary sensor signals of the industrial machines using Discrete Wavelet Transform (DWT). The next step involves developing the Convolutional Recurrent Neural Network (CRNN) model, where layers of convolution are used for feature extraction, while layers of recurrent are used for capturing temporal dependencies for better fault detection, and the final step includes applying ElephantClan Optimization (ECO) to tune CRNN hyperparameters.

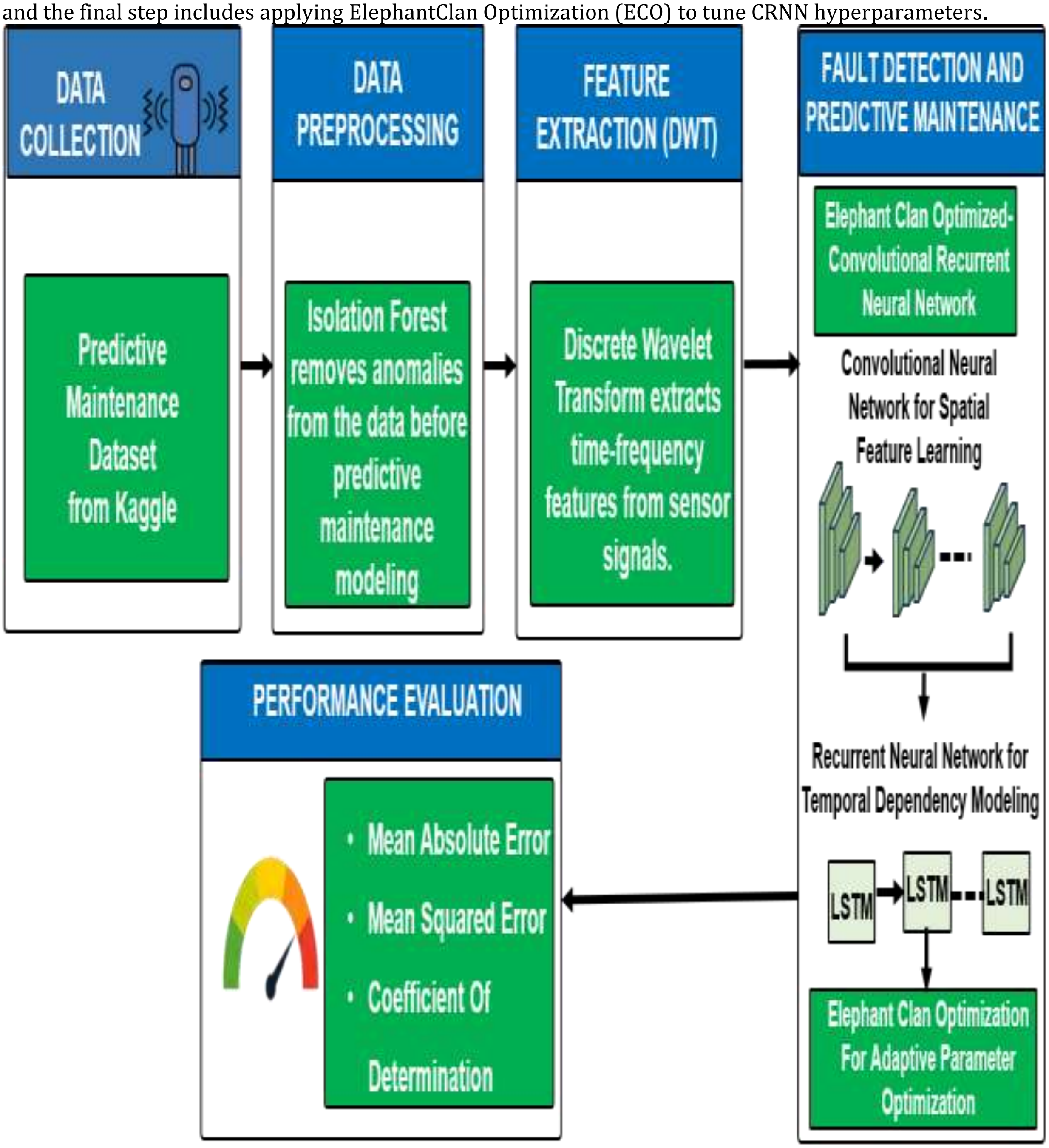


Figure 1: Methodology flow of Proposed Method

3.1 Data Acquisition

The PdM Dataset collected from the Kaggle source (https://www.kaggle.com/datasets/hiimanshuagarwal/predictive-maintenance-dataset) contains 124,494 samples that record vibrations, temperature, and current from industrial machinery running under normal and abnormal conditions. Data consists of several parameters which include readings from sensors, equipment parameters, and time-stamps. It enables individuals to build models for prediction and detection of failures in machinery, using the usage level as the basis for failure estimates. The use of past data collected by the sensors makes the model more accurate in its predictions. To test the validity of the model, data is generally partitioned 80/20.

3.2 Data Preprocessing using Isolation Forest (IF)

The IF algorithm works by constructing an isolation tree, which recursively partitions sensor observations to isolate anomalies and faults effectively from normal behavior. The shorter average distances from the root nodes of isolation trees are the basis for identifying anomalies, which could be early warning signs of faulty equipment. Using various parameters like the number of trees and maximum features, IF successfully distinguishes abnormal vibration, temperature, and current.

$$d(m) = 2G(m-1) - \left(\frac{2(m-1)}{m}\right) \quad (1)$$

In Equation (1), d denotes the length of the anomaly path, mand $m-1$ refers to the present and previous size count of the sample, $d(m)$ represents the length of the average fault detection path in industries, G refers to the function of harmonic distribution, and $G(m-1)$ denotes the number function of harmonic measure with the size count of the sample.

$$t(W, m) = 2^{-\frac{F(g(W))}{d(m)}} \quad (2)$$

In Equation (2), trefers to the value of faults in industrial machines, W denotes the sample of input data, m indicates thesize count of the sample, $t(W, m)$represents the score value of anomaly detected in machines, F denotes function of the path length, g refers to the node of the isolation tree, $g(W)$ indicates the structure of tree partition in fault detection,$F(g(W))$ represents the function of path length measure in industrial fault detection, d denotes the length of the anomaly path, and $d(m)$ represents the length of average fault detection path in industries.

3.3 Feature Extraction using Discrete wavelet transforms (DWT)

DWT is used to analyze time-frequency characteristics of signals from sensors in industry for detecting faults and conducting PdM. Contrary to Fourier transform methods, DWT can efficiently work with non-stationary signals by detecting transients and trends, as well as variations in frequency. The mother wavelet acts as a prototype that is then scaled and shifted to approximate signals in terms of various scales. Using DWT to analyze the time scale behavior of signals such as vibration, temperature, and current allows one to detect anomalies in operation, indicating potential faults of equipment.

$$DWT(n, m) = \frac{1}{\sqrt{2^n}} \sum_l e(l) \varphi\left[\frac{m - l2^n}{2^n}\right] \quad (3)$$

In Equation (3), DWT denotes the discrete wavelet transform, n refers to the scale level of decomposition, m indicates the time position of the signal, $DWT(n, m)$ refers to the coefficient value of the wavelet,2^n refers to the scaling factor of wavelet,$\frac{1}{\sqrt{2^n}}$represents the normalized factor of scaling, $\sum$ indicates the summation function of the coefficients, l denotes the index value of the samples,$\sum_l$ depicts the sum of all samples, e refers to the signal of the sensors, $e(l)$ represents the signal of fault diagnostics, φ is the pattern extractor of faults, and $m - l2^n$ indicates the time shift of scaling.

3.4 Fault Detection and Predictive Maintenance (PdM) using Elephant Clan Optimized-Convolutional Recurrent Neural Network (ECO-CRNN)

The proposed PdM model employing CRNN uses the convolution layer to extract the spatial feature from the signal data, whereas recurrent layer considers temporal relations for the fault progression analysis. Normalization of input signals is done via max-absolute scaling for stability. Feature extraction can also be improved using the convolution operation, together with nonlinear activation, for better representation of the fault. To optimize the performance of the model, the hyperparameters of learning rate, filter size, and recurrent units are optimized using ECO.

Fault Detection and Predictive Maintenance (PdM) using Convolutional Recurrent Neural Network (CRNN): CRNN helps detect defects and perform predictive maintenance in industrial sensor systems. It makes use of convolution to identify the presence of patterns in data such as vibration, temperature, and current readings, while recurrence is used to enable learning through patterns. The convolution layers detect instantaneous abnormalities, while the recurrent layers detect deterioration over time. CRNN enhances efficiency through consideration of spatial and temporal data, which leads to efficient machine performance.

$$\mathrm{W}' = \frac{\mathrm{W}}{\mathrm{Max}[\mathrm{Abs}(\mathrm{W})]} \tag{4}$$

In Equation (4), W refers to the sensor signal from the input, W' denotes the vector of normalized output, Max represents the function of maximum value, Abs indicates the operator of absolute value, Abs(W)indicates the magnitude vector of the fault signals, andMax[Abs(W)] depicts the magnitude of the peak fault signal.

$$\mathrm{d}_s^j = \mathrm{ReLU}(\mathrm{w}' * \mathrm{L}^j + \mathrm{a}^j)\mathrm{Max}(0, \textstyle\sum_k \mathrm{W}'_{s+k}\, \mathrm{L}_k^j + \mathrm{a}^j) \tag{5}$$

In Equation (5), d refers to the output of the extracted feature, j denotes the index of the layer, s indicates the index of the time step, d_s^jdepicts the output of the fault feature, ReLU represents the rectified linear unit, wrefers to the weight matrix of convolution, w'denotes the normalized matrix of weight, $*$represents the operations of convolution, L refers to the feature map of input, L^jindicates the feature map of vibration,a refers to the adjustment term of bias, a^j indicates the value of bias term, Max represents the is the function of maximum value,$\sum$ indicates the summation function of the coefficients,kdenotes the index position of the feature, $\sum_k$ represents the aggregation sum of the features, W'_{s+k}refers to the value of shifted weight, and L_k^j indicates the feature of local input.

Hyperparameter tuning with Elephant Clan Optimization (ECO): ECO is used in the optimization of the CRNN model for fault detection and PdM in industrial machines. Guided by the social organization of elephants, ECO automatically tunes the most important hyperparameters, such as the learning rate, the size of convolutional filters, and the configuration of the recurrent units, to achieve a more accurate prediction. In the context of fault detection and PdM, each elephant clan is a possible solution, and the matriarch serves as a guide, leading the search through her experience to find the best solutions. Elephants are able to explore independently, keeping diversity and preventing early convergence in males. ECO enhances the capability of the CRNN to detect faults and make reliable predictions in noisy datasets with multiple sensors to balance exploration and exploitation.

$$\mathrm{w}_{\mathrm{FC_j},\mathrm{F_n},\mathrm{i}}^{(\mathrm{it}+1)} = \mathrm{w}_{\mathrm{FC_j},\mathrm{F_n},\mathrm{i}}^{(\mathrm{it})} + \mathrm{q} \times \alpha \times [\mathrm{w}_{\mathrm{FC_j},\mathrm{N},\mathrm{i}}^{(\mathrm{it})} - \mathrm{w}_{\mathrm{FC_i},\mathrm{F_n}\mathrm{i}}^{(\mathrm{it})}] \tag{6}$$

In Equation (6), w denotes the weight of fault prediction, i indicates the index of the neighbor solution, t refers to the counter value of iteration,(it + 1)indicates the next iteration,F represents the set of feature vectors, C denotes the cluster solution of the candidate, n refers to the index of the node, $\mathrm{F_n}$refers to the node index of the feature, $\mathrm{w}_{\mathrm{FC_j},\mathrm{F_n},\mathrm{i}}^{(\mathrm{it}+1)}$and $\mathrm{w}_{\mathrm{FC_j},\mathrm{F_n},\mathrm{i}}^{(\mathrm{it})}$represents the weight of the updating and current solution, q denotes the factor of random scaling, α indicates the counter factor of exploration,N denotes the new candidate and$\mathrm{w}_{\mathrm{FC_j},\mathrm{N},\mathrm{i}}^{(\mathrm{it})}$refers to the weight of the current neighbor solution.

$$\mathrm{w}_{\mathrm{FC_j},\mathrm{F_n}\mathrm{i}}^{(\mathrm{it}+1)} = \mathrm{w}_{\mathrm{FC_j},\mathrm{F_n}\mathrm{i}}^{(\mathrm{it})} + \mathrm{q} \times \beta \times [\mathrm{w}_{\mathrm{best},\mathrm{i}} - \mathrm{w}_{\mathrm{FC_j},\mathrm{F_i}}^{(\mathrm{it})}] \tag{7}$$

In Equation (7),w denotes the weight of fault prediction, i indicates the index of the neighbor solution, t refers to the counter value of iteration,$(it + 1)$ indicates the next iteration, F represents the set of feature vectors, C denotes the cluster solution of the candidate, n refers to the index of the node, F_n refers to the node index of the feature, $w_{FC_j,F_n,i}^{(it+1)}$and $w_{FC_j,F_n,i}^{(it)}$represents the weight of the updating and current solution, q denotes the factor of random scaling,β refers to the control parameter of refinement, $best$ denotes the best solution, $w_{best,i}$represents the weight of global best solution, and$w_{FC_j,F_i}^{(it)}$ indicates the weight value of the solution.

$$w_{NC,F_n,i}^{(it+1)} = w_{NC,F_n,i}^{(it)} + q \times o \times [w_i^{Max} - w_i^{Min}] \tag{8}$$

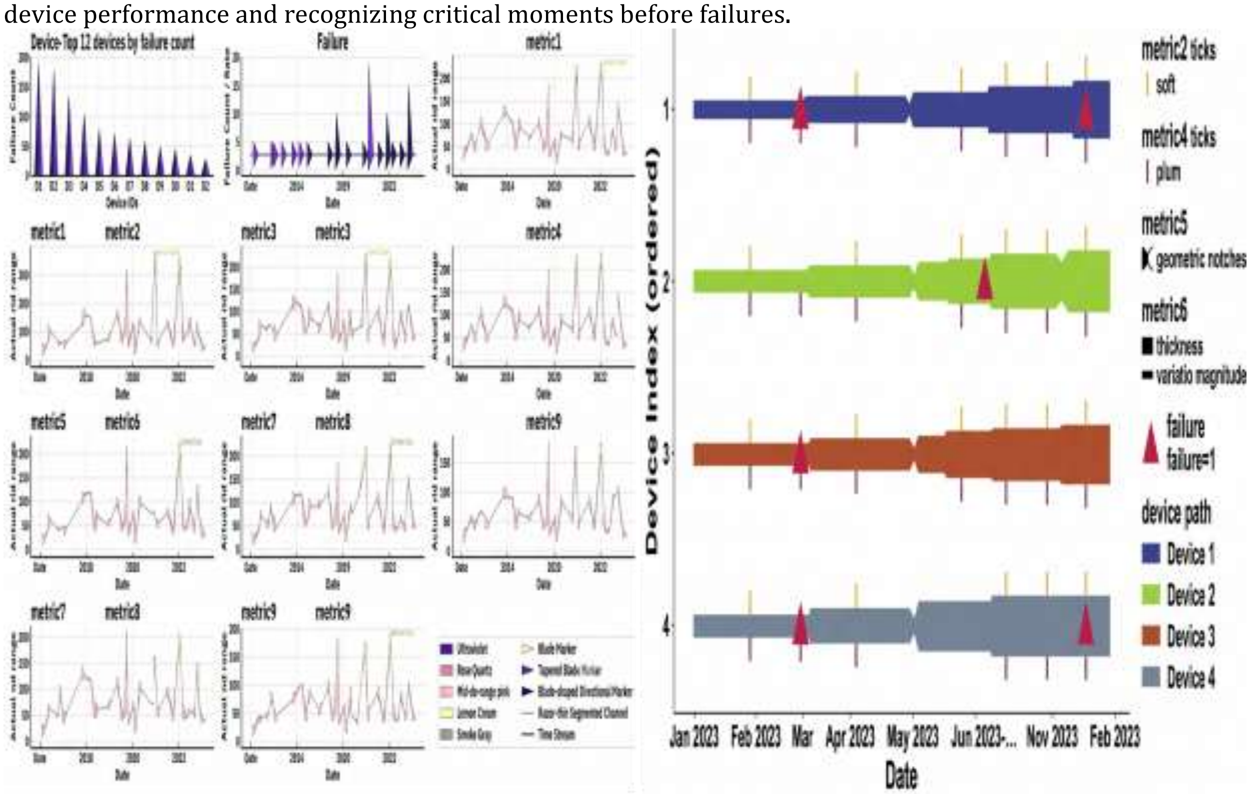
In Equation (8),w denotes the weight of fault prediction, i indicates the index of the neighbor solution, t refers to the counter value of iteration,$(it + 1)$ indicates the next iteration, F represents the set of feature vectors, C denotes the cluster solution of the candidate, n refers to the index of the node, F_n refers to the node index of the feature,N denotes the new candidate,$w_{NC,F_n,i}^{(it+1)}$ and $w_{NC,F_n,i}^{(it)}$denotes the updated and current candidate weight solution, q denotes the factor of random scaling,o denotes the factor of roaming,Max and Min represents the upper and lower weight, andw_i^{Max} and w_i^{Min} indicatesmaximum and minimumweight value.

Hyperparameters of the Proposed ECO-CRNN Model: The CRNN model for fault detection and PdM uses an input shape of 128 x 128 x 1. It starts with three convolutional layers, featuring 32, 64, and 128 filters, each with a 3 x 3 kernel size. They make use of the convolution 2x2 pooling and Rectified Linear Unit (ReLU)activation function. They have also used the LSTM layer of size 128 to process the temporal data. To improve generalization, they have applied a dense layer of 256 and a dropout of 0.30, which is followed by Softmax output layer. In their model architecture, they used the Adam optimizer with a learning rate of 0.001 during training. The batch size was 64, and the number of epochs was 100. Hyperparameters were tuned using ECO with a population of 30, a hundred iterations, 0.80 crossover rates, and 0.10 mutation rates.

4. Result and discussion

The model of the PdM and fault detection was designed using a high-performance workstation that can enable the training of the CRNN model and the execution of the ECO algorithm effectively. It features NVIDIA RTX 3080 graphics card that contains 10 GB of Graphics Double Data Rate 6 (GDDR6) memory,32 GB Double Data Rate 4 (DDR4) Random Access Memory(RAM), Intel Core i7-12700K Central Processing Unit (CPU), and 1 TB Solid State Drive (SSD) that offers fast data processing and storage. It is designed using a 64-bit system. After hardware setup, the system can effectively process sensor data, train models, and conduct evaluations, providing insights for effective fault detection and predictive maintenance.

Explorative Data Analytics: Figure 2 represents the patterns of device failures and trends of operational metrics, showing reliability behavior and the pattern of failure progressions. Device failure distribution and metrics' time-varying trends are shown in Figure 2(a). This helps identify frequently failed devices and the spikes that are evident in their failure trends and in the metrics. Figure 2(b) provides an image of the ordered process flow of devices over time, with regard to failure events, metrics, and geometry. It assists in monitoring device performance and recognizing critical moments before failures.



(a) (b)

Figure 2: Visualization of (a) Analysis of Failure Trend, and (b) Timeline of Device Failure

Figure 3 represents the device failure corridors, event cascades, and metrics that show the reliability patterns during operation. Figure3(a) shows the failures per device, failure intensity pulses, and nine operational parameters. It shows differences between device reliability, temporal failure characteristics, and operational parameters that affect system performance. Figure 3(b) displays the top-ranking devices based on their failure intensity and the cascading nature of events over time. Each operational parameter panel identifies the repeating patterns, facilitating failure detection.

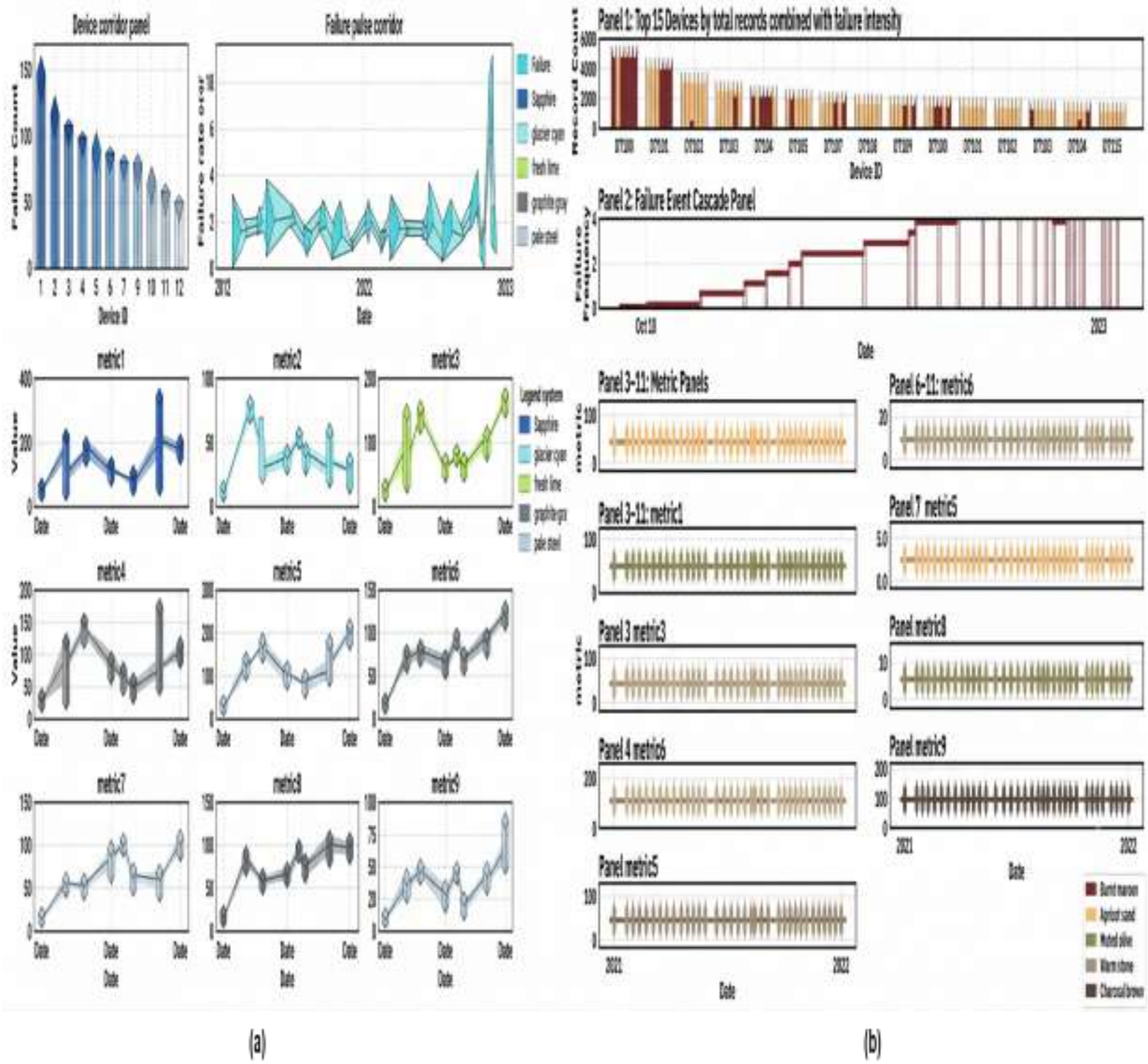


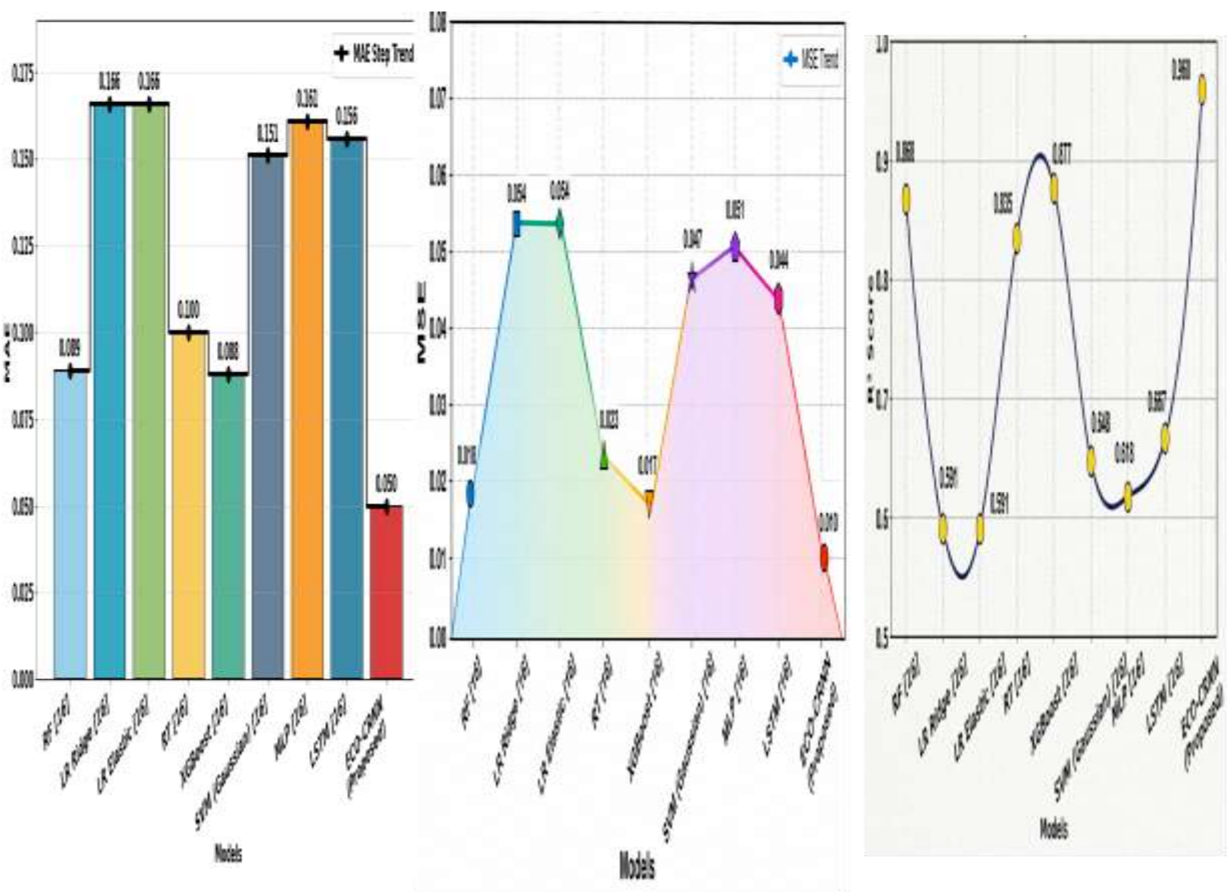
Figure 3: Visualization of (a)Insights of Operational Failure, and (b)Monitoring of Failure Cascade

Performance Evaluation: This research uses MAE, MSE, and R^2 to assess the CRNN model for fault detection and PdM. MAE is used to quantify the average comparison among the estimated and actual sensor readings. MSE goes one step ahead by penalizing large errors and helps determine the stability of the model as well as its susceptibility to unusual machine performance. The $\mathbf{R^2}$ score indicates the extent to which the proposed model captures the variance within the dataset. It signifies the effectiveness of the model predictions and the learning capabilities of CRNN.

Performance Evaluation: Table 1 and Figure 4 demonstrate the performance of various ML models against one another during their predictive capabilities. Classical approaches, including Ridges of Linear Regression (LR), LR of Elastic, and SVM, exhibit high error values alongside low R^2 scores. In contrast to them, ensemble methods for trees, namely Random Forest (RF), Regression Tree (RT), and XGBoost, prove far superior in terms of reduced MAE and MSE scores along with high R^2 scores. Among the DL models, Multilayer Perceptron (MLP) and LSTM prove relatively good but fall short of competing with the aforementioned ensemble methods. The proposed ECO-CRNN proves superior to all other baseline models, with low MAE score (0.050), low MSE score (0.010), and high R^2 value (0.960).

Table 1: Performance Evaluation of Baseline Methods with Proposed Model

Model	MAE	MSE	R^2
RF [16]	0.089	0.018	0.868
LR Ridge [16]	0.166	0.054	0.591
LR Elastic [16]	0.166	0.054	0.591
RT [16]	0.100	0.023	0.835
XGBoost [16]	0.088	0.017	0.877
SVM (Gaussian) [16]	0.151	0.047	0.648
MLP [16]	0.161	0.051	0.618
LSTM [16]	0.156	0.044	0.667
ECO-CRNN [proposed]	0.050	0.010	0.960



(a) (b) (c)

Figure 4: Visualization of (a)Performance Comparison of MAE, (b) Trend Analysis of MSE, and (c) Evaluation Comparison of R^2

All baseline models, like RF, Ridge, Elastic Net, RT, XGBoost, SVM, MLP, and LSTM, were retrained by using PdM dataset using an industrial context shown in Table 2. The IF algorithm was used for data preprocessing, whereas DWT was used to extract features from vibration, temperature, and current signals. In comparison with other algorithms, the proposed ECO-CRNN proved its superiority by exploiting spatial-temporal dependency information. Performance parameters for this model are given by MSE 0.010, R2 0.960, and MAE 0.050.

Table 2: Retrained evaluation of the Baseline Model with the Proposed Architecture

Model	MAE	MSE	R^2
RF	0.150	0.040	0.750
LR Ridge	0.220	0.070	0.550
LR Elastic	0.225	0.072	0.540
RT	0.180	0.045	0.720
XGBoost	0.140	0.038	0.780
SVM (Gaussian)	0.200	0.060	0.600
MLP	0.210	0.065	0.590
LSTM	0.195	0.055	0.620
ECO-CRNN [proposed]	0.050	0.010	0.960

Discussion: There exist several merits and demerits of employing predictive models in industrial maintenance. RF model [16] is not easy to interpret, and its tendency to overfit is high; besides, it cannot handle temporal data. LR Ridge [16] does not capture complicated relationships within data. LR Elastic Net [16] cannot process non-linear or temporal data. RT [16] suffers from over-fitting and tends to be overwhelmed by noisy data. XGBoost [16] is computationally intensive and requires much care in the choice of hyper-parameters. Gaussian SVMs [16] have problems handling large amounts of data. MLP [16] requires a huge amount of data to prevent over-fitting. LSTMs [16] are very slow and computationally expensive, and their operations are affected by noise. To address the limitations of previously mentioned predictive models, the proposed ECO-CRNN outperforms by employing convolutional layers to learn spatial features, recurrent layers to deal with temporal features and ECO to optimize hyper-parameters of the network, thereby enabling highly accurate predictions and robust immediate performance for the maintenance of industry machines.

5. Conclusion

The ECO-CRNN model utilizes IF in the form of anomaly removal and DWT for time-frequency feature extraction, then the model becomes capable of identifying the features based on temporal and spatial data obtained from the vibration, temperature, and current sensors. In addition, ECO could improve the effectiveness of the CRNN algorithm by tweaking certain parameters to identify faults and give predictive maintenance. The new approach is compared to ML-based approaches and DL-based methods in terms of experimental performance metrics showing higher R^2 of 0.960, MAE of 0.050, and MSE of 0.010 values. Despite this, the ECO-CRNN model needs high computational capability and could face some challenges when adapting to a variety of industrial environments, in which the distribution of the sensors would be unknown. Future scope is to develop an efficient, interpretable, and adaptable version of ECO-CRNN in real world applications.

Reference

1. Dera, P., Talaśka, T. and Długosz, R., 2026. A new, cost-efficient modular sensor platform for IoT and predictive maintenance in industrial applications. Journal of Computational and Applied Mathematics, 473, p.116777.https://doi.org/10.1016/j.cam.2025.116777
2. Aminzadeh, A., SattarpanahKarganroudi, S., Majidi, S., Dabompre, C., Azaiez, K., Mitride, C. and Sénéchal, E., 2025. A machine learning implementation to predictive maintenance and monitoring of industrial compressors. Sensors, 25(4), p.1006. https://doi.org/10.3390/s25041006
3. Mavrelis, I., Chatzopoulos, A. and Zacharia, P., 2026. Hybrid Deep Learning for Predictive Maintenance in Industrial Machinery Using LSTM and MLP Models. Machines, 14(2), p.191.https://doi.org/10.3390/machines14020191
4. Fantozzi, I.C., Santolamazza, A., Loy, G. and Schiraldi, M.M., 2025. Digital twins: Strategic guide to utilize digital twins to improve operational efficiency in Industry 4.0. Future Internet, 17(1), p.41.https://doi.org/10.3390/fi17010041
5. Rakholia, R., Suárez-Cetrulo, A.L., Singh, M. and Carbajo, R.S., 2025. Integrating AI and IoT for predictive maintenance in Industry 4.0 manufacturing environments: A practical approach. Information, 16(9), p.737.https://doi.org/10.3390/info16090737
6. Stefanovska, E. and Pepelnjak, T., 2025. Optimising predictive accuracy in sheet metal stamping with advanced machine learning: A LightGBM and neural network ensemble approach. Advanced Engineering Informatics, 65, p.103103.https://doi.org/10.1016/j.aei.2024.103103
7. Li, W. and Li, T., 2025. Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data. Scientific Reports, 15(1), p.23545.https://doi.org/10.1038/s41598-025-08515-z
8. Ramya, S. and Srinath, S., 2025. Advanced temporal deep learning framework for enhanced predictive modeling in industrial treatment systems. Results in Engineering, 25, p.104158.https://doi.org/10.1016/j.rineng.2025.104158
9. Elkateb, S., Métwalli, A., Shendy, A. and Abu-Elanien, A.E., 2024. Machine learning and IoT–Based predictive maintenance approach for industrial applications. Alexandria Engineering Journal, 88, pp.298-309.https://doi.org/10.1016/j.aej.2023.12.065
10. Singh, R.R., Bhatti, G., Kalel, D., Vairavasundaram, I. and Alsaif, F., 2023. Building a digital twin powered intelligent predictive maintenance system for industrial AC machines. Machines, 11(8), p.796.https://doi.org/10.3390/machines11080796
11. Lekidis, A., Georgakis, A., Dalamagkas, C. and Papageorgiou, E.I., 2024. Predictive maintenance framework for fault detection in remote terminal units. Forecasting, 6(2), pp.239-265.https://doi.org/10.3390/forecast6020014
12. Eang, C. and Lee, S., 2024. Predictive maintenance and fault detection for motor drive control systems in industrial robots using CNN-RNN-based observers. Sensors, 25(1), p.25.https://doi.org/10.3390/s25010025
13. Syamsuddin, A., Adhi, A.C., Kusumawardhani, A., Prahasto, T. and Widodo, A., 2024. Predictive maintenance based on anomaly detection in photovoltaic system using SCADA data and machine learning. Results in Engineering, 24, p.103589.https://doi.org/10.1016/j.rineng.2024.103589
14. Bunyan, S.T., Khan, Z.H., Al-Haddad, L.A., Dhahad, H.A., Al-Karkhi, M.I., Ogaili, A.A.F. and Al-Sharify, Z.T., 2025. Intelligent thermal condition monitoring for predictive maintenance of gas turbines using machine learning. Machines, 13(5), p.401.https://doi.org/10.3390/machines13050401
15. Alshkeili, H.M.H.A., Almheiri, S.J. and Khan, M.A., 2025. Privacy-preserving interpretability: an explainable federated learning model for predictive maintenance in sustainable manufacturing and industry 4.0. AI, 6(6), p.117.https://doi.org/10.3390/ai6060117

16. Rosati, R., Romeo, L., Cecchini, G., Tonetto, F., Viti, P., Mancini, A. and Frontoni, E., 2023. From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0. Journal of Intelligent Manufacturing, 34(1), pp.107-121.https://doi.org/10.1007/s10845-022-01960-x