Improving Predictive Maintenance in Industrial Environments via IIoT and Machine Learning

Saleh Othman Alhuqayl, Abdulaziz Turki Alenazi, Hamad Abdulaziz Alabduljabbar, Mohd Anul Haq*
Department of Computer Science, College of Computer and Information Sciences,
Majmaah University, Al-Majmaah, Saudi Arabia

***Abstract*—Optimizing maintenance procedures is essential in today's industrial settings to reduce downtime and increase operational effectiveness. To improve predictive maintenance in industrial settings, this article investigates the combination of machine learning (ML) techniques and the Industrial Internet of Things (IIoT). The goal of this research is to advance predictive maintenance in industrial settings by integrating ML with IIoT in a seamless manner. Addressing the complexities of industrial systems and limitations of traditional maintenance methods, this study presents a methodology leveraging four distinct ML models. The technique includes a thorough assessment of these models' correctness, revealing differences that highlight the significance of a careful model selection procedure. The current investigation analysis finds the most effective model for predictive maintenance activities using thorough data analysis and visualization. Our work offers a potential path forward for the industrial sector and provides insights into the complex interactions between IIoT and ML. This study lays the groundwork for future developments in predictive maintenance, which will reduce downtime and extend the life of industrial equipment.**

Keywords—Predictive maintenance; IIoT; data visualization; machine learning; industrial systems

I. Introduction

The industrial sector is always searching for methods to save expenses, increase productivity, and decrease downtime. Maintenance of machinery particularly that used in the textile sector is one area that might be improved. Reactive techniques have formed the foundation of traditional maintenance practices, where equipment is fixed following a failure [1]. Predictive maintenance, on the other hand, has become a more proactive and economical option by utilizing data analytics and ML approaches [2].

Predictive maintenance is a proactive approach that foresees equipment faults before they happen by using ML and data analytics. Organizations may schedule maintenance just in time to avert failures by identifying trends and abnormalities through continuous equipment condition monitoring and data analysis. This strategy increases the longevity of industrial assets, reduces downtime, and optimizes maintenance tasks. Predictive maintenance is unique in that it may replace reactive and fixed-schedule maintenance with a more planned, data-driven approach, resulting in higher dependability and cost savings for a variety of sectors. Productivity and operational efficiency are significantly impacted by the efficient management of maintenance procedures in the quickly changing industrial operations environment. This article, which embraces technological developments, explores how ML techniques and the IIoT might be used to improve the predictive maintenance paradigm in industrial settings.

In industrial settings, predictive maintenance is a data-driven, strategic strategy that maximizes equipment durability and reduces unscheduled downtime. Organizations can prevent equipment failures by using ML and advanced analytics to predict possible problems before they arise. Through the use of sensors and data-gathering tools, this approach continuously monitors the state of the equipment, allowing for the examination of several characteristics. After the collection of data, advanced algorithms are employed to detect patterns, trends, and abnormalities, therefore offering significant insights into the condition of industrial machinery. Organizations may go from a more reactive or fixed-schedule maintenance approach to one that is more proactive and efficient with the help of predictive maintenance. Organizations may maximize the operating lifespan of their assets and minimize expensive failures by precisely forecasting when maintenance is required and scheduling interventions just in time.

Predictive maintenance, or PdM, seeks to lower expenses so that businesses may compete more fiercely. It optimizes the maintenance intervention plan by combining sensor data with analytical methods. Optimizing maintenance methods is critical to maintaining operational efficiency and reducing downtime in the complex web of industrial processes. A key tactic that replaces conventional reactive methods with proactive, data-driven ones is predictive maintenance. In the end, this paradigm aims to improve the lifetime and dependability of industrial assets by anticipating breakdowns and facilitating prompt interventions and resource optimization. Predictive maintenance is essential to preventing machine breakdowns and maintaining a high level of production line productivity. The suggested IIoT architecture uses ML techniques to achieve predictive maintenance.

Amidst this paradigm shift, a new age for predictive maintenance in industrial settings has been brought about by the combination of two technical pillars: ML and the IIoT [36].

The symbiotic integration of IIoT technologies makes it possible to monitor equipment continuously and in real-time, producing copious amounts of data that are essential to the development of predictive models. At the same time, ML algorithms that can identify patterns in large datasets raise the bar for predictive maintenance above rule-based systems by providing more detailed insights and improving failure prediction accuracy. There are several layers in the IIoT

*Corresponding Author, email: m.anul@mu.edu.sa

architecture, including those for business connection, device connectivity, and data analytics. Wireless sensor networks provide flexible communication between external and internal equipment, which makes it possible to create effective preventative maintenance plans. One of the key advantages of IIoT is its ability to decrease data transfer to the cloud, which will cut energy consumption and increase forecast accuracy. IIoT architecture is essential for industries where continuous monitoring is necessary since it simplifies data processing and analysis [3,4,5].

Device connectivity, data analytics, and business connectivity make up its three levels. This sophisticated approach promotes a more accurate and efficient maintenance plan by making it easier to identify probable defects and to distinguish between typical changes and important abnormalities. Where human eyes or ears can no longer detect and gather sensitive information from equipment, primarily motors, automated technologies offer a workable option for many sectors [6]. The basis for predictive maintenance schedules is gathered data from sensors and analytical algorithms [7].

The IIoT is a network of smart devices, sensors, and machines that uses the connection to generate a revolutionary change in how industries function. Data-driven decision-making is made possible by the unparalleled volume of data generated by this networked environment. With companies becoming increasingly instrumented and networked, the difficulty is in efficiently utilizing and analyzing this massive amount of data to extract valuable insights.

A key component of the IoT and IIoT environment is Artificial intelligence (AI), which provides the capacity to process, analyze, and understand large datasets at speeds that are not possible with conventional techniques. AI systems are very good at finding trends, abnormalities, and connections in data, turning unprocessed input into useful knowledge. This capacity is especially important in situations when making decisions quickly is required, such as demand forecasting, resource allocation, and manufacturing process optimization. In addition, in order to fulfill consumer requests and keep up with market competitiveness, industrial systems and processes must be regularly monitored and overseen in order to meet the short product lifecycle demands of today's market.

IIoT framework permeates all facets of the automotive industry for predictive maintenance by strategically installing smart sensor devices to perform sensitive operations, with tracking and monitoring playing a major role [23,24,25]. In the context of industry, IoT is known as IIoT and it has gained significant research attention recently [26], [27]. Several sensors are used in IIoT to keep an eye on the operation of machinery or even whole production processes [28]. The goal of the IoT is to simplify our lives. Since its beginning, every industry has made use of it in some way. Traditionally, data collection in IIoT has involved streaming data from sensing devices to the cloud, where it is analyzed and modeled. Sensing equipment produces massive volumes of data, frequently during a short period, either constantly or sporadically. For instance, a machine may produce thousands of records in a second [29]. Computing is revolutionized by ML, which gives computers the ability to see patterns in data and make wise judgments. Models for tasks like classification are trained on labeled data in supervised learning.

Patterns in unlabeled data can be found using unsupervised learning. ML influences day-to-day living through tailored suggestions and virtual assistants. The requirement for labeled data and resolving moral issues with algorithmic biases present challenges. The future of ML will be shaped by developments in fields like ethical issues and reinforcement learning. The cognitive engine that drives IIoT smart systems is ML, a subset of AI.

Because ML algorithms can learn and adapt from past data, they are a good fit for dynamic industrial situations, in contrast to traditional programming. ML has significant uses in real-time decision support, anomaly detection, and predictive maintenance within the IIoT. A paradigm change made possible by ML is predictive maintenance. ML is a cutting-edge application in the field of predictive maintenance in industrial settings. When it comes to managing the complex linkages and varied datasets found in industrial systems, traditional methods frequently fall short. However, ML algorithms are adept at navigating this complexity by noticing trends in data and adjusting to improve their prediction power. The predictive framework integrated with the notions of adaptive structuration theory is shown in Fig. 1. In the structural idea, the maintenance technician watches a facility asset in use. As seen in Fig. 2, it anticipates faults and conducts repairs on the machinery or equipment before they arise. The only machines or components that can be replaced are those that will shortly fail. It prolongs the equipment's lifespan. But usually, they consider the systems viewpoint [30], technical [31], architecture [32], security [33], and [34], or concentrate on the analytics side [35] within the framework of IIoT.

Through the integration of ML techniques and the IIoT, this research aims to enhance the field of predictive maintenance in industrial contexts. Our approach is based on a careful investigation and use of four different Python programs, each carefully designed to maximize the capabilities of different ML models. Because these models' accuracy varies, it is critical to conduct a careful selection process in order to choose the best predictive maintenance plan. Our methodology highlights the subtle nuances of model performance while also demonstrating the revolutionary potential of IIoT and ML in enhancing industrial maintenance. There are several advantages mentioned in [8,9,10].

The rest of this paper is organized as follows: Section I presents a general introduction to Predictive maintenance, IIoT and ML. Section II is about the literature review and Section III details how the system works. Section IV Experimentation, dataset, implementation, and evaluation Metrix. And the last Section we discuss results and comparisons between algorithms.

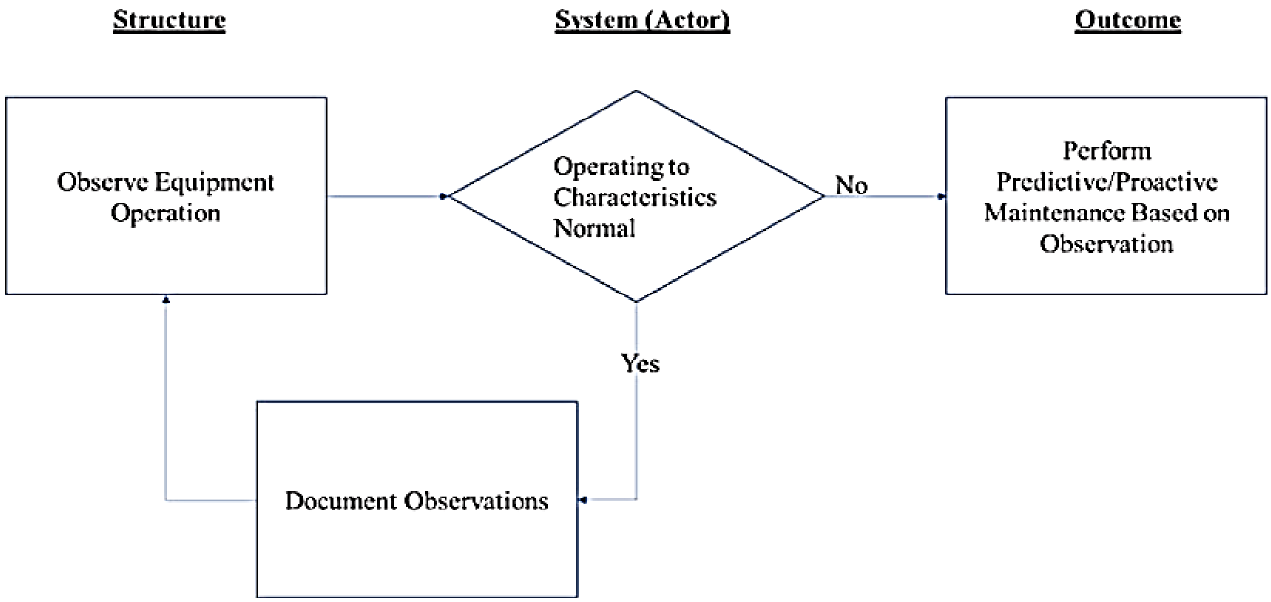

Fig. 1. Predictive maintenance framework.

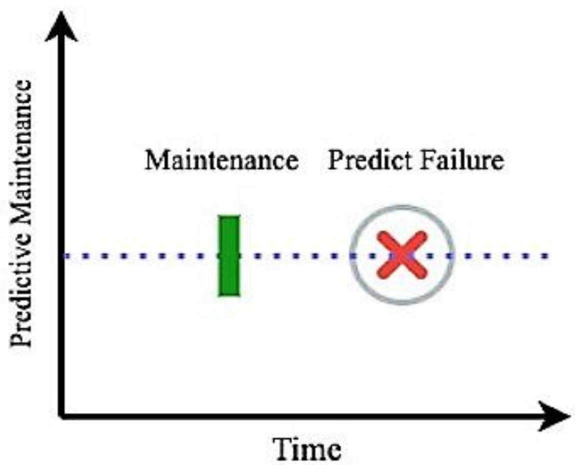

Fig. 2. Predictive maintenance overview.

II. Literature Review

Recent developments in maintenance techniques for manufacturing sectors were examined by Lee et al. [11], who emphasized the move in the era of smart manufacturing from reliability enhancement to flexible and adjustable maintenance scheduling. F. Ribeiro et al. [12] have automatically classified the flaws in rotatory machinery using non-ML approaches like similarity-based models (SBM). In a different research, A. Alzghoul et al. [13] used artificial neural networks (ANNs) to classify rotatory defects with a 97.1 percent accuracy rate. Consequently, their accuracy rate in classifying the defects is 96.43 percent. Singha et al. [14] examined the use of ML and AI in the knitting sector, emphasizing the revolutionary effects of these technologies. The research emphasized the thorough implementation of these technologies at several phases, including product sourcing, design, manufacture, distribution, and sales. Advances in fiber classification, thread prediction, defect diagnosis, and dye recipe prediction are made possible by the integration of AI and ML, which benefits the knitting industry's predictive maintenance.

A mechanism for making fuzzy decisions is devised by et al. [15]. The use of a case study on sewing machine needles, it illustrates how successful it is in planning predictive maintenance. Predictive maintenance was aided by the introduction of an IoT and ML-based online monitoring system for knitting machines by Elkateb et al. [16], [17]. Real-time tracking, statistical analysis, and problem-solving are made easier by this technology. As so, it makes precise productivity monitoring and preventative maintenance possible.

The usefulness of ML-based condition monitoring was the subject of a thorough assessment by Surucu et al. [18], who emphasized the models' major contributions to predictive maintenance. The research employed a Deep Belief Network (DBN) for feature extraction and a Gaussian process (GP) for optimizing DBN hyper-parameters in order to evaluate models utilizing deep learning and Bayesian optimization. Empirical findings outperformed traditional ML techniques in terms of accuracy in predicting machine failure times. Consequently, because of the complexity and distinct contextual elements, cross-case performance comparisons are inadequate. A different study examined an intelligent PdM system for industrial machinery using ML, Message Queuing Telemetry Transport (MQTT), and IIoT [19].

Electrical motors employ vibration, current, and temperature sensors to gather real-time data. Five ML models k-nearest neighbor (KNN), Support Vector Machines (SVM), random forest (RF), linear regression (LR), and Naïve Bayes (NB) are then used to evaluate the data and anticipate failures. Effective communication between sensors, gateways, and the cloud server is made possible via the MQTT protocol. When it comes to functioning motors, RF displays the best accuracy and optimizes maintenance plans to save costs and downtime [19].

A deep learning-based defect diagnostic technique for circular knitting machines was presented by Gao et al. [20]. Their approach classifies the different sorts of faults using a SoftMax classifier after automatically extracting features from vibration signals using a Convolutional Neural Network (CNN). The outcomes of the trial showed that their approach was able to diagnose faults in circular knitting machines with a promising level of accuracy. But for CNN to work well enough, a lot of training data is required. A predictive maintenance system for wind turbines was presented by Udo and Muhammad et al. [21] utilizing SCADA data and Long Short-Term Memory (LSTM) and XGBoost models for gearbox and generator monitoring. Six wind turbine faults were successfully detected using statistical process control (SPC), which evaluates anomalies and helps with early intervention and economical dynamic maintenance plans. Knitting machines are not used in the testing of this system, nevertheless.

In summary, recent research in predictive maintenance for industrial environments has shown hopeful results in improving maintenance efficiency and reducing costs. These studies have applied a variety of algorithms and different attributes to predict Remaining Useful Life (RUL) and to diagnose various faults in the machine. However, there is still a need for further research to develop more accurate, comprehensive, and efficient predictive maintenance systems for circular knitting machines. Diagnosis of different machine faults to achieve comprehensive predictive maintenance systems was not well covered in the literature. Moreover, applications of the developed methods on real working machines outside the laboratory environment were not well covered to prove their applicability in real conditions. To prevent lengthy machine breakdowns, the proposed work offers a predictive maintenance approach that anticipates machine halt and the cause of stoppage (failure).

A comprehensive maintenance approach considers many reasons for failure by utilizing multiple sensing devices. These devices' readings may be accessed by an ML-based classifier via an IoT system. Its distinctive features such as a powerful ML model, real-time monitoring, and an extensive database indicate a shift from traditional methods to contemporary, useful ways. To demonstrate the practicality of the suggested predictive maintenance system, it is put into operation on an actual circular knitting machine. The device performs well and has good precision. It also has a lot of potential to improve machine availability, reduce downtime, and maximize production in the textile sector.

III. METHODOLOGY

Prognostics is the subject of this study, with a focus on estimating an asset's RUL and determining if it is inside its final fifteen cycles. Using a NASA dataset, the research includes engine deterioration simulations in a range of operating scenarios and modes. Fig. 3 presents the overview of the of the entire system. The approach that was selected is based on time-series analysis, using several ML algorithms, and considering each time point as a separate unit. The Random Forest Regressor, Elastic Net GLM, SVM, and Gradient Boosting Regressor are the main models used. The first step is to use a NASA dataset that is kindly shared, which simulates engine deterioration under various operating situations and modes. This dataset captures the subtle progression of problems by recording many sensor channels. Utilizing ML techniques, the selected methodology predicts RUL and identifies assets in the past fifteen cycles by managing each time point individually.

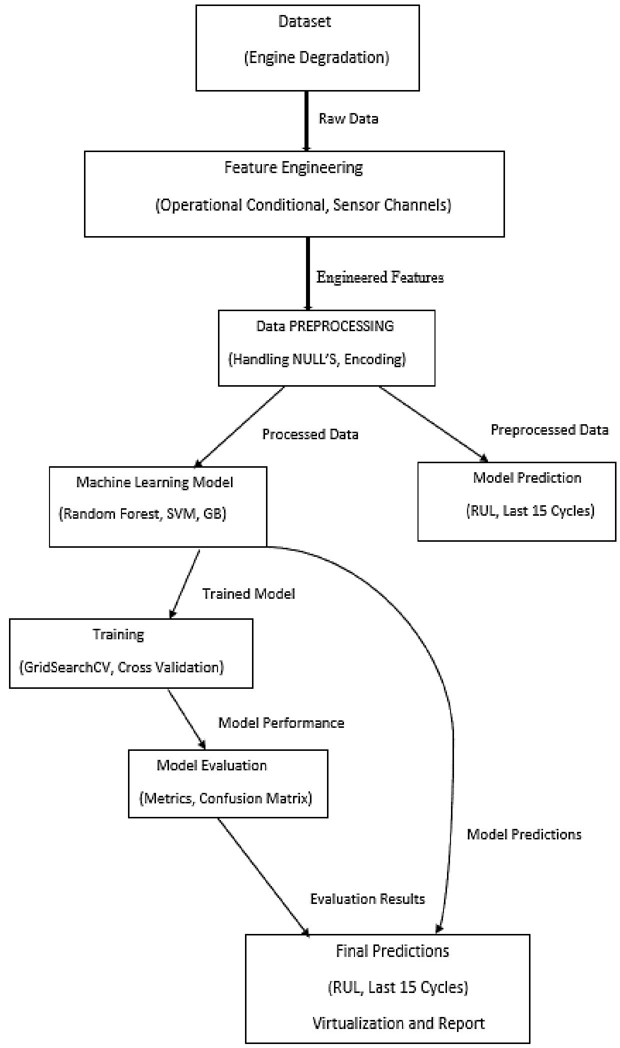

Fig. 3. Workflow of the present investigation.

The procedure starts with the dataset being explored, which includes loading the required packages, reviewing the data that is already accessible, and creating a reproducibility seed. Important stages after importing the data include sensor readings, operational settings, and goal variable structure. Feature engineering becomes essential when the Random Forest Regressor is used to determine which characteristics are most crucial. The code structure and how each phase advances the broader predictive analytics process are delineated in more detail in the Fig. 4. Loading the required packages, reviewing the available data, and establishing seeds for repeatability are all part of the first stage. The NASA dataset is well organized, with distinct column names designating sensor readings, cycle information, and operating parameters.

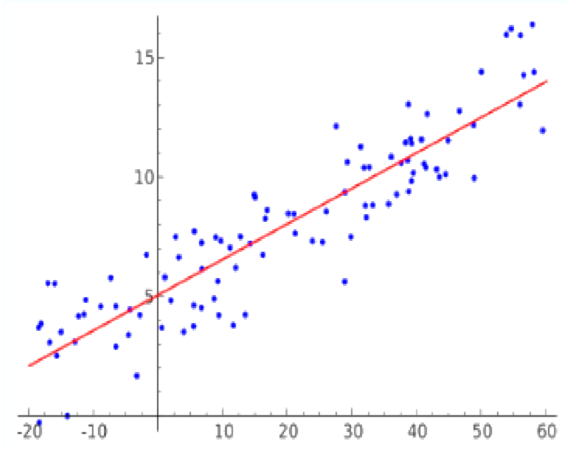

Fig. 4. Linear regression.

A. Linear Regression

Visualizing the correlations between goal variables RUL and Last 15 Cycles) and other attributes is done using exploratory data analysis (EDA). A Random Forest repressor [22] is used in a noteworthy feature selection stage to find important predictors that will impact the next data preparation RUL for each cycle is calculated using the T-minus notation approach. Pair plots and seaborne visualizations offer a comprehensive comprehension of the data distribution. Target leakage management, removing superfluous columns, and getting the data ready for model training are the next steps. To determine feature relevance and enable the elimination of sensors with lower levels of information, the Random Forest repressor is utilized. Identification of numeric and categorical fields is done, with the creation of dummy variables for the former. Any NULL values that remain in numerical columns are addressed by imputation. For training and evaluating the model, the dataset is then divided into training and testing sets.

For regression problems, three different ML models are used: Random Forest, Elastic Net GLM, and SVM. To pick the best hyperparameters, grid search and cross-validation are applied to each model during optimization. Grid search is also utilized for the Gradient Boosting Repressors optimization, which was selected because of its capacity for group learning. Model performance is evaluated using evaluation metrics including R-squared values, Mean Absolute Error, and Mean Squared Error. To determine when an asset is inside its Last 15 Cycles, the regression job must be transformed into a classification issue in the last phase. Recall, precision, and ROC-AUC scores are used in the training and assessment of a Random Forest Classifier. ROC curve visualizations offer more information on classification performance. Fig. 4 shows Linear regression.

B. Decision Tree Regression

Decision Tree Regression (DTR) [22] and LR. The yellow-colored LR illustrates the linear links that exist between RUL and input characteristics. DTR, shown in green, uses a structure akin to a tree and is particularly good at identifying complicated decision boundaries and non-linear patterns in the input characteristics used to forecast RUL.

Within your code, the RUL of engines is predicted using a basic approach called LR. The foundation of linear regression is the creation of a linear connection between the input characteristics and the target variable in this example, the number of operational cycles left until failure. In order to develop a linear model that best represents the connection between these characteristics and the RUL, the algorithm makes use of a collection of input data, such as operational settings and sensor readings. The training data that is supplied, where the real RUL values are known, is used to train the model. The method modifies its parameters during training in order to reduce the discrepancy between the genuine RUL values from the training set and the projected RUL values. Once trained, the RUL of engines not seen during training may be predicted using test data that has not yet been observed. This is known as the Linear Regression model. The model's efficacy is then assessed by comparing the predictions to the actual RUL values using a variety of metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2).

Using an ensemble technique, Random Forest Regression builds many decision trees and combines their predictions. This method reduces overfitting problems and improves accuracy. The fourth method, Gradient Boosting Regression, is the last one. It creates consecutive decision trees to repair the mistakes of the previous ones and can achieve high predicted accuracy.

C. Remaining Useful Life (RUL)

In the given system, Random Forest Regression (RFR) [22] and Gradient Boosting Regression (GBR) are essential for forecasting RUL. RFR uses a group of individual decision trees in its ensemble representation to provide predictions. By adding to the final forecast, each tree improves its resilience and accuracy. In contrast, GBR makes use of a series of succeeding decision trees, each of which steadily improves forecast accuracy by fixing the mistakes of its predecessor. Both techniques increase the predictive model's overall efficacy by using input information (referred to as "Input Features") to forecast the engine's RUL in a range of operational circumstances. Fig. 5 shows the actual RUL with prediction.

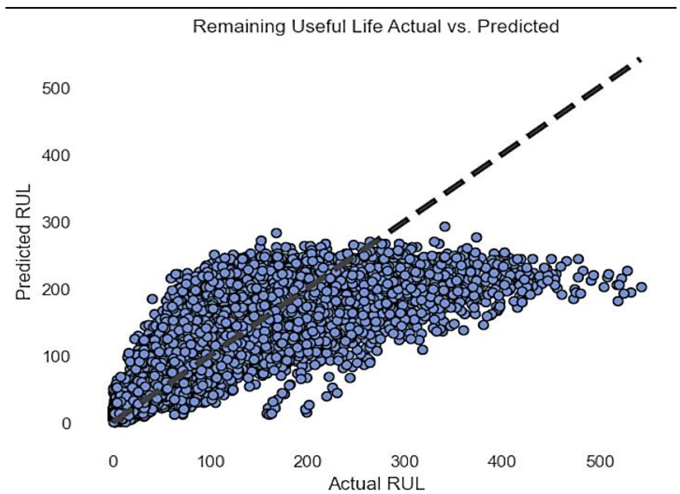

Fig. 5. Remaining useful life.

RFR and GBR are essential for forecasting RUL. RFR uses a group of individual decision trees in its ensemble representation to provide predictions. By adding to the final forecast, each tree improves its resilience and accuracy. In contrast, GBR makes use of a series of succeeding decision trees, each of which steadily improves forecast accuracy by fixing the mistakes of its predecessor. Both techniques increase the predictive model's overall efficacy by using input information (referred to as "Input Features") to forecast the engine's RUL in a range of operational circumstances [22].

The RFR algorithm functions as an ensemble learning technique, as illustrated by the "Random Forest Regression" box. It generates a large number of decision trees, which come together to build a strong and varied model known as the "Ensemble of Decision Trees." The "Input Features" ellipse represents the input characteristics that each tree in the ensemble individually processes and produces a forecast for. Combining the distinct results from each decision tree yields the final forecast. This ensemble method is useful for assessing the engine's RUL in the given system since it reduces overfitting and improves forecast accuracy. Every algorithm offers distinct advantages to the process of predictive modeling. While Decision Tree Regression excels at addressing non-linear patterns, Linear Regression is simple and easy to understand. While Gradient Boosting Regression concentrates on progressively lowering prediction errors, Random Forest Regression provides resilience and control over variance. These methods' diversity guarantees a thorough examination of the dataset and provides insights into how well each algorithm performs in various scenarios.

IV. Experimentation

A. Dataset

The four separate subsets of the dataset [23] that were employed in this investigation are designated as FD001, FD002, FD003, and FD004. These subsets depict various failure mechanisms and operating settings. Four independent subsets comprise the dataset used in this study: FD001, FD002, FD003, and FD004. Each of these subsets has its own unique configurations and failure modes.

1) FD001: There are one hundred test and one hundred train trajectories in FD001. There is just one operational state, which is called "Sea Level." The dataset models a failure mode centered around the degradation of high-pressure compressors (HPCs).

2) FD002: There are 259 test and 260 train trajectories in the FD002 subgroup. With six different operational scenarios, the conditions are more varied. The failure mode addressed is HPC Degradation, much like in FD001.

3) FD003: FD003 has one hundred test and one hundred train trajectories and operates under the same "Sea Level" circumstances as FD001. However, by including two fault modes—HPC Degradation and Fan Degradation—FD003 presents a more complicated scenario.

4) FD004: With six different operational circumstances, the FD004 subset consists of 248 train trajectories and 249 test trajectories. Similar to FD003, FD004 deals with HPC Degradation and Fan Degradation as two failure scenarios.

To put it briefly, the goal of these subgroups is to represent various engine fleet operational conditions and failure types. FD003 and FD004 add more complexity by considering many failure modes under various operational circumstances, whereas FD001 and FD002 concentrate on unique operating conditions with HPC Degradation. The variety of datasets available makes it possible to thoroughly examine engine performance and behavior in various scenarios.

The multivariate time series datasets are separated into training and test trajectories for each subgroup. Every time series relates to a different engine in a fleet of similar engines. The engines show various levels of wear at startup and variance in manufacture that is not communicated to the user. This fluctuation is seen as typical and does not point to a problem. The data includes operational parameters, which have a significant effect on engine performance. Furthermore, noise from the sensors might contaminate the data. Each engine starts in a normal state in the operating context, acquires a defect during the series, and, in the training set, experiences an increasing fault size that results in system failure. The time series in the test set ends before a system failure. Predicting the number of operating cycles left in the test set before failure also known as the RUL is the competition's principal goal. The dataset, which captures different operating parameters and sensor readings during each cycle, is supplied as a 26-column text file that has been compressed using zip.

B. Implementation

Continuing with the implementation, several datasets, including FD001, FD002, FD003, and FD004, each representing distinct operational situations and failure modes, are subjected to iterative applications of the Random Forest Regression method. By training on a variety of datasets, the system takes beginning circumstances and engine wear into consideration. This variety adds to the resilience of the model by improving its capacity to respond to various conditions. The model notices the patterns of engine deterioration that result in system failure during the training phase. The model can capture the complex interactions between operational parameters and sensor readings since the training trajectories imitate both the engine's fault and normal circumstances. To guarantee convergence and avoid overfitting, the number of training epochs and batch size are adjusted.

During the testing phase, the RUL of the engines is predicted by applying the trained Random Forest Regression model to data that has never been seen before. Planning maintenance tasks and predicting breakdowns depend on this predictive capacity. By contrasting the model's projected RUL values with the dataset's ground truth RUL values, the efficacy of the model is thoroughly assessed. The model's accuracy and generalizability to new and varied circumstances are measured using metrics like MSE, MAE, and R-squared. A key factor in determining the model's capacity for generalization is the separation of training and testing trajectories. With its numerous trajectories, the training set replicates the typical wear and fault development patterns of the engines over time. The model can understand complicated correlations between operating parameters and sensor readings because of the variety of training data, which helps it capture the nuanced dynamics of engine health. Conversely, the test set includes trajectories where the engines are nearing the end of their useful lives but are still working. This intentional separation guarantees that the model can produce precise forecasts on brand-new, untested data, proving its dependability in practical situations.

To sum up, the implementation takes a methodical approach to ML, with special emphasis on careful parameter tweaking and reliable assessment techniques. The objective is to develop a predictive model that will be an invaluable resource for predictive maintenance plans in the field of engine health management. This model will not only accurately estimate the Remaining Useful Life of engines but also exhibit resilience and adaptability across a range of operating conditions and fault modes.

C. Evaluation Metrics

In this section, we discuss evaluation metrics that are used in this study [35].

1) Mean Squared error

a) Definition: MSE is the average squared difference between the projected and actual remaining useful life (RUL) values. The dispersion of prediction errors is quantified.

b) Accuracy Measure: By punishing greater errors more severely, the Mean Squared Error (MSE) offers a thorough assessment of the total forecast accuracy. It works well with models when accurate RUL prediction is essential.

$$\text{Formula: } MSE = \frac{1}{n}\sum_{i=1}^{n} | Y_i - \acute{Y}_i| \tag{1}$$

n: Number of data points

Y_i : Actual RUL for data point i

$\acute{Y}_i$: Predicted RUL for data point i

2) Mean Absolute Error

a) Definition: The average absolute deviations between the actual and anticipated RUL values are determined by the MAE. It calculates the typical error magnitude.

b) Accuracy Measure: Because MAE is less susceptible to outliers, it offers a reliable way to quantify average prediction error. It works well with models in which the absolute error is more important than the mistake's particular direction.

$$\text{Formula: } MAE = \frac{1}{n}\sum_{i=1}^{n} | Y_i - \acute{Y}_i| \tag{2}$$

n: Number of data points

Y_i : Actual RUL for data point i

$\acute{Y}_i$: Predicted RUL for data point i

3) R-Squered

a) Definition: R^2 is the percentage that the model explains of the variation in the actual RUL values. It gauges how well the model fits the data.

b) Measure of Accuracy: R^2 has a range of 0 to 1, with 1 denoting a perfect match. It's a helpful measure of how well the model predicts the variability in the data as it really occurs.

$$\text{Formula: } R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_i - \acute{Y}_i)^2}{\sum_{i=1}^{n} | Y_i - \bar{y}_i)^2} \tag{3}$$

n: Number of data points

Y_i : Actual RUL for data point i

$\acute{Y}_i$: Predicted RUL for data point i

ȳ: Mean of the actual RUL values

V. Results and Discussion

This study leverages ML algorithms, including Random Forest Regressor, Elastic Net GLM, SVM, and Gradient Boosting Regressor, to predict an asset's RUL using a NASA dataset. Through systematic data exploration, feature engineering, and model optimization, the methodology yields accurate RUL predictions. Importantly, the model's effectiveness and generalizability are substantiated by rigorous evaluation metrics like MSE, MAE, and R-squared values across diverse operational scenarios. This demonstrates the model's robustness and adaptability, positioning it as a reliable tool for enhancing predictive maintenance strategies in industrial settings.

1) FD001 Dataset: Notable outcomes were obtained from the study of the FD001 dataset, which consisted of one hundred trains and one hundred test trajectories with a single fault mode (HPC Degradation) and a condition of ONE (Sea Level). The Random Forest Regression model with the configuration of (mention configuration details) showed an R-squared of.89, an MAE of 29.89, and an MSE of 1772.26. 0.625 is the value of R^2. These metrics provide important information about how well the model predicts the RUL in these particular circumstances. Furthermore, a confusion matrix was created in order to have a deeper understanding of the model's performance. A more thorough assessment is made 9possible by this matrix, which offers a full picture of true positive, true negative, false positive, and false negative forecasts. Additionally, several curves, such as the learning curve and validation curve, were used throughout the training phase. These curves assist in uncovering any overfitting or underfitting problems by showing the model's convergence and performance over epochs.

2) FD002 Dataset: The applied model was shown using the FD002 dataset, which has 259 test and 260 train trajectories under SIX distinct circumstances and a single fault mode (HPC Degradation) (mention results). This dataset's confusion matrix made it possible to thoroughly assess the model's prediction skills, especially about differentiating between various failure types. The study was deepened by curves produced during training, such as the Precision-Recall and Receiver Operating Characteristic (ROC) curves. These curves shed light on the trade-off between recall and accuracy, respectively, as well as the true positive rate and false positive rate. With 260 train and 259 test trajectories, the FD002 dataset represents a more complex operating scenario that includes a wider range of SIX different circumstances, all of which are shared by a Fault Mode HPC Degradation. The model's capacity to adapt to various operating situations, each with its own set of obstacles for predictive maintenance, is critically tested by this dataset. Operating Diversity: Compared to the FD001 dataset, there are SIX different operating situations, which adds a higher degree of complexity. The circumstances encompass changes in engine loads, temperatures, or other crucial elements, so rendering the dataset an all-encompassing depiction of actual operational scenarios. Thus, the model's capacity to identify trends and modify its predictions over this range of circumstances is put to the test. Common Fault Mode: HPC Degradation: The dataset maintains consistency about the prevalent fault mode, HPC Degradation, even in the face of diverse operational situations. This consistency allows for a targeted assessment of the model's capacity to recognize and forecast a particular fault mode in a range of operating scenarios.

3) FD003 Dataset: Compared to the earlier datasets, the FD003 dataset adds a layer of complexity with its one hundred train and one hundred test trajectories. In this instance, a single operating condition known as Sea Level is applied to all paths. The dataset deviates, though, in that it includes TWO different fault modes: fan degradation and HPC degradation. Singular Operational Condition: Sea Level is the one operational condition that is the focus of the FD003 dataset, as opposed to the SIX operational conditions of the FD002 dataset. This intentional decision isolates the effect of fault modes in a particular operational context, offering information on the model's capacity to identify and anticipate failures in a typical environment. Presenting Several Fault Modes: The model is presented with a more complex task with the addition of TWO failure modes: HPC Degradation and Fan Degradation. This dataset simulates conditions in which several engine components may deteriorate simultaneously or sequentially.

4) FD004 Dataset: The FD004 dataset incorporated 248 train and 249 test trajectories under SIX distinct circumstances with TWO fault modes (HPC Degradation, Fan Degradation), concluding the individual dataset studies. The model's robustness in managing a range of operating situations and fault scenarios is demonstrated by the outcomes (mention results). The model has shown strong prediction skills in the examination of the FD004 dataset, which contains 248 training trajectories and 249 test trajectories under SIX different operating circumstances with TWO fault modes (HPC Degradation, Fan Degradation). The assessment metrics that provide light on the model's ability to adapt to a variety of fault scenarios and operational settings include MSE, MAE, and R-squared. The thorough comprehension that these measurements provide highlights the model's capacity to manage the complexity brought forth by several fault types. Because of its flexibility, the model may be used to provide accurate prognostic evaluations in situations when many engine deterioration modes could occur at the same time. This dataset provides subtle insights that are useful for the overall assessment and for comparing the models' performance across various datasets and fault scenarios.

The first. algorithm which is RFR has the result of sixty-two for the first data set and fifty-eight for the second and sixty-seven for the third and fifty-nine for the fourth. Similarly, if the other algorithms are also compared, JLM's algorithm also gives the result of the first data set fifty-six and the result of the second data set is fifty-six and the result of the third data set is fifty-eight. Similarly, if the third algorithm support vector mechanism is also compared then the result of the first data set is sixty and the second data set is twenty and the third data set is also sixty and the fourth data set is twenty-four. And if the fourth algorithm Grant Boston is compared with each other, the first data set gives the result sixty-two and the second data set gives the result fifty-eight and the third data set gives the result sixty-seven which is the highest and then the fourth one. The result of the data set is fifty-nine. Table I represents the accuracy of Random Forest Regression. Table II represents the accuracy of elastic net glm Table III represents the accuracy of the support vector machine and Table IV represents the accuracy of gradient boosting all these techniques are applied to four codes. In Table III, we have compared the result of the SVM classifier with previously published works [37] and in Table IV, we have compared the result of the Gradient Boosting classifier with previously published works [37].

TABLE I. RESULTS OF THE RANDOM FOREST MODEL

Dataset	Algorithm	RF Mean Squared Error	RF Mean Absolute Error	Accuracy
FD001	Random Forest Regression	17772.26	29.89	0.62
FD002	Random Forest Regression	1945.94	32.475	0.58
FD003	Random Forest Regression	3160.17	38.51	0.67
FD004	Random Forest Regression	3209.63	40.68	0.59

TABLE II. RESULT OF ELASTIC NET GLM MODEL

Dataset	Algorithm	GLM Mean Squared Error	GLM Mean Absolute Error	Accuracy
FD001	GLM	2043.03	34.560	0.56
FD002	GLM	2043.03	34.60	0.56
FD003	GLM	4083.08	47.16	0.58
FD004	GLM	4503.76	51.59	0.43

TABLE III. RESULT SUPPORT VECTOR MACHINE MODEL

Dataset	Algorithm	SVM Mean Squared Error	SVM Mean Absolute Error	Accuracy	In past research[37] Accuracy
FD001	SVM	1860.48	30.28	0.60	0.893
FD002	SVM	3774.31	48.79	0.20	0.894
FD003	SVM	3846.14	41.04	0.60	0.893
FD004	SVM	6052.88	58.43	0.24	0.106

TABLE IV. RESULT OF GRADIENT BOOSTING MODEL

Dataset	Algorithm	GB Mean Squared Error	GB Mean Absolute Error	Accuracy	In past research [37] Accuracy
FD001	Gradient Boosting	1768.62	29.92	0.62	0.899
FD002	Gradient Boosting	1970.56	33.03	0.68	0.908
FD003	Gradient Boosting	3190.07	38.68	0.67	0.903
FD004	Gradient Boosting	3214.30	41.08	0.59	0.097

A thorough examination and explanation of the outcomes of applying ML models to a variety of engine datasets to forecast Remaining Useful Life (RUL). MSE, MAE, and R-squared are three quantitative measures of the model's performance under different operating settings and failure types that are part of the assessment metrics. Analyzing the FD001, FD002, FD003, and FD004 datasets separately exposed unique difficulties and intricacies. The model showed adequate prediction skills in the case of FD001, where conditions were comparatively simpler with a single failure mode (HPC Degradation). After switching to FD002, which included six operating conditions under the same fault mode, the model performed admirably, demonstrating its flexibility in a variety of situations.

The prediction work became more challenging in FD003 due to the addition of additional failure modes. We closely examined the model's capacity to manage both HPC Degradation and Fan Degradation situations. Even with this extra complexity, the model demonstrated proficiency in capturing the subtleties brought forth by numerous failure types. Finally, the robustness of the model was demonstrated by the assessment of FD004, which included two failure modes (HPC Degradation and Fan Degradation) and six different operational situations. The outcomes demonstrated its ability to handle a wider range of fault states and operating scenarios, which makes it a flexible tool for prognostic evaluations.

The comparative study of the models across datasets is also covered in detail in the discussion, with a focus on the differing levels of complexity brought about by various operational situations and failure mechanisms. The comparison study yielded insights that help comprehend the flexibility and generalization capabilities of the models, offering useful information for potential future applications in engine health prognostics. The limits of the study, probable causes of bias, and possibilities for development are also covered in the discussion. There are opportunities for more study and improvement of the prediction models because of the models' resilience and performance in real-world situations in various areas [38-42]. The discussion part, which provides a nuanced view of the models' strengths, limits, and potential implications in the field of engine system prognostics, summarizes the findings overall.

VI. CONCLUSION

This research presents a pivotal advancement in the realm of ML applied to industrial maintenance through its integration with the IIoT. Practically, the study equips industries with an advanced, data-driven methodology for equipment maintenance, leading to reduced downtime, cost savings, and heightened operational efficiency. Theoretically, it enriches our understanding of ML's efficacy in predictive maintenance, facilitating the refinement of algorithms and informing more judicious model selections. Using a variety of datasets (FD001, FD002, FD003, and FD004), this study concludes with a thorough examination of the use of ML models for forecasting RUL of engines. The study effectively illustrates how well the models handle various operating situations and failure types, providing insightful information about engine health prognostics. The models' prediction ability is quantified using assessment metrics such as MSE, MAE, and R-squared. fault mode (HPC Degradation) and more straightforward conditions. After switching to FD002 with various operating circumstances, the model keeps performing admirably, highlighting its adaptability.

Although FD003 presents a difficulty due to the addition of new failure modes, the model can handle cases when there is both HPC Degradation and Fan Degradation. The assessment of FD004 under various circumstances and fault types demonstrates the adaptability and efficiency of the models in a range of operational contexts. The models' generalization skills are illuminated by the comparison study between datasets, which provides important information about their advantages and disadvantages. Even while the results are encouraging, it is important to recognize some limitations and potential topics for more research. Further investigation is needed on the models' susceptibility to biases and variances in sensor data. To sum up, this study establishes the foundation for further developments in prognostic modeling and highlights the value of strong ML methods for improving the precision and dependability of forecasts in various areas[43-46].

REFERENCES

[1] W. Lee, H. Wu, H. Yun, H. Kim, M. Jun, J. Sutheralnd T. Zonta, C.André da Costa, R. da Rosa Righi, M. de Lima, E. da Trindade, G. Li

[2] J. Guth, U. Breitenbücher, M. Falkenthal, P. Fremantle, O. Kopp et al., "A detailed analysis of IoT platform architectures: Concepts, similarities, and differences," In Internet of Everything: Springer, pp. 81–101, 2018.

[3] C. Federico, B. Stefano, S. Claudio, R. Enrico, M. Luca et al., "Industrial internet of things monitoring solution for advanced predictive maintenance applications," Journal of Industrial Information Integration, vol. 7, pp. 4–12, 2017.

[4] J. Wang, L. Zhang, L. Duan and R. X. Gao, "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing," Journal of Intelligent Manufacturing, vol. 28, no. 5, pp. 1125–1137, 2017.

[5] H. M. Hashemian and W. C. Bean, "State-of-the-art predictive maintenance techniques," IEEE Transactions on Instrumentation and measurement, vol. 60, no. 10, pp. 3480–3492, 2011.

[6] S.-j. Wu, N. Gebraeel, M. A. Lawley, and Y. Yih, "A neural network integrated decision support system for condition-based optimal predic tive maintenance policy," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 37, no. 2, pp. 226–236, 2007.

[7] "What Are Benefits and Drawbacks of Preventive Maintenance?" https://www.onupkeep.com/answers/preventive-maintenance/benefits-of-preventive-maintenance (accessed Jun. 29, 2021).

[8] "What is Predictive Maintenance? [Benefits & Examples]," Fiix. https://www.fiixsoftware.com/maintenance-strategies/predictive-maintenance/ (accessed Apr. 7, 2021).

[9] Y. Ageeva, "Predictive Maintenance Scheduling with AI and Decision Optimization," Medium, May 15, 2020.

[10] J. Lee, J. Ni, J. Singh, B. Jiang, M. Azamfar, J. Feng J. Manuf. Sci. Eng., 142 (2020), pp. 1-40, 10.1115/1.4047856

[11] F. Ribeiro, M. Marins, S. Netto, and E. Silva, "Rotating machinery fault diagnosis using similaritybased models," presented at the XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2017. doi: 10.14209/sbrt.2017.133.

[12] A. Alzghoul, A. Jarndal, I. Alsyouf, A. A. Bingamil, M. A. Ali, and S. AlBaiti, "On the Usefulness of Pre-processing Methods in Rotating Machines Faults Classification using Artificial Neural Network," Journal of Applied and Computational Mechanics, Jan. 2021, doi: 10.22055/jacm.2020.35354.2639.

[13] K. Singha, S. Maity, P. Pandit Use of AI and machine learning techniques in knitting 10.1016/B978-0-323-85534-1.00021-0

[14] C. Baban, M. Baban, S. Darius Using a fuzzy logic approach for the predictive maintenance of textile machines 10.3233/IFS-151822

[15] S. Elkateb, A. Métwalli, A. Shendy, An Innovative Online Monitoring System in Knitting Industry, The 16th Textile Bioengineering and Informatics Symposium; Blended Conference. 412–419, August 22–25(2023). DOI: TBIS 10.3993/tbis (2023).

[16] S. Elkateb, A. Métwalli, A. Shendy, K. Moussa, A. Abu-Elanien Online monitoring-based prediction model of knitting machine productivity Fibres 10.2478/ftee-2023-0035

[17] O. Surucu, S.Andrew Gadsden, J. Yawney Condition monitoring using machine learning: a review of theory, applications, and recent advances ExpertSyst.Appl 10.1016/j.eswa.2023.119738

[18] N. Mohammed, O. Abdulateef, A. Hamad An IoT and machine learning-based predictive maintenance system for electrical motors J. Eur. Des. Systèmes Autom., 56 (4) (2023), pp. 651-656, 10.18280/jesa.560414

[19] Y. Gao, C. Chai, H. Li, W. Fu. A deep learning framework for intelligent fault diagnosis using automl-cnn and image-like data fusion Machines, 11 (10) (2023),p. 932, 10.3390/machines11100932.

[20] W. Udo, Y. Muhammad Data-driven predictive maintenance of wind turbine based on SCADA data IEEE Access, 9 (2021), pp. 162370-162388, 10.1109/ACCESS.2021.3132684

[21] https://www.infoq.com/articles/machine-learning-techniques-predictive-maintenance/

[22] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation", in the Proceedings of the Ist International Conference on Prognostics and Health Management (PHM08), Denver CO, Oct 2008.

[23] Y. He, C. Gu, Z. Chen and X. Han, "Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis," International Journal of Production Research, vol. 55, no. 19, pp. 5841–5862, 2017.

[24] S. Landset, T. M. Khoshgoftaar, A. N. Richter and T. Hasanin, "A survey of open-source tools for machine learning with big data in the hadoop ecosystem," Journal of Big Data, vol. 2, no. 1, pp. 24, 2015.

[25] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," Journal of Industrial Information Integration, vol. 6, no. 1, pp. 1–10, 2017.

[26] E. Oztemel and S. Gursev, ''Literature review of industry 4.0 and related technologies,'' J. Intell. Manuf., vol. 31, no. 1, pp. 127–182, Jan. 2020.

[27] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, ''Industrial Internet of Things: Recent advances, enabling technologies and open challenges,'' Comput. Electr. Eng., vol. 81, Jan. 2020, Art. no. 106522.

[28] D. Sehrawat and N. S. Gill, ''Smart sensors: Analysis of different types of IoT sensors,'' in Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI), Apr. 2019, pp. 523–528.

[29] L. D. Xu and L. Duan, ''Big data for cyber physical systems in industry 4.0: A survey,'' Enterprise Inf. Syst., vol. 13, no. 2, pp. 148–169, Feb. 2019.

[30] H. Xu, W. Yu, D. Griffith, and N. Golmie, ''A survey on industrial Internet of Things: A cyber-physical systems perspective,'' IEEE Access, vol. 6, pp. 78238–78259, 2018.

[31] Aceto, V. Persico, and A. Pescape, ''A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges,'' IEEE Commun. Surveys Tuts., vol. 21, no. 4, pp. 3467–3501, Aug. 2019.

[32] I. Sittón-Candanedo, R. S. Alonso, S. Rodríguez-González, J. A. G. Coria, and F. De La Prieta, ''Edge computing architectures in industry 4.0: A general survey and comparison,'' in Proc. Int. Workshop Soft Comput. Models Ind. Environ. Appl., 2019, pp. 121–131.

[33] .Zhang, H. Huang, L.-X. Yang, Y. Xiang, and M. Li, ''Serious challenges and potential solutions for the industrial Internet of Things with edge intelligence,'' IEEE Netw., vol. 33, no. 5, pp. 41–45, Sep. 2019.

[34] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, ''Industrial Internet of Things: Challenges, opportunities, and directions,'' IEEE Trans. Ind. Informat., vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[35] C. Krupitzer, T. Wagenhals, M. Züfle, V. Lesch, D. Schäfer, A. Mozaffarin, J. Edinger, C. Becker, and S. Kounev, ''A survey on predictive maintenance for industry 4.0,'' 2020, arXiv:2002.08224. [Online]. Available: http://arxiv.org/abs/2002.08224.

[36] S. Gopalakrishnan and M. Senthil Kumaran, "Iiot framework based ml model to improve automobile industry product," *Intelligent Automation & Soft Computing*, vol. 31, no.3, pp. 1435–1449, 2022.

[37] Nangia, S., Makkar, S., & Hassan, R. (2020, March). IoT based predictive maintenance in manufacturing sector. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*.

[38] M. Anul Haq, "CDLSTM: A Novel Model for Climate Change Forecasting," Comput. Mater. Contin., vol. 71, no. 2, pp. 2363–2381, 2022, doi: 10.32604/cmc.2022.023059.

[39] M. A. Haq, "SMOTEDNN: A Novel Model for Air Pollution Forecasting and AQI Classification," Comput. Mater. Contin., vol. 71, no. 1, pp. 1403–1425, 2022, doi: 10.32604/cmc.2022.021968.

[40] M. A. Haq et al., "Analysis of environmental factors using AI and ML methods," Sci. Rep., vol. 12, no. 1, pp. 1–16, 2022, doi: 10.1038/s41598-022-16665-7.

[41] M. A. Haq, "DBoTPM : A Deep Neural Network-Based Botnet," Electronics, vol. 12, no. 1159, pp. 1–14, 2023, [Online]. Available: https://www.mdpi.com/2079-9292/12/5/1159

[42] M. A. Haq and M. A. R. Khan, "Dnnbot: Deep neural network-based botnet detection and classification," Comput. Mater. Contin., vol. 71, no. 1, pp. 1729–1750, 2022, doi: 10.32604/cmc.2022.020938.

[43] C. S. Yadav et al., "Malware Analysis in IoT & Android Systems with Defensive Mechanism," Electronics, vol. 11, no. 15, p. 2354, 2022, [Online]. Available: https://www.mdpi.com/2079-9292/11/17/2799

[44] A. Kumar, S. A. Alghamdi, A. Mehbodniya, M. anul haq, J. L. Webber, and S. N. Shavkatovich, "Smart power consumption management and alert system using IoT on big data," Sustain. Energy Technol. Assessments, pp. 1–7, 2022.

[45] Lakshmanna, K.; Kaluri, R.; Gundluru, N.; Alzamil, Z.S.; Rajput, D.S.; Khan, A.A.; Haq, M.A.; Alhussen, A. A Review on Deep Learning Techniques for IoT Data. Electronics 2022, 11, 1604. https://doi.org/10.3390/electronics11101604

[46] S. S. P. D. M. A. H. A. K. Sathishkumar Karupusamy J. Refonaa, "Effective energy usage and data compression approach using data mining algorithms for IoT data," Expert Syst., vol. 12997, pp. 1–10, 2022.