IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2022.Doi Number*

# Explainable Predictive Maintenance of Rotating Machines using LIME, SHAP, PDP, ICE

**SHREYAS GAWDE[1#], SHRUTI PATIL[2], SATISH KUMAR[2], POOJA KAMAT[1], KETAN KOTECHA[2] and SULTAN ALFARHOOD[3]**

[1] Symbiosis Institute of Technology Pune Campus, Symbiosis International (Deemed University) (SIU), Lavale, Pune 412115, India
[2] Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis Institute of Technology Pune Campus, Symbiosis International (Deemed University) (SIU), Lavale, Pune 412115, India
[3] Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O.Box 51178, Riyadh 11543, Saudi Arabiasultanf@ksu.edu.sa
[#] *Goa University, Goa, India,* Goa, India

Corresponding author: Dr. Shruti Patil (e-mail: shruti.patil@sitpune.edu.in)

"This research is funded by the Researchers Supporting Project Number (RSPD2023R890), King Saud University, Riyadh, Saudi Arabia."

**ABSTRACT** Artificial Intelligence (AI) is a key component in Industry 4.0. Rotating machines are critical components in manufacturing industries. In the vast world of Industry 4.0, where an IoT network acts as a monitoring and decision-making system, predictive maintenance is quickly gaining importance. Predictive maintenance is a method that uses AI to handle potential problems before they cause breakdowns in operations, processes or systems. However, there is a significant issue with the AI models' (also known as "black boxes") inability to explain their decisions. This interpretability is vital for making maintenance decisions and validating the model's reliability, leading to improved trust and acceptance of AI-driven predictive maintenance strategies. Explainable AI is the solution because it provides human-understandable insights into how the AI model arrives at its predictions. In this regard, the paper presents Explainable AI-based predictive maintenance of Industrial rotating machines. The proposed approach unfolds in four comprehensive stages: a) Multi-sensor based multi-fault (5 different fault classes) data acquisition, b) Frequency-domain statistical feature extraction, c) Comparison of results for multiple AI algorithms, and d) XAI integration using "Local Interpretable Model Agnostic Explanation (LIME)", "SHapley Additive exPlanation (SHAP)", "Partial Dependence Plot (PDP)" and "Individual Conditional Expectation (ICE)" to interpret the results.

**INDEX TERMS** Explainable AI, ICE, Industry 4.0, Industrial Rotating Machines, LIME, PDP, Predictive maintenance, SHAP.

## I. INTRODUCTION

In the realm of industrial operations, the implementation of predictive maintenance strategies has become pivotal for ensuring the optimal performance and longevity of rotating machines [1]. However, the black-box nature of many predictive maintenance models can present challenges in understanding the reasoning behind their predictions, potentially hindering their adoption in critical applications. This article delves into the realm of Explainable Predictive Maintenance (XPM) for rotating machines, employing advanced techniques such as Local Interpretable Model-agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), Partial Dependence Plots (PDP), and Individual Conditional Expectation (ICE) to shed light on the intricate decision-making processes within these models. By unraveling the complexity of predictive maintenance algorithms, we aim to enhance transparency, trust, and usability, ultimately empowering industry professionals to make informed decisions about the maintenance and performance optimization of the rotating machinery.

Rotating machines are critical components in manufacturing industries [2]. The uninterrupted functioning of these machines is the top priority of maintenance engineers. Various condition-monitoring techniques are available to maintain these critical machines; however, they demand a maintenance expert to interpret the analysis. Researchers have been working to

create a generalised method for fault diagnosis in rotating machines for the past few years [3], focussing mainly on a) fault pattern identification and b) developing a classification algorithm to distinguish the faults based on the patterns. Recently, predictive maintenance techniques have been getting important where multiple sensors' data is used to predict the machinery condition using various AI algorithms [4]. Online condition monitoring is another rising technique that allows online access to the health data of these machines [5].

To implement predictive maintenance, the first step is data collection, where the researchers have used either online data [6] or manually collected data [7]. The latter is a better option, as collecting data manually on the test setup allows more faults to be incorporated under different conditions. Multiple sensors and multiple types of sensors used for data collection give room for better condition monitoring of machines. Sensors such as accelerometers, acoustic, temperature, and current sensors are effective while doing predictive maintenance of rotating machines [7]. After data collection, the next step is signal processing and feature engineering, extracting meaningful information from the raw data. Each fault type gives a unique vibration pattern that can be analysed in the time or frequency domain. Compared to the time domain signals, the signals in the frequency domain are better interpretable by maintenance engineers due to their reduced complexity [8][9] and Fault Characteristic Frequencies [10]. For example, an unbalance in the machinery is depicted in the FFT spectrum by a 1x peak at rotational frequency. Similarly, Misalignment in the machinery is depicted by 1x and 2x peaks at the rotational frequency. However, there is no such clear distinction in the time-domain signal. Hence, the maintenance engineers prefer analysis by collecting data in the frequency domain or converting time domain data to the frequency domain using Fast Fourier Transform (FFT). It is also seen that some statistical features such as RMS, Kurtosis, Crest Factor, Standard deviation, Shape Factor, peak frequency and corresponding amplitude, etc., are also effective in identifying different faults in rotating machines—for example, the more the crest factor, the healthier the bearing. The crest factor also provides early signs of fault occurrence. Also, the Kurtosis value is less than or equal to 3 for healthy bearings. The level of skewness increases as the faults rise. Increased RMS denotes faulty condition. The increased amplitude at 1Xrpm denotes Unbalance.

Extracted features provide quantifiable information that aids maintenance engineers in making informed decisions about the machinery's condition and the appropriate course of action based on their experience and training [11]. While feature extraction plays a significant role in manual fault diagnosis by maintenance engineers, its importance also extends to automated fault diagnosis. Automated systems can uncover subtle or complex relationships in the data that might not be immediately apparent to human operators [9]. By leveraging advanced algorithms, these systems can detect anomalies and patterns that could go unnoticed in manual analysis. Various algorithms used by the researchers for data-driven predictive maintenance are Machine Learning (ML) algorithms such as Support Vector Machine (SVM) [12][13][14][15], Random Forest (RF) [16][17][18][19], K-Nearest Neighbour (KNN) [16][20][21][22], Decision Tree (DT) [23][24][25], Artificial Neural Network (ANN) [26][27][28][29] and Deep Learning (DL) algorithms such as Deep Neural Network (DNN) [27][30][31][32], Recurrent Neural Network (RNN) [33][34], Convolutional Neural Network (CNN) [35][36][37][38][39], Long Short-Term Memory (LSTM) [40][34][41][42], Auto Encoder (AE) [43][6][44][45], etc. Some researchers have also used the hybrid of ML and DL algorithms for better results [15]. A systematic literature review on multi-fault diagnosis in rotating machines is addressed by authors in [4] and [46].

Different ML and DL algorithms are used for fault detection, classification, or Remaining Useful Life (RUL) Prediction. Results also show that the accuracy is very high using these algorithms. However, some issues or research gaps still need to be addressed.

- Rotating machines are a group of driver and driven machines comprising multiple components, which tend to possess multiple faults. When a fault is simulated on the test setup for data collection, it is imperative to validate it with the help of condition monitoring experts. It is seen from the maximum literature that there is no proof of data validation at the data acquisition stage.
- Most literature mentions the data being extracted in the time domain, which is difficult to validate due to the signal complexity. The data can be easily validated using the FFT Spectrum by mapping the unique fault frequencies. Hence, frequency-domain raw FFT data is needed for data validation.
- Also, when PdM systems employing AI predict that a component will fail and must be replaced, the engineer or the customer has many concerns since a considerable amount of cost is associated with the maintenance or replacement of machines or components. These traditional black box models do not offer clear explanations for their predictions on the health and condition of industrial machinery. This lack of transparency makes it difficult for maintenance personnel to understand why a machine is flagged as faulty or when maintenance is needed.

The eXplainable AI (XAI), an emerging field, provides a clear and understandable method and assists in solving all of the company's issues. Explainable Artificial Intelligence (XAI) in Predictive Maintenance refers to integrating interpretability techniques into AI models for predicting and detecting faults in industrial machinery and equipment.

When an AI model detects a fault in a machine, maintenance experts need to understand the reasons or root causes behind the prediction. XAI techniques offer interpretable insights into the features and factors contributing to the prediction, facilitating accurate root cause analysis and targeted maintenance interventions. Maintenance personnel can complement the AI model's insights with their domain knowledge, resulting in a more robust and reliable maintenance strategy. Understanding the model's decision-making process can also identify and address potential errors, improving the model's overall performance. Table I analyses the Applications of some XAI techniques related to Predictive Maintenance in different domains.

TABLE I
APPLICATIONS OF SOME XAI TECHNIQUES RELATED TO PREDICTIVE MAINTENANCE IN DIFFERENT DOMAINS

| | | Application of XAI Techniques | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ref.** | **Domain** | ***LIME*** | ***SHAP*** | ***Rule Based*** | ***Tree Based*** | ***Saliency*** | ***LRP*** | ***GradCAM*** | ***Counterfactual Explanations*** |
| [47] | PdM in Hard Disk Drives | ✓ | ✓ | | | | | | |
| [48] | PdM in Battery, Engine, Gearbox | ✓ | ✓ | | | ✓ | ✓ | ✓ | |
| [34] | PdM in Bearings | | | | | | ✓ | | |
| [49] | PdM in Bearings | | ✓ | | ✓ | | | | |
| [50] | Predictive Business Process Monitoring | | | | ✓ | | | | ✓ |
| [51] | PdM in Healthcare | | ✓ | | | | | | |
| [52] | PdM in Aerospace Industry | ✓ | ✓ | | | | | | |
| [53] | PdM in Water Pumping Ind. | | | ✓ | | | | | |
| [54] | Prediction in Banking | | | | | | | | ✓ |
| [55] [56] [57] | Prediction in Healthcare | ✓ | ✓ | | | | | | |
| | | | ✓ | | | | | | |

The contribution of the study is as follows:

- The study makes a significant contribution by introducing an Explainable Predictive Maintenance technique designed to address key issues identified in the existing research gaps.
- Our approach is exemplified through a comprehensive case study that leverages FFT raw data and employs multi-sensor data fusion for nuanced multi-fault diagnosis in Industrial Rotating Machines. To assess the efficacy of our proposed technique, we undertake a thorough comparison of various AI algorithms, including SVM, Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbour (KNN).
- In order to enhance the interpretability of the AI models employed, we employ advanced techniques such as "Local Interpretable Model Agnostic Explanation (LIME)," "SHapley Additive exPlanation (SHAP)," "Partial Dependence Plot (PDP)," and "Individual Conditional Expectation (ICE)." These methodologies not only contribute to the transparency of our predictive maintenance approach but also provide valuable insights into the inner workings of the AI models, addressing a critical aspect often overlooked in traditional predictive maintenance strategies.

By combining innovative predictive maintenance techniques with a comprehensive comparative analysis of AI algorithms and a commitment to interpretability through state-of-the-art methods, our study presents a holistic and impactful contribution to the field, bridging crucial gaps in existing research and setting a new standard for Explainable Predictive Maintenance in the domain of Industrial Rotating Machines. Fig. 1 shows the graphical representation of the study proposed.

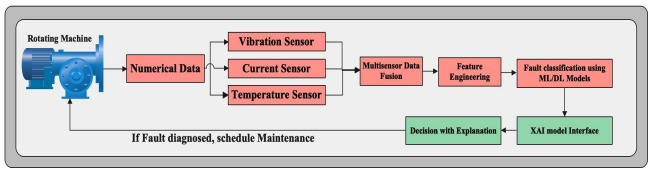



**FIGURE 1. Block diagram of the proposed study**

The rest of the paper is organised as follows: Section II gives the methodology related to Explainable Predictive Maintenance, covering topics such as AI Algorithms, Explainable AI models using LIME, SHAP, PDP and ICE, Fast Fourier Transform (FFT) Data Generation. The results of the case study are presented in Section III. Section IV is a Discussion section that presents a comprehensive analysis of results while addressing significant challenges and the future scope of the work. Section V concludes the paper.

## II. METHODOLOGY

Let us analyse the materials and methods required in the present study.

### *A. MULTI-CLASS CLASSIFICATION ALGORITHMS*

Several algorithms can be used for multi-class classification in the context of fault diagnosis or fault classification. The choice of algorithm depends on various factors such as the dataset's size and nature, the fault patterns' complexity, interpretability, and available resources. It is recommended to experiment with different algorithms and compare their performance on the specific fault classification task at hand. The algorithms that were used for multiclass classification are Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Decision Tree (DT) and Random Forest (RF). This paper refrains from explanations of these algorithms, given their widely recognized standard definitions available in existing literature [58][16][25][14]. Readers are encouraged to consult established sources for comprehensive insights

into these well-known machine learning techniques. This approach streamlines content, allowing a focus on the core aspects of the proposed methodology.

### B. EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI)

Explainable AI, also known as interpretable AI, refers to the concept of developing and designing artificial intelligence systems and algorithms that can provide understandable explanations for their decisions and predictions. The XAI aims to provide a collection of new or enhanced ML approaches that produce explainable models that, when combined with strong explanation methodologies, allow end-users to grasp, effectively trust, and ensure the successful management of the next generation of AI technologies [59]. Explanation techniques may be classified according to a set of criteria. The first is the Traditional techniques for data explanation. This comprises exploratory analysis and visualisation techniques (such as dimensionality reduction and clustering) as well as model performance evaluation metrics (such as accuracy, precision, recall, ROC curve, mean absolute error for regression models, AUC for classification models, and coefficient of determination (R-square), root-mean-square error). These traditional approaches are quite useful in better understanding our data (Data explanation), characteristics, and which models are most likely to be productive. However, they are extremely limited when it comes to attempting to figure out how a model works in a way that humans will understand. Hence, as discussed below, we go for new approaches for a better model explanation.

The new XAI techniques for Model explanation are of two types. The first is to use algorithms that produce explainable models. Such models are called Interpretable models or Transparent Models [60]. This approach is also called the Intrinsic approach. Linear Regression, Logistic Regression, tree-based models, rule-fits, k-NN, and Naive Bayes are frequently employed in this approach [61]. These algorithms are explainable by themselves. These models provide competitive accuracy, but their performance depends on data quality, model complexity, and tuning. However, in some cases, these models may yield low accuracy due to their inherent simplicity, which can struggle to capture complex relationships in data and underfitting. Some of these models may be too simple, necessitating the development of new methods for building and interpreting more complicated and high-performing models. This can be achieved by separating the explanations from ML Models. These second types of techniques involve the models that can be explained using external XAI techniques called extrinsic or post-hoc explainable approaches [60]. Since the interpretation techniques may be applied to any ML model after the model training, the extrinsic approach provides a significant benefit over the intrinsic approach in terms of flexibility, as the developers are free to use any ML algorithms [61]. Post-hoc Explainability can be applied to the intrinsically explainable model, which is model-specific, and also to other black-box AI models, which are Model-Agnostic. Model-agnostic techniques may be used on any machine learning model and are utilised after the model has been trained (post hoc). Post-hoc explainable techniques include explanations based on text or visual explanations, local explanations, using an example, by simplification, and also by giving feature-relevant explanations [62]. Functional Decomposition, Partial Dependence Plot (PDP), Individual Conditional Expectation (ICE), Accumulated Local Effects (ALE) Plot, Feature Interaction, Permutation Feature Importance, Global Surrogate, Local Surrogate (LIME), Scoped Rules (Anchors), Shapley Values, SHAP (SHapley Additive exPlanations), etc. are some examples of Post-hoc explainability techniques. Another broad way of classification of XAI techniques is based on the scope of Explanation, that is, Local (explaining a single prediction) or Global (explaining the entire model) [63]. The XAI concept is illustrated in Fig. 2.

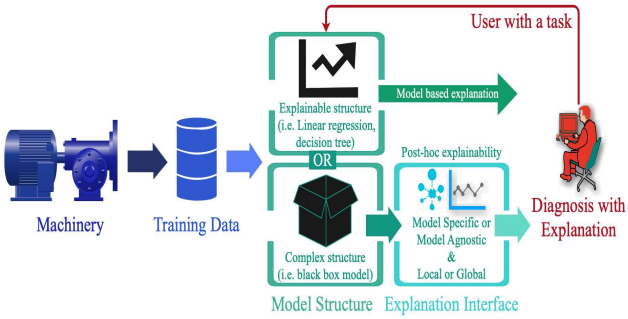



**FIGURE 2. Concept of Explainable AI.**

The AI Model takes input from the current task to recommend or decide. Model structure can aim for direct knowledge of model architecture (by using directly explainable models) or use the black box models to explain the model after studying model behaviour (post-hoc Explainability). Post-hoc explainability techniques are either model-specific (applied to transparent models) or model-agnostic (applied to other AI models). Furthermore, post-hoc explainability approaches are characterised as global or local (global: for explaining what the model learned from the entire variable space; local: for explaining how each prediction is made based on the values at the instance)[74]. Finally, the user makes the decision based on the Explanation. In this article, four of the post-hoc explainable models are studied for a local and global explanation as follows:

#### 1) LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS (LIME)

LIME (Local Interpretable Model-agnostic Explanations) provides local explanations by approximating a complex model's decision boundaries with a simpler, more interpretable model near a specific data instance. LIME can

be applied to any machine learning model, regardless of its underlying algorithm or architecture. LIME supports three types of input formats: tabular data, text data and image data.

Let us understand the working of LIME for tabular data as the data to be analysed in this article is in the tabular form: Select a specific data sample or an instance from the feature set that you want to explain the program's diagnosis, say "Machine Data A." LIME will create slightly different versions of "Machine Data A" by making small changes to its values (Perturbations). For example, it might slightly increase the temperature or change the RMS level. These new data samples are different versions of "Machine Data A." LIME uses a simple, easy-to-understand model (like a linear model or decision tree) to explain what is happening with the complex model. LIME fits the simple interpretable model using the perturbed data points and the corresponding complex model predictions. Essentially, it creates a simplified model that represents how the complex model behaves near "Machine Data A". Next, LIME looks at all the fault predictions to find patterns. It wants to understand which features (like temperature or RMS) influenced the program's diagnosis the most. Based on these patterns, LIME will tell you which feature values were most important in the program's decision for "Machine Data A." It might say, "The high RMS levels and the increased temperature had the biggest impact on predicting Fault Type X for this machine." By using LIME, you can get a clear explanation of why the program diagnosed a specific fault for "Machine Data A." This helps you understand which sensor/feature values are critical in determining the fault and why the program made that diagnosis. It makes the program's fault diagnosis more understandable and helps maintenance experts make better decisions to keep the rotating machines running smoothly. Fig. 3 gives the pseudocode format for working of LIME.

```
FUNCTION LIME(tabular_data, black_box_model, instance_to_explain, num_samples, num_features):
        selected_instance = tabular_data[instance_to_explain]

        #Create Perturbations
        perturbations = create_perturbations(selected_instance, num_samples, num_features)

        #Generate Perturbed Instances
        perturbed_instances = apply_perturbations(selected_instance, perturbations)

        #Obtain Predictions for Perturbed Instances
        perturbed_predictions = []
        FOR each perturbed_instance IN perturbed_instances:
                perturbed_predictions.append(black_box_model.predict(perturbed_instance))

        #Train an Interpretable Model (Linear Regression)
        interpretable_model = train_interpretable_model(perturbed_instances, perturbed_predictions)

        #Calculate Feature Importance
        feature_importance = compute_feature_importance(interpretable_model)

        RETURN feature_importance
```

**FIGURE 3. Pseudocode format for working of LIME.**

### 2) SHAPLEY ADDITIVE EXPLANATIONS (SHAP)

SHAP (Shapley Additive Explanations) is another interpretability technique that can be used to explain the predictions of machine learning models. SHAP values are based on cooperative game theory and provide a unified framework for explaining the contribution of each feature to the prediction outcome. SHAP can be applied to various types of models, including tree-based models, linear models, and deep neural networks. SHAP can generate local explanations by calculating the SHAP values for individual data instances. These local SHAP values represent the contribution of each feature to the prediction outcome for a specific instance. By aggregating the SHAP values across multiple instances, SHAP can also provide global explanations. This allows for understanding each feature's overall importance and impact across the entire dataset. SHAP can be applied to explain predictions in multi-class classification scenarios for tabular data. When working with tabular data and multi-class classification, SHAP extends its methodology to handle multiple classes and provide explanations specific to each class.

Let us understand the working of SHAP: choose a specific data sample from a rotating machine that you want to explain the program's diagnosis. Let us call it "Machine Data A." SHAP will consider all possible combinations of features (or sensor readings) for "Machine Data A" to understand how each feature affects the fault prediction. For each combination of features, SHAP calculates a "SHAP value" for each feature. This value tells us how much each feature influenced the program's prediction for "Machine Data A." By considering all the SHAP values, SHAP shows you which features were most crucial in the program's decision for "Machine Data A." it might say, "The high kurtosis level and the increased temperature had the biggest impact on predicting Fault Type X for this rotating machine." SHAP also considers how each feature's absence in a combination affects the prediction. This helps you understand the importance of each feature when it is not present along with others. The SHAP values obtained for each feature and each class are used to generate explanations for the model's prediction on the selected instance in the multi-class context. Visualisation techniques such as Shapley value plots, summary plots, or individual feature importance plots can be employed to present the SHAP values for each class separately. These plots illustrate the impact of each feature on the prediction outcome for each class, helping to understand the model's decision-making process and feature importance across multiple classes in a multi-class classification problem. Fig. 4 is the pseudocode format for working of SHAP.

```
FUNCTION SHAP(instance_to_explain, black_box_model):

#Generate all possible coalitions of features
coalitions = generate_coalitions(instance_to_explain)

#Calculate SHAP values for each feature
shap_values = []
FOR each feature IN instance_to_explain:
        shap_value = calculate_shap_value(feature, coalitions, black_box_model)
        shap_values.append(shap_value)

RETURN shap_values
```

**FIGURE 4. Pseudocode format for working of SHAP.**

### 3) PARTIAL DEPENDENCE PLOT (PDP)

A Partial Dependence Plot (PDP) is a data visualisation tool used in machine learning and statistical analysis to

understand the relationship between a specific feature (variable) and the predicted outcome (target) while keeping other features fixed or at a constant value. PDPs provide insights into how changes in the chosen feature affect the model's predictions.

In PDP, we choose one of the machine's features (e.g., vibration amplitude at a certain frequency) as the feature of interest. Determine a range of values for the selected feature. This range could span from the minimum to the maximum observed values in your dataset. For each value within the defined range of the feature, create synthetic data points by keeping all other features fixed. Use the trained model to predict the likelihood of a fault occurring for each synthetic data point. Calculate the average prediction for each value of the feature. This provides an estimate of how the likelihood of a fault changes as the feature of interest varies while keeping other factors constant. Create a PDP plot where the x-axis represents the values of the chosen feature, and the y-axis represents the average predicted probability of a fault occurring. The plot will show how changes in the selected feature influence the likelihood of a fault. Analyse the PDP to understand the relationship between the selected feature and fault occurrence. Look for patterns, thresholds, or non-linearities indicating specific conditions or values associated with higher fault probabilities. Based on the PDP analysis, you can determine thresholds for the feature that, when exceeded or fallen below, indicate an increased likelihood of a fault. These thresholds can be used for real-time fault detection. Deploy the trained model and the PDP-derived thresholds in a monitoring system for continuous fault detection in the rotating machine. The system can issue alerts or trigger maintenance actions when feature values exceed specified thresholds.

By using PDPs in this way, you can gain a deeper understanding of the relationships between machine features and fault occurrences, enabling more effective fault detection and predictive maintenance in rotating machines. Fig. 5 shows the working of PDP in pseudocode format.

```
# Load the dataset, encode labels, and train a Random Forest classifier
# Fit the model to the data
# Define the feature of interest
# Define a range of values for the feature
feature_range = linspace(min(X[feature_of_interest]), max(X[feature_of_interest]), num=100)
# Initialize an array to store PDP values
pdp_values = []
# Calculate the PDP values for each value in the feature range
for value in feature_range:
  X_pdp = copy(X)
  X_pdp[feature_of_interest] = value
  # Predict class probabilities for this modified data point
  class_probs = clf.predict_proba(X_pdp)
  # Calculate the mean class probabilities across all classes
  pdp_value = mean(class_probs, axis=0) # Replace 0 with the index of the class you are interested in
  append pdp_value to pdp_values
# Plot the PDP
plot(feature_range, pdp_values)
```

**FIGURE 5.** Pseudocode format for the working of PDP.

#### 4) INDIVIDUAL CONDITIONAL EXPECTATION (ICE)

An Individual Conditional Expectation (ICE) plot is a data visualisation technique used in machine learning and statistical analysis to understand how a single feature affects the predicted outcome of a model for individual data points. ICE plots provide a more granular view of the relationship between a feature and predictions by showing multiple curves, each representing the effect of the feature on the outcome for different data instances**.**

In ICE, we select one of the relevant features as the "feature of interest." This could be, for example, the vibration amplitude at a specific frequency or the temperature at a certain location in the machine. For each instance in the dataset, a separate ICE curve is generated. To do this, all other features are held constant, and only the chosen feature of interest is varied within its range. An AI/ML model is used for each data point to make a prediction. For each data point, an ICE curve is plotted. The x-axis represents the variations in the chosen feature, and the y-axis represents the predicted values or fault likelihood as the feature changes. Each ICE curve shows how the predicted outcome changes with different values of the feature while keeping other features fixed. Looking at the non-linearities or abrupt changes in the curves may indicate fault-related behaviours. For example, you may find that certain vibration frequencies are strongly correlated with specific types of machine faults. ICE plots can also be used to validate and interpret the performance of the AI model used. If the ICE curves align with expectations and known fault behaviours, it builds confidence in the model's correctness. Applying ICE analysis to fault detection in rotating machines can gain insights into the relationship between individual sensor measurements and the presence of faults. This can aid in feature selection, model interpretation, and ultimately improve the accuracy of fault detection systems. Fig. 6 shows the working of ICE in pseudocode format.

```
# Load the dataset, encode labels, and train a Random Forest classifier
# Choose a specific feature for which you want to plot ICE curves
# Find the unique values of the feature
unique_values = unique_values_in_column(df[feature_of_interest])
# Create an empty array to store ICE values
ice_values = empty_array(shape=(len(unique_values), number_of_classes))
# Calculate ICE values for each unique value of the feature
for i, value in enumerate(unique_values):
  # Create a copy of the original data with the feature set to the current value
  X_ice = copy(X)
  X_ice[feature_of_interest] = value
  # Predict class probabilities for this modified data point using the Random Forest model
  class_probs = predict_proba(RandomForestClassifier, X_ice)
  # Store the class probabilities for each class
  ice_values[i, :] = class_probs[0]
# Plot the ICE curves for each class
for class_index, class_label in enumerate(class_labels):
  plot(unique_values, ice_values[:, class_index], label='Class ' + class_label)
```

**FIGURE 6.** Pseudocode format for working of ICE.

### C. FREQUENCY DOMAIN ANALYSIS USING FFT SPECTRUM

FFT transforms a time-domain signal into its frequency-domain representation. By analysing the frequency content of a signal, you can identify specific patterns or signatures associated with faults or anomalies. Different faults often

produce distinct frequency components, such as harmonics, which can be detected through FFT analysis. The traditional practice is to regularly monitor a machine or system's FFT data to identify early signs of developing faults. Changes in the frequency spectrum, such as the appearance of new frequencies or variations in the amplitude of existing frequencies, can indicate the presence of a fault even before it becomes severe enough to cause noticeable issues. This early detection allows for timely intervention and preventive maintenance.

Sampling in both time- and frequency domains is equivalent as long as the Nyquist-Shannon theorem conditions are met. However, the data in the frequency domain is preferred in this study to highlight its utility in certain scenarios where specific frequency-related patterns are of particular interest, primararily while validating the data being collected in “real-time” and diagnosing the type of fault in predictive maintenance. Firstly, collecting frequency-domain raw data enables the researchers to validate the data being collected at the very important data acquisition stage in real-time, which may not be as feasible if time-domain raw data is collected. The FFT spectrum generated by Industrial VibeXpert II was used to validate the FFT spectrum generated by the Data Acquisition (DAQ) setup acquired from the test setup while the data was being collected in real-time. The top 5 frequency peaks were validated on both the Data Acquisition devices, which were found to be similar at the respective frequencies. Fig. 7 compares the FFT Spectrum on VibXpert II and the data acquisition technique used. Secondly, the decisions of Black-box AI models or the faults predicted by AI models can also be validated using the input FFT data since FFT data is human-interpretable. Fault characteristic frequencies in the FFT spectrum enable maintenance engineers to diagnose the type of fault in the machinery, which is difficult in time-domain representation due to signal complexity. Hence, in the case of fault prediction by AI model, the predicted fault can be validated in using the FFT data. From the above-mentioned perspective, FFT data analysis can be preferred over time-domain data.

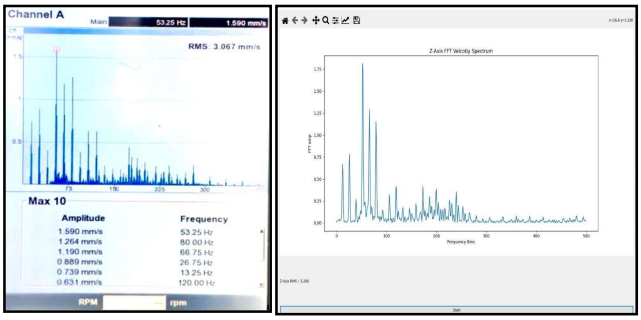



**FIGURE 7. Data validation using FFT spectrum generated by VibXpert II.**

While time-domain data still holds value in certain applications, FFT data provides a more comprehensive, insightful and human-understandable representation of the underlying machinery behaviour, making it a preferred choice in predictive maintenance. There are two ways to obtain the frequency domain data from an accelerometer. One is to collect time domain data and convert it into FFT data using steps such as Data Acquisition, preprocessing, applying the FFT algorithm, getting the FFT Spectrum, and analysis. The other way is to use an accelerometer that would give direct FFT output. In this article, FFT Data was acquired using four Industrial grade, screw lock-type, stainless steel, triaxial vibration gauge VB-310 SCB with built-in signal processing capabilities to convert the raw acceleration data into the frequency domain. With the direct FFT output, you can directly analyse the frequency content of the accelerometer data without explicitly performing the FFT calculation yourself. However, it's important to note that the features of accelerometers can vary depending on the specific model or manufacturer. So, it's recommended to refer to the accelerometer's specifications to understand the details of its FFT output and how to interpret the provided frequency domain information.

### *D. MULTI-FAULT DATASET GENERATION USING MULTI-SENSOR DATA FUSION*

Generating a multi-fault dataset involves creating a dataset with multiple types of faults or anomalies. As shown in Fig. 8, the test setup was used for data collection. It consists of 3 phase induction motor, Variable Frequency Drive (VFD), shaft with a diameter of 3 cm and a total length of 95 cm, three-Jaw spider Coupling connecting the driver and driven shafts, two Timken 22207KEJW33C3 double-row spherical bearings, two Deep groove ball bearings within the induction motor, two rotors: Simply supported rotor with a diameter of 25 cm and over-hung rotor with a diameter of 35 cm. The setup is mounted on a Mild Steel (MS) foundation supported by a concrete base. The test setup is designed to simulate multiple faults such as bearing faults, unbalance, Misalignment, and structural Looseness.

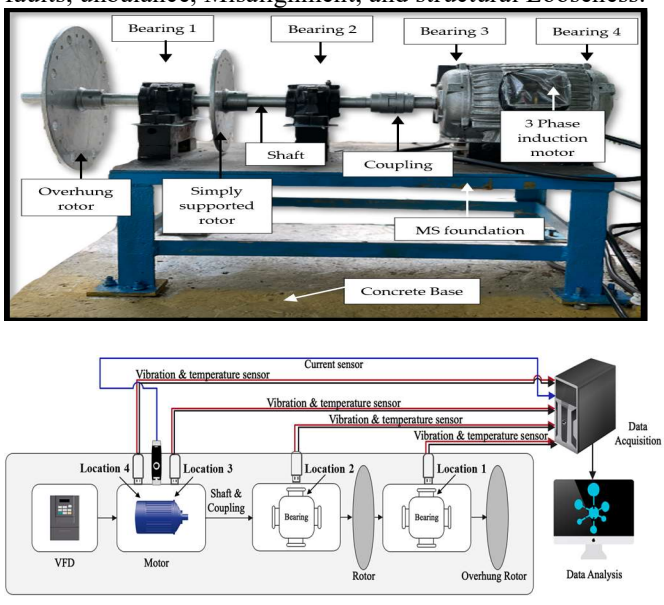



**FIGURE 8. The Test Setup used for Experimentation.**

The test setup is employed with four VB-310 SCB accelerometers mounted at each bearing location. Multiple sensors enable more precise diagnosis, and the use of multiple "types" of sensors enables fusing multiple condition monitoring techniques, giving scope to accurately diagnose multiple faults at early stages. To achieve this, along with accelerometers, four MAX6675 thermocouple temperature sensors are mounted at each bearing location and one PZEM 004T current sensor around the electrical cable of the motor. The connection diagram for various sensors is as shown in Fig. 7.

The VB-310 SCB accelerometer features an all-in-one integrated design that combines both the sensor and transducer within a compact package. Its output signals are in an industrial RS-485 format. The output data includes three-axis velocity-acceleration RMS (Root Mean Square), three-axis velocity-acceleration FFT (Fast Fourier Transform), and three-axis displacement peak-to-peak measurements. Notably, the VB-310 SCB internally collects time-domain data and provides direct FFT raw data as the output. This accelerometer can be seamlessly connected to third-party Programmable Logic Controllers (PLCs) and Distributed Control Systems (DCS) without the need for Data Acquisition (DAQ) devices, as it supports open signal protocols like MODBUS Remote Terminal Unit (RTU). When connecting the sensor to a Personal Computer (PC), an RS485 to Universal Serial Bus (USB) converter is employed. Moreover, the accelerometer can be configured to obtain FFT velocity data in X, Y, or Z direction. Similarly, PZEM 004T current sensor facilitates data transfer via an RS485 communication interface, making it compatible with a PC for data acquisition. Additionally, the MAX6675 thermocouple temperature sensor can be easily affixed to the bearing housing for temperature monitoring. MAX6675 is connected to Arduino via SPI protocol. Arduino UNO has been configured to be a Modbus slave using the library "ModbusRTUSlave". Arduino UNO is connected to PC directly without any converter. Python application connects to Arduino to collect the temperature data. Hence, the VB310 SCB and PZEM 004 T are connected to PC via USB to RS485 convertor and MAX6675 is connected to PC via Arduino UNO. All sensors in this system employ the Modbus-RTU protocol at the application layer for communication. Python application is developed to interface and collect data from all the above mentioned devices and store it in csv file so that it can fed to AI algorithm for further processing.

Data is collected from multiple types of multiple sensors for 5 different fault types (single plane unbalance, two plane unbalance, misalignment, looseness and no-fault) at three different speeds (800 rpm, 1000 rpm and 1200 rpm). Multiple sensor data fusion is achieved using feature-level fusion. Feature-level fusion is a key process that involves combining the features obtained from multiple sensors into a unified feature vector. This amalgamation necessitates careful data mapping during the data acquisition phase. In our case, we collected FFT data (equal to 2000 datapoints) from four sensors, each measuring data along three axis, temperature data from four sensors, and current data from one sensor making a Total= ((4 * 3) + 4 + 1) = 17 datapoints as one set. From this set, various features were extracted. It is important to note here that, although the data was collected from four different locations, however, data from that sensor nearest to the fault location for that particular fault type was considered for further evaluation. 10 Standard statistical features are extracted from FFT data (of 2000 datapoints) such as RMS, Mean, Standard Deviation, Variance, Kurtosis, Crest Factor, Shape Factor, Impulsion Factor, Sum of Squares, and Skewness from all three axis along with Current and Temperature (Total= (10 features * 3 axis) + 1 temp + 1 current= 32 features). These features from one set are mapped into a single feature vector, thus achieving feature-level fusion. Likewise, 150 sets are collected at each speed, 450 sets at 3 speeds (one fault condition), 2250 sets for all the 5 fault conditions. The final feature set consists of 2250 rows and 32 features columns. The final feature set is analysed using multiple AI models, followed by XAI models' explanations.

## III. RESULTS

This section discusses the results obtained after using AI models for multi-fault classification. Multi-fault classification is explained using LIME and SHAP in subsequent subsections.

### *A. MULTI-FAULT CLASSIFICATION USING AI MODELS*

The final feature set obtained after multi-sensor data fusion is further processed using feature scaling techniques such as Standardization and Normalization. These techniques are essential when features have a diverse range. Further feature ranking techniques were employed to select the essential features. Random Forest and XGBoost techniques were used for feature ranking. Fig. 9 and Fig. 10 show the visualisation of important features using Random Forest and XGBoost, respectively.

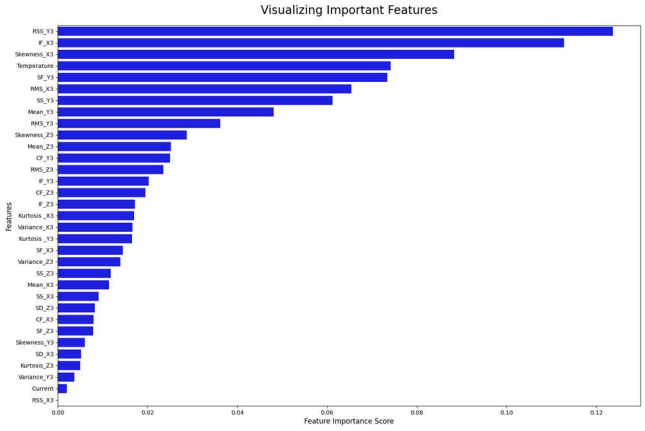



**FIGURE 9. Feature Importance using Random Forest.**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3367110



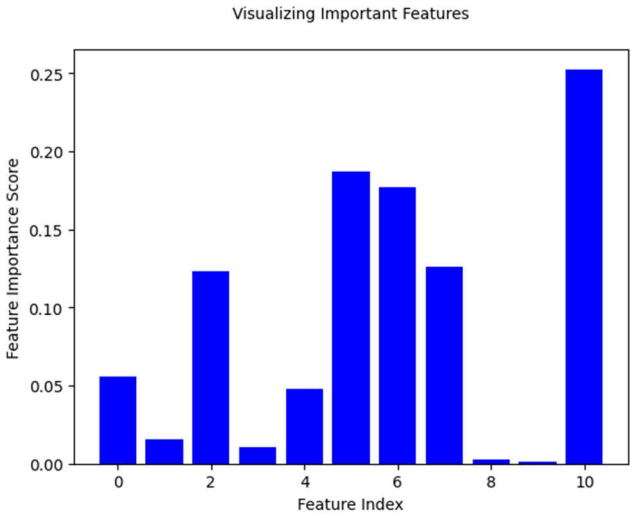


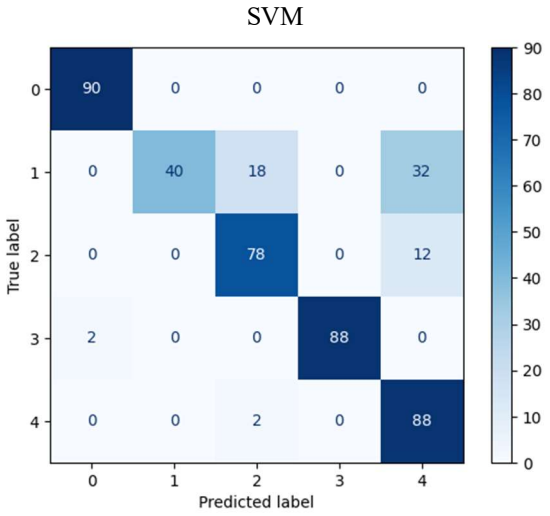

**FIGURE 10.** **Feature Importance using XGBoost.**

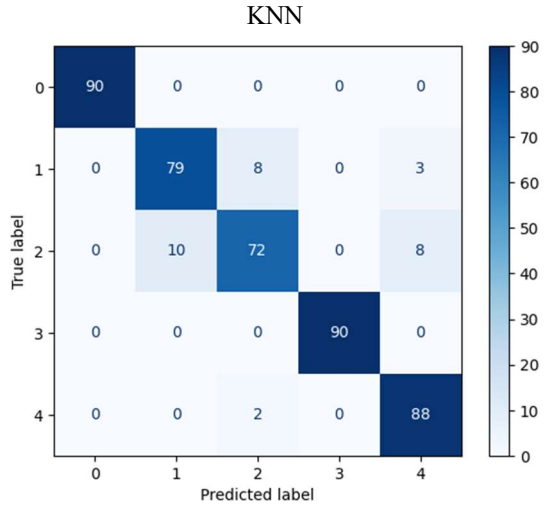
Finally, 9 features out of 32 were selected for further classification using AI Models. Multi-class classification algorithms such as SVM, KNN, Decision Tree and Random Forest are applied, and the results are analysed in Table II.

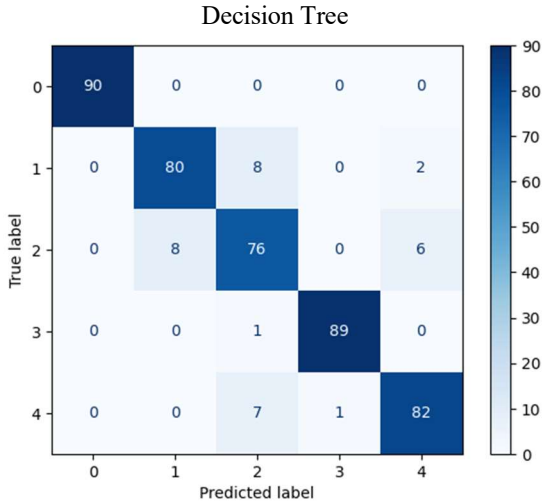
TABLE II
RESULTS BY VARIOUS MULTI-CLASS CLASSIFICATION ALGORITHMS

| **Algorithms** | **Results** | | | |
|---|---|---|---|---|
| | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1 score*** |
| SVM | 85% | 0.89 | 0.85 | 0.84 |
| KNN | 93% | 0.93 | 0.93 | 0.93 |
| Decision Tree | 93% | 0.93 | 0.93 | 0.93 |
| Random Forest | 96% | 0.96 | 0.96 | 0.96 |

The algorithm was run 10 times for each model, and the average accuracy is displayed in the table. It is seen that Random Forest gives the highest Accuracy of 96%, followed by Decision Tree and KNN, giving 93% accuracy, and SVM, giving 85% accuracy. Fig. 11 shows the confusion matrix of all four classification models. It can be seen that the Accuracy has reduced due to the misclassification of 1st and 2nd classes, that is, Single-plane and Two-plane Unbalance classes. A one-way ANOVA test was conducted to compare the performance of different machine learning algorithms. The result obtained from running the one-way ANOVA test indicates that there are statistically significant differences in the performance of the different machine learning algorithms. The F-statistic is a measure of the variation between the means of the algorithms relative to the variation within each group. A larger F-statistic indicates a greater difference between group means compared to the variation within each group. The F-statistic was 1520.00 in this case, indicating substantial variation between the algorithms' performances. The P-value represents the probability of observing extreme variation in group means (or more extreme) if there were no real differences between the groups. The P-value obtained was 5.997389484070222e-38, indicating strong statistical evidence suggesting that at least one of the algorithms performs differently in accuracy compared to the others. In summary, the result indicates that there are statistically significant differences in the accuracy of the machine learning algorithms.



This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3367110



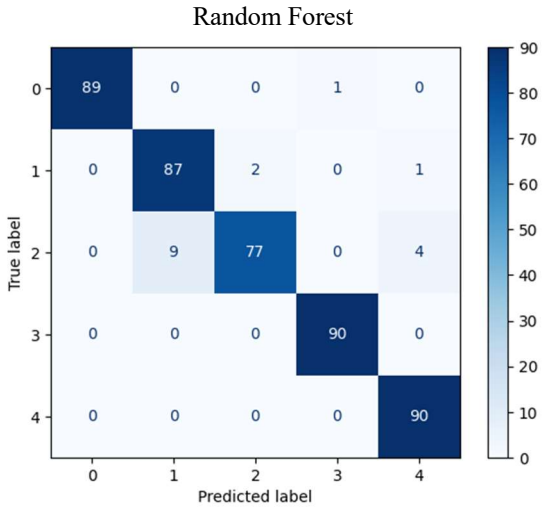
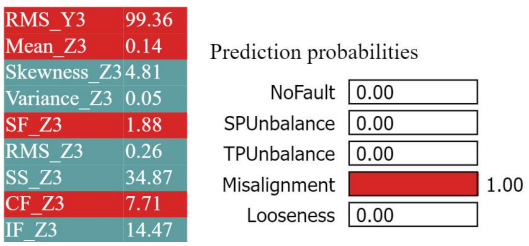



**FIGURE 11. Confusion Matrix for SVM, KNN, DT, RF.**

### *B. LOCAL EXPLANATIONS USING LIME*

To understand the classification implemented by the AI models, LIME is used. LIME gives local explanations. This means it will explain why an instance was classified as a particular class based on its features. Fig. 12 is a typical output that is explained by the LIME interpreter for the 300$^{th}$ instance of a Random Forest Classifier. It can be seen that the model predicts the output of the 300$^{th}$ instance as Misalignment with 100% confidence (prediction probability). LIME also identifies the most influential features that contribute to the prediction for the specific instance. To quantify the relevance of the selected features, LIME gives weights to them. These weights represent the influence of each feature on the model's prediction for the instance. Higher weights imply a greater effect, whereas weights near zero or zero indicate less influence. In this situation, the feature RMS and Mean impact the predicted outcome most. To visually display the impact of these features, the weights are also depicted using various graphic elements such as bar lengths or colour intensity. In addition to the weights, LIME also provides an explanation based on the value of features, as shown below. The feature representation and its associated value may not precisely correlate to the original dataset's feature values since LIME generates and manipulates these values during the perturbation process.



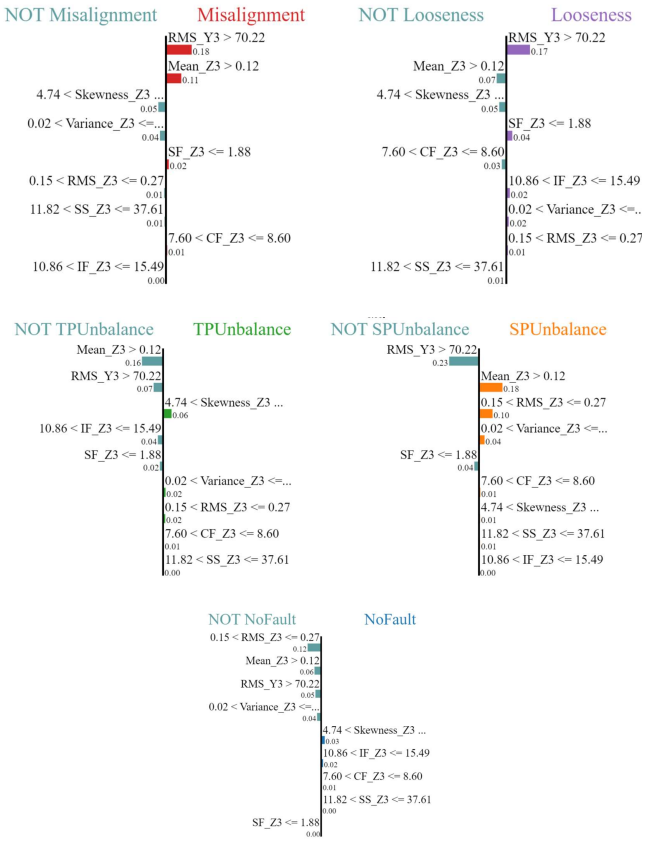




**FIGURE 12. Explanation by LIME for 300$^{th}$ instance of RF classifier.**

Fig. 13 depicts the prediction probability of the 300$^{th}$ instance explained by LIME for the KNN, Decision Tree and SVM model. It is seen that SVM classifies the 300$^{th}$ instance as Misalignment with 28% confidence and the instance as Looseness with 72% confidence.

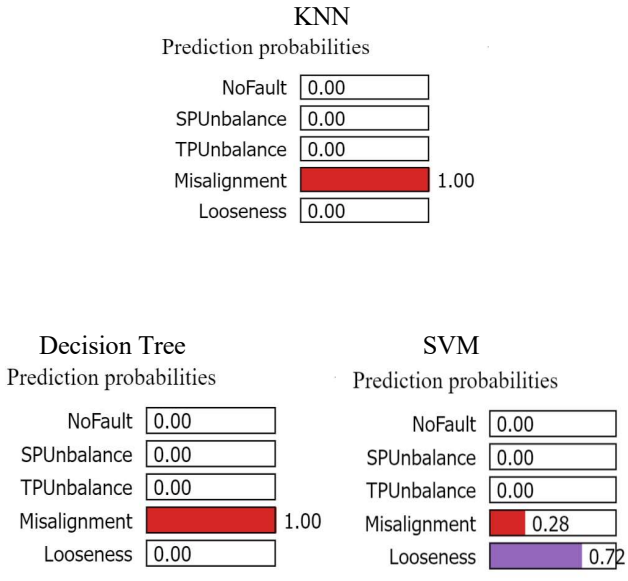
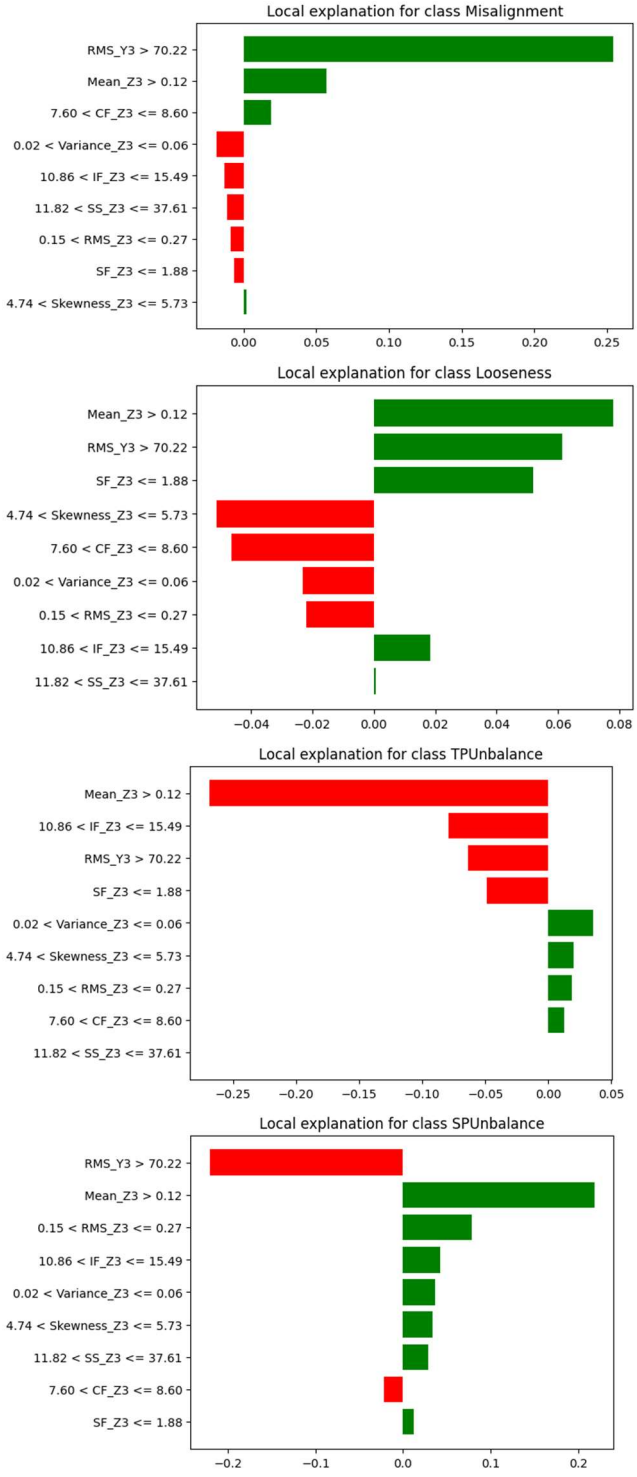



**FIGURE 13. Prediction Probability explained by LIME for the 300$^{th}$ instance of KNN, DT and SVM.**

Fig. 14 depicts the local Explanation by presenting feature contributions to each class using LIME. This method allows you to see the explanations for each class independently, obtaining insights into the feature relevance for each class prediction in a multi-class classification scenario. Fig. 15 depicts the real global feature significance weights generated from random forest and decision tree models. The significance is arranged in descending order to show the most significant elements at the top.



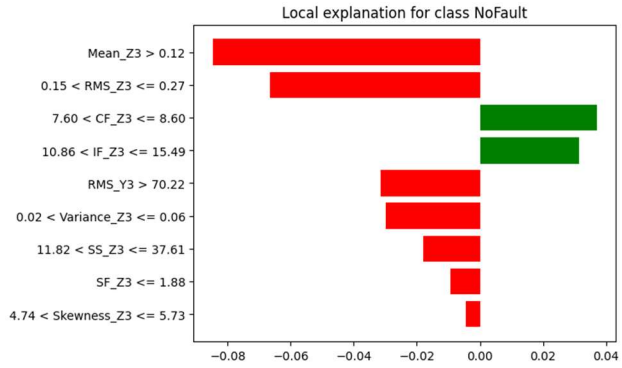




**FIGURE 14. Local Explanation by LIME for Individual Fault Class.**

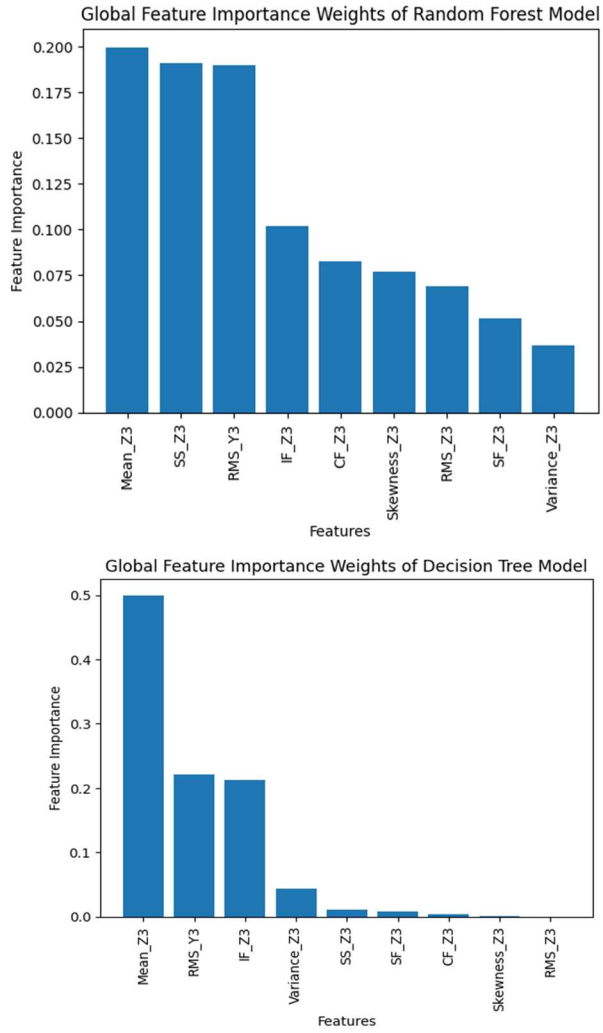



**FIGURE 15. Global Explanation by LIME showing feature importance for Random Forest and Decision Tree.**

### *C. LOCAL AND GLOBAL EXPLANATIONS USING SHAP*

Let us analyse the prediction locally and globally using a SHAP interpreter. Fig. 16 explains the 301$^{st}$ instance predicted by the RF model. It is predicted that the output is class 3 (Misalignment). The results of the SHAP explainer include the Shapley values for each feature in the model. Positive Shapley values indicate that a feature positively contributes to the

prediction, while negative values indicate a negative contribution. The magnitude of the Shapley value represents the strength of the contribution. This is called a force plot of SHAP.


**FIGURE 16. Local Explanation by SHAP for 301st instance of RF model.**

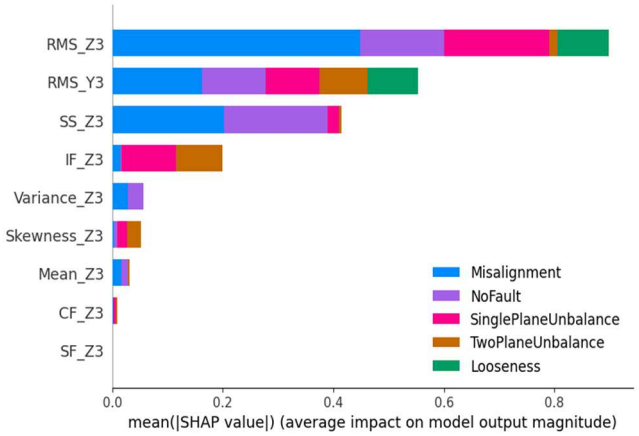



**FIGURE 17. SHAP summary plot (Local) showing the contribution of each feature towards the prediction for the 301st instance of the RF model**

Additionally, the SHAP explainer can produce summary plots such as bar charts or scatter plots to visualise the impact of features on individual predictions or on the overall model behaviour. Fig. 17 shows the SHAP summary plot for the 301st instance. This plot shows the Shapley values for each feature, allowing us to understand each feature's relative importance and direction of influence. This was locally explaining an instance using SHAP. Now, let us consider the Global Explanation for all the instances from the feature set using SHAP. Fig. 18 is a screenshot of the interactive Global Explanation given by SHAP (Force plot). As the cursor moves from left to right, the output changes corresponding to each instance. Currently, the cursor is placed on the 113th instance (highlighted dark black), and the output predicted by the RF model is 1 (No fault). Fig. 19 shows the global SHAP summary plot for all the instances. Fig. 20 is a dependency plot in the context of SHAP. It is a visual representation that helps to understand the relationship between a feature and its corresponding SHAP values. The x-axis represents the feature values, and the y-axis represents the SHAP values. Each data point on the plot represents an instance from the dataset. By analysing a dependency plot, we can identify different patterns and understand the impact of a specific feature on the model's predictions. Fig. 21 is a SHAP waterfall plot which is a visual representation of the contributions of different features to the prediction made by a machine learning model. The plot is structured like a waterfall, with each step representing the contribution of a specific feature. The baseline prediction is at the top, and the final prediction for the instance is at the bottom. The plot shows the cumulative effect of adding or subtracting the contributions of each feature.

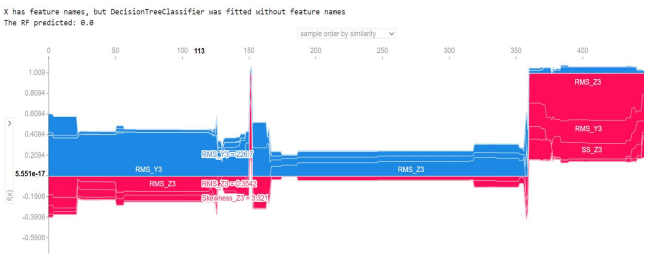



**FIGURE 18. Global Explanation by SHAP for all the instances of the RF model.**

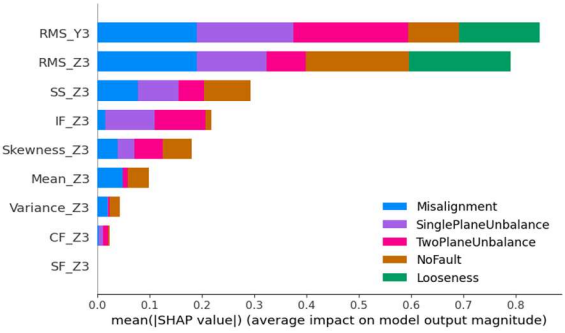



**FIGURE 19. Global Explanation by SHAP showing the summary plot of all the instances of the RF model.**

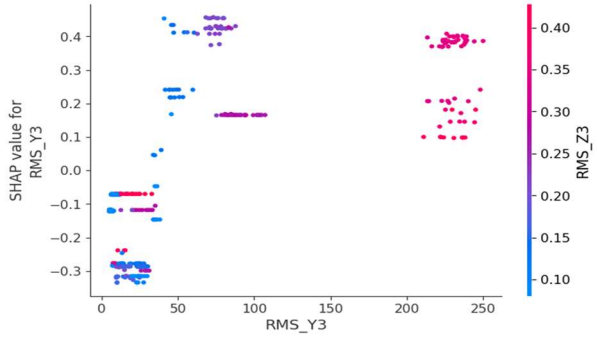



**FIGURE 20. Global Explanation by SHAP showing Dependence plot.**

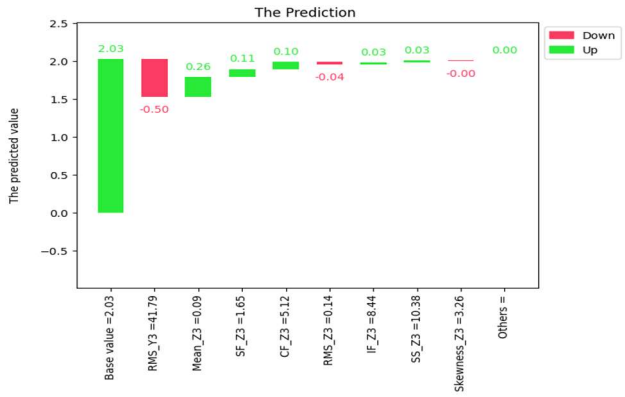



**FIGURE 21. SHAP waterfall plot**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3367110



### D. EXPLANATIONS USING ICE AND PDP PLOT

Let us analyse the ICE and PDP pots. Fig. 22 gives the ICE plot of feature IF_Z3 for multiple classes. The underlying prediction model is a Random Forest. There is an ICE curve for each class in a multi-class classification problem. The feature is plotted on the x-axis of the ICE plot, and the y-axis represents the probability of a particular class being assigned to a data point as you vary a specific feature while keeping all other features constant. Each ICE curve shows how the predicted probability of that class changes as the feature of interest varies. The ICE curves are typically centred around a base rate. This base rate represents the predicted probability for the class when all other features are held constant at their average or some other reference value. If the ICE curve for a class is mostly increasing as the feature value increases, it indicates that higher values of the feature are associated with a higher probability of that class. If the ICE curve is mostly decreasing, it suggests that higher feature values are associated with a lower probability of that class. A flat ICE curve suggests that the feature does not strongly influence the predicted probability for that class. Crossing ICE curves indicates that the relative importance of the analysed feature changes along its range. It suggests that the feature has different effects on class probabilities for different classes at different points. In other words, at those specific feature values, the predicted probability for one class becomes higher than the predicted probability for another class.

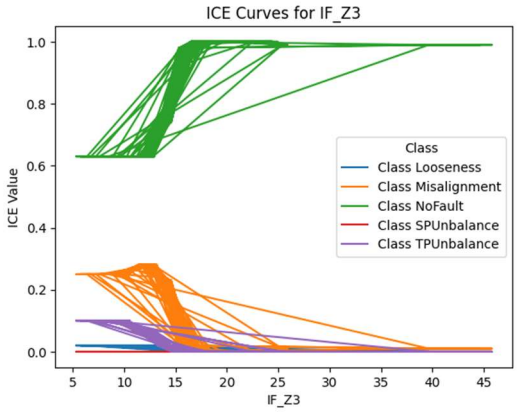



**FIGURE 22. ICE plot using Random Forest Model of Feature IF_Z3 for all the Classes.**

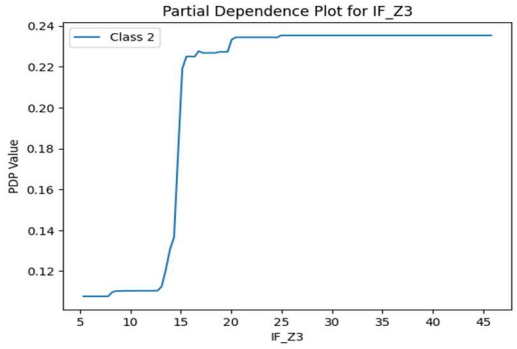



**FIGURE 23. PDP plot using Random Forest Model of Feature IF_Z3 for Class 2.**

Fig. 23 gives the PDP of the IF_Z3 feature for Class 2. The x-axis represents the range of values for the feature being analysed. The y-axis represents the effect of the feature on the model's prediction. The horizontal line (usually at y=0) on the PDP plot represents the base rate. The curve shows how predictions deviate from this base rate as the feature changes. If the PDP curve slopes upward as you move along the x-axis, it indicates that increasing the feature value leads to higher model predictions. Conversely, if the curve slopes downward, it suggests that increasing the feature value leads to lower model predictions. A flat PDP curve means the feature has little to no impact on the model's predictions. PDPs help understand how the model arrives at its predictions and can guide feature engineering and model improvement.

## IV. DISCUSSION

The analysis of the results shows that Explanation interfaces such as LIME, SHAP, PDP, and ICE can aid in the decision-making of maintenance activities for Rotating Machines. Following can be the possible inferences that can be taken from XAI models:

- For a specific rotating machine flagged for maintenance, LIME can highlight which sensor readings or features significantly influenced the prediction. Maintenance personnel can easily identify the specific indicators that led to the maintenance recommendation. For example, LIME might reveal that high RMS levels and increased temperature strongly influenced the prediction of a particular fault.
- SHAP might show that temperature readings have the highest overall impact on predicting faults in rotating machines, making temperature monitoring a top priority. This way, Explainable AI can aid in trust towards the black-box model's decisions.
- PDP and ALE can further add to analysing how a change in feature value leads to a change in prediction. They show how model predictions change as specific parameters change, making it easier to justify and understand diagnosis decisions.

When comparing LIME and SHAP to other explainable AI techniques, LIME and SHAP both provide local explanations, focusing on explaining individual predictions for specific data instances. They excel at providing insights into how individual features impact a particular prediction, making them highly interpretable on a case-by-case basis. Other explainable AI techniques, such as Feature Importance from Trees or even SHAP, offer global explanations. Secondly, both LIME and SHAP are model-agnostic and can be used with any black box model without requiring knowledge of the internal model architecture. This flexibility makes them widely applicable in various scenarios. Also, LIME uses simpler interpretable models, like linear models or decision trees, to approximate the black box model locally. This simplicity enhances the comprehensibility of explanations but may lead to some loss of fidelity compared to the original model. LIME might also

not capture complex feature interactions since it relies on simple models. SHAP is based on cooperative game theory, provides consistent feature attributions, and is more effective at capturing complex feature interactions. However, SHAP explanations might be more complex to understand due to the use of Shapley values and the inclusion of interaction effects between features.

PDPs are often used to interpret global models. In this context, PDPs provide insights into how changing one feature affects predictions across the entire dataset. ICE plots are particularly useful for interpreting local models. ICE plots visualise how the predictions of a local model change for individual data points within a specified region of feature space. However, it's important to note that PDPs and ICE plots can also be applied to global models for feature interpretation. However, PDPs assume that the features are independent of each other, which may not be the case in complex real-world datasets. When features are correlated or interact with each other, PDPs can provide incomplete or misleading insights. Hence, using them in conjunction with other interpretability techniques is essential.

Comparing the available literature on applications of XAI in predictive maintenance, several researchers have published related work as discussed in Table I. However, a very selective literature focusses on XAI in predictive maintenance of rotating machines. A study presented in [64] implements LIME on prediction of bearing faults. Study in [65], [49] and [3] presents the use of SHAP in explaining predictions made for bearing faults. Another study [66] proposes Grad-CAM method superior in comparison with LIME and SHAP in explaining the predictions made for gearbox faults in rotating machines. In most comparable studies, researchers predominantly concentrate on a singular fault type, such as bearings, rather than addressing multiple faults. Additionally, the majority of studies assess a single eXplainable Artificial Intelligence (XAI) technique, overlooking the potential benefits derived from leveraging a combination of multiple techniques to enhance decision-making processes. It is essential to note that due to the limited literature available on this specific topic, generalizations should be approached with caution.

### A. LIMITATIONS AND CHALLENGES IN EXPLAINABLE PREDICTIVE MAINTENANCE

The choice of the feature representation used may influence the interpretation. Therefore, it is crucial to interpret the results from LIME, SHAP, PDP, and ICE cautiously and consider them in association with domain knowledge and other interpretability techniques. Also, these techniques can only explain "why" a prediction was given. However, it is crucial to know "what next?" step to be taken to rectify the predicted output.

Considering these limitations, the following can be the future scope in the evolving Explainable Predictive Maintenance field.

### B. FUTURE DIRECTION IN EXPLAINABLE PREDICTIVE MAINTENANCE

Researchers can explore and develop new techniques that provide even more comprehensive and accurate explanations for predictive maintenance models. This could involve improving existing methods like LIME, SHAP, PDP, and ICE or exploring alternative explainability approaches. The goal would be to achieve a deeper understanding of the complex relationships between features and predictions in the context of maintenance.

Counterfactual Explanation is another aspect of Explainability that, if used with a set of right feature sets, can revolutionise the field of Predictive Maintenance concerning possible fault rectification factors.

## V. CONCLUSION

Predictive maintenance is an important application of machine learning in various industries, aiming to optimise maintenance operations and reduce equipment downtime. LIME, SHAP, PDP and ICE are popular techniques used to interpret the predictions of machine learning models. They provide insights into the features' contributions and their impact on the model's predictions, helping us understand the reasoning behind the maintenance recommendations made by the model. While these techniques serve a similar purpose, they differ in their approach and the types of explanations they offer. LIME generates explanations for individual predictions by locally approximating the model's behaviour. LIME explanations can provide valuable insights into how specific features influence the prediction for a given instance. On the other hand, SHAP offers a more holistic view of feature importance by considering all possible feature combinations. SHAP explanations provide a comprehensive understanding of the relative importance and direction of influence for each feature in the model. Finally, PDP and ICE aid in analysing "the point" where the feature value changes the prediction.

LIME, SHAP, PDP and ICE can help identify the critical features that contribute significantly to the maintenance recommendations in Predictive maintenance. By understanding the importance of different features, maintenance teams can prioritise their actions accordingly, focusing on the most influential factors affecting equipment failure or performance degradation. However, due to the novelty of this XAI field, there are some limitations and challenges related to Explainable Predictive Maintenance.

In summary, the future of explainable predictive maintenance lies in advancing the sophistication and usability of explanation techniques. These advancements will enable organisations to make informed maintenance decisions, improve equipment reliability, and optimise resource allocation based on a deep understanding of the underlying machine learning models.

## ACKNOWLEDGMENT

The authors extend their appreciation to King Saud University for funding this research through Researchers Supporting

Project Number (RSPD2023R890), King Saud University, Riyadh, Saudi Arabia.

## REFERENCES

[1] Z. M. Çinar, A. A. Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustain.*, vol. 12, no. 19, 2020, doi: 10.3390/su12198211.

[2] S. S. Gawde and S. Borkar, "Condition monitoring using image processing," *Proc. Int. Conf. Comput. Methodol. Commun. ICCMC 2017*, vol. 2018-Janua, pp. 1083–1086, 2018, doi: 10.1109/ICCMC.2017.8282638.

[3] M. J. Hasan, M. Sohaib, and J. M. Kim, "An explainable ai-based fault diagnosis model for bearings," *Sensors*, vol. 21, no. 12, 2021, doi: 10.3390/s21124070.

[4] S. Gawde, S. Patil, S. Kumar, P. Kamat, K. Kotecha, and A. Abraham, "Multi-Fault Diagnosis Of Industrial Rotating Machines Using Data-Driven Approach : A Review Of Two Decades Of Research," *Eng. Appl. Artif. Intell.*, vol. 123, no. 9 March 2023, pp. 1–39, 2023, doi: 10.1016/j.engappai.2023.106139.

[5] A. L. Dias, A. C. Turcato, G. S. Sestito, D. Brandao, and R. Nicoletti, "A cloud-based condition monitoring system for fault detection in rotating machines using PROFINET process data," *Comput. Ind.*, vol. 126, p. 103394, 2021, doi: https://doi.org/10.1016/j.compind.2021.103394.

[6] W. Mao, W. Feng, Y. Liu, D. Zhang, and X. Liang, "A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis," *Mech. Syst. Signal Process.*, vol. 150, p. 107233, 2021, doi: 10.1016/j.ymssp.2020.107233.

[7] S. Kumar *et al.*, "A Low-Cost Multi-Sensor Data Acquisition System for Fault Detection in Fused Deposition Modelling," *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020517.

[8] G. Betta, C. Liguori, A. Paolillo, and A. Pietrosanto, "A DSP-based FFT-analyzer for the fault diagnosis of rotating machine based on vibration analysis," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 6, pp. 1316–1321, 2002, doi: 10.1109/TIM.2002.807987.

[9] A. Yunusa-Kaltungo and R. Cao, "Towards developing an automated faults characterisation framework for rotating machines. Part 1: Rotor-related faults," *Energies*, vol. 16, no. 3, 2020, doi: 10.3390/en13061394.

[10] R. Tiwari and V. Analysis, "EFFECT OF FREQUENCY RESOLUTION ON INTELLIGENT MULTIPLE FAULT DIAGNOSIS OF INDUCTION MOTOR BASED ON SUPPORT VECTOR MACHINE," no. December, 2021.

[11] S. S. Patil, "Vibration Analysis of Electrical Rotating Machines using FFT A method of predictive maintenance," 2013.

[12] X. Yan and M. Jia, "A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing," *Neurocomputing*, vol. 313, pp. 47–64, 2018, doi: 10.1016/j.neucom.2018.05.002.

[13] S. Zhou, S. Qian, W. Chang, Y. Xiao, and Y. Cheng, "A novel bearing multi-fault diagnosis approach based on weighted permutation entropy and an improved SVM ensemble classifier," *Sensors (Switzerland)*, vol. 18, no. 6, 2018, doi: 10.3390/s18061934.

[14] Z. Liu, H. Cao, X. Chen, Z. He, and Z. Shen, "Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings," *Neurocomputing*, vol. 99, pp. 399–410, 2013, doi: 10.1016/j.neucom.2012.07.019.

[15] W. Gong *et al.*, "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion," *Sensors (Switzerland)*, vol. 19, no. 7, 2019, doi: 10.3390/s19071693.

[16] R. V. Sánchez, P. Lucero, R. E. Vásquez, M. Cerrada, J. C. Macancela, and D. Cabrera, "Feature ranking for multi-fault diagnosis of rotating machinery by using random forest and KNN," *J. Intell. Fuzzy Syst.*, vol. 34, no. 6, pp. 3463–3473, 2018, doi: 10.3233/JIFS-169526.

[17] T.-C. T. Chen, "Applications of XAI for Forecasting in the Manufacturing Domain," in *Explainable Artificial Intelligence (XAI) in Manufacturing: Methodology, Tools, and Applications*, Cham: Springer International Publishing, 2023, pp. 13–50. doi: 10.1007/978-3-031-27961-4_2.

[18] O. Janssens *et al.*, "Convolutional Neural Network Based Fault Detection for Rotating Machinery," *J. Sound Vib.*, vol. 377, pp. 331–345, 2016, doi: 10.1016/j.jsv.2016.05.027.

[19] M. Nacchia, F. Fruggiero, A. Lambiase, and K. Bruton, "A systematic mapping of the advancing use of machine learning techniques for predictive maintenance in the manufacturing sector," *Appl. Sci.*, vol. 11, no. 6, pp. 1–34, 2021, doi: 10.3390/app11062546.

[20] P. Baraldi, F. Cannarile, F. Di Maio, and E. Zio, "Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions," *Eng. Appl. Artif. Intell.*, vol. 56, pp. 1–13, 2016, doi: 10.1016/j.engappai.2016.08.011.

[21] M. S. Safizadeh and S. K. Latifi, "Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell," *Inf. Fusion*, vol. 18, no. 1, pp. 1–8, 2014, doi: 10.1016/j.inffus.2013.10.002.

[22] J. Tao, Y. Liu, and D. Yang, "Bearing Fault Diagnosis Based on Deep Belief Network and Multisensor Information Fusion," *Shock Vib.*, vol. 2016, 2016, doi: 10.1155/2016/9306205.

[23] A. Iqbal *et al.*, "Performance analysis of machine learning techniques on software defect prediction using NASA datasets," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 300–308, 2019, doi: 10.14569/ijacsa.2019.0100538.

[24] P. Ong, Y. K. Tan, K. H. Lai, and C. K. Sia, "A deep convolutional neural network for vibration-based health-monitoring of rotating machinery," *Decis. Anal. J.*, vol. 7, p. 100219, 2023, doi: https://doi.org/10.1016/j.dajour.2023.100219.

[25] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6634811.

[26] K. C. Luwei, A. Yunusa-Kaltungo, and Y. A. Sha'aban, "Integrated fault detection framework for classifying rotating machine faults using frequency domain data fusion and Artificial Neural Networks," *Machines*, vol. 6, no. 4, pp. 1–16, 2018, doi: 10.3390/MACHINES6040059.

[27] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vol. 72–73, pp. 303–315, 2016, doi: 10.1016/j.ymssp.2015.10.025.

[28] M. Huang and Z. Liu, "Research on mechanical fault prediction method based on multifeature fusion of vibration sensing data," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010006.

[29] J. Phillips, E. Cripps, J. W. Lau, and M. R. Hodkiewicz, "Classifying machinery condition using oil samples and binary logistic regression," *Mech. Syst. Signal Process.*, vol. 60–61, pp. 316–325, 2015, doi: https://doi.org/10.1016/j.ymssp.2014.12.020.

[30] M. N. Utah and J. C. Jung, "Fault state detection and remaining useful life prediction in AC powered solenoid operated valves based on traditional machine learning and deep neural networks," *Nucl. Eng. Technol.*, vol. 52, no. 9, pp. 1998–2008, 2020, doi: https://doi.org/10.1016/j.net.2020.02.001.

[31] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning," *IEEE Access*, vol. 7, no. c, pp. 19990–19999, 2019, doi: 10.1109/ACCESS.2018.2890566.

[32] M. Sohaib, C. H. Kim, and J. M. Kim, "A hybrid feature model and deep-learning-based bearing fault diagnosis," *Sensors (Switzerland)*, vol. 17, no. 12, 2017, doi: 10.3390/s17122876.

[33] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, no. January, pp. 213–237, 2019, doi: 10.1016/j.ymssp.2018.05.050.

[34] H. Wu, A. Huang, and J. Sutherland, "Layer-wise relevance propagation for interpreting LSTM-RNN decisions in predictive maintenance," *Int. J. Adv. Manuf. Technol.*, vol. 118, pp. 1–16, 2022, doi: 10.1007/s00170-021-07911-9.

[35] J. He, X. Li, Y. Chen, D. Chen, J. Guo, and Y. Zhou, "Deep Transfer Learning Method Based on 1D-CNN for Bearing Fault Diagnosis," vol. 2021, no. 1, 2021.

[36] J. He, X. Li, Y. Chen, D. Chen, J. Guo, and Y. Zhou, "Deep Transfer Learning Method Based on 1D-CNN for Bearing Fault Diagnosis," *Shock Vib.*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/6687331.

[37] J. He, X. Li, Y. Chen, D. Chen, J. Guo, and Y. Zhou, "Deep Transfer Learning Method Based on 1D-CNN for Bearing Fault Diagnosis," *Shock Vib.*, vol. 2021, pp. 1–16, 2021, doi: 10.1155/2021/6687331.

[38] J. Wang, D. Wang, S. Wang, W. Li, and K. Song, "Fault Diagnosis of Bearings Based on Multi-Sensor Information Fusion and 2D Convolutional Neural Network," *IEEE Access*, vol. 9, pp. 23717–23725, 2021, doi: 10.1109/ACCESS.2021.3056767.

[39] V. B. N. S. S. U. Krishnaraj Chadaga Srikanth Prabhu and R. Chadaga, "Artificial intelligence for diagnosis of mild–moderate COVID-19 using haematological markers," *Ann. Med.*, vol. 55, no. 1, p. 2233541, 2023, doi: 10.1080/07853890.2023.2233541.

[40] S. Hao, F. X. Ge, Y. Li, and J. Jiang, "Multisensor bearing fault diagnosis based on one-dimensional convolutional long short-term memory networks," *Meas. J. Int. Meas. Confed.*, vol. 159, p. 107802, 2020, doi: 10.1016/j.measurement.2020.107802.

[41] D. Pagano, "A predictive maintenance model using Long Short-Term Memory Neural Networks and Bayesian inference," *Decis. Anal. J.*, vol. 6, p. 100174, 2023, doi: https://doi.org/10.1016/j.dajour.2023.100174.

[42] X. Xu, Z. Tao, W. Ming, Q. An, and M. Chen, "Intelligent monitoring and diagnostics using a novel integrated model based on deep learning and multi-sensor feature fusion," *Meas. J. Int. Meas. Confed.*, vol. 165, p. 108086, 2020, doi: 10.1016/j.measurement.2020.108086.

[43] Z. H. Liu, B. L. Lu, H. L. Wei, L. Chen, X. Li, and C. T. Wang, "A Stacked Auto-Encoder Based Partial Adversarial Domain Adaptation Model for Intelligent Fault Diagnosis of Rotating Machines," *IEEE Trans. Ind. Informatics*, no. February 2021, 2020, doi: 10.1109/TII.2020.3045002.

[44] P. Shi, X. Guo, D. Han, and R. Fu, "A sparse auto-encoder method based on compressed sensing and wavelet packet energy entropy for rolling bearing intelligent fault diagnosis," *J. Mech. Sci. Technol.*, vol. 34, no. 4, pp. 1445–1458, 2020, doi: 10.1007/s12206-020-0306-1.

[45] J. Lu, H. Zhang, and X. Tang, "A Novel Method for Intelligent Single Fault Detection of Bearings Using SAE and Improved D–S Evidence Theory," *Entropy*, vol. 21, no. 7, p. 687, 2019, doi: 10.3390/e21070687.

[46] S. Gawde, S. Patil, S. Kumar, and K. Kotecha, *A scoping review on multi-fault diagnosis of industrial rotating machines using multi-sensor data fusion*, no. 0123456789. Springer Netherlands, 2022. doi: 10.1007/s10462-022-10243-z.

[47] A. Ferraro, A. Galli, V. Moscato, and G. Sperlì, "Evaluating eXplainable artificial intelligence tools for hard disk drive predictive maintenance," *Artif. Intell. Rev.*, vol. 56, 2022, doi: 10.1007/s10462-022-10354-7.

[48] D. Solís-Martín, J. Galán-Páez, and J. Borrego-Díaz, "On the Soundness of XAI in Prognostics and Health Management (PHM)," *Inf.*, vol. 14, no. 5, pp. 1–24, 2023, doi: 10.3390/info14050256.

[49] T. Mansouri and S. Vadera, "Explainable fault prediction using learning fuzzy cognitive maps," *Expert Syst.*, no. January, pp. 1–17, 2023, doi: 10.1111/exsy.13316.

[50] T. H. Huang, A. Metzger, and K. Pohl, "Counterfactual Explanations for Predictive Business Process Monitoring," *Lect. Notes Bus. Inf. Process.*, vol. 437 LNBIP, no. January, pp. 399–413, 2022, doi: 10.1007/978-3-030-95947-0_28.

[51] N. Nordin, Z. Zainol, M. H. Mohd Noor, and L. F. Chan, "An Explainable Predictive Model for Suicide Attempt Risk using An Ensemble Learning and Shapley Additive Explanations (SHAP) Approach," *Asian J. Psychiatr.*, vol. 79, p. 103316, 2022, doi: 10.1016/j.ajp.2022.103316.

[52] W. Jmoona *et al.*, "Explaining the Unexplainable: Role of XAI for Flight Take-Off Time Delay Prediction," no. June, pp. 81–93, 2023, doi: 10.1007/978-3-031-34107-6_7.

[53] S. J. Upasane, H. Hagras, M. H. Anisi, S. Savill, I. Taylor, and K. Manousakis, "A Type-2 Fuzzy Based Explainable AI System for Predictive Maintenance within the Water Pumping Industry," *IEEE Trans. Artif. Intell.*, pp. 1–14, 2023, doi: 10.1109/TAI.2023.3279808.

[54] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and Counterfactual Explanations for Black-Box Decision Making," *IEEE Intell. Syst.*, vol. PP, no. c, p. 1, 2019, doi: 10.1109/MIS.2019.2957223.

[55] N. S. R. C. S. K. S. Krishnaraj Chadaga Srikanth Prabhu and S. Sengupta, "Predicting cervical cancer biopsy results using demographic and epidemiological parameters: a custom stacked ensemble machine learning approach," *Cogent Eng.*, vol. 9, no. 1, p. 2143040, 2022, doi: 10.1080/23311916.2022.2143040.

[56] V. V. Khanna *et al.*, "A decision support system for osteoporosis risk prediction using machine learning and explainable artificial intelligence," *Heliyon*, vol. 9, no. 12, Dec. 2023, doi: 10.1016/j.heliyon.2023.e22456.

[57] N. Jothi, W. Husain, and N. A. Rashid, "Predicting generalized anxiety disorder among women using Shapley value," *J. Infect. Public Health*, vol. 14, no. 1, pp. 103–108, 2021, doi: 10.1016/j.jiph.2020.02.042.

[58] S. Gawde, S. Patil, S. Kumar V C, P. Kamat, K. Kotecha, and A. Abraham, "Multi-fault diagnosis of Industrial Rotating Machines using Data-driven approach : A review of two decades of research," vol. Volume 123, 2023, doi: 10.1016/j.engappai.2023.106139.

[59] D. Gunning and D. Aha, "DARPA's Explainable Artificial Intelligence (XAI) Program," *AI Mag.*, vol. 40, no. 2, pp. 44–58, 2019, doi: 10.1609/aimag.v40i2.2850.

[60] F. Hussain, R. Hussain, and E. Hossain, "Explainable Artificial Intelligence (XAI): An Engineering Perspective," pp. 1–11, 2021, [Online]. Available: http://arxiv.org/abs/2101.03613

[61] C. Molnar, "Interpretable Machine Learning. A Guide for Making Black Box Models Explainable.," *Book*, p. 247, 2020, [Online]. Available: https://christophm.github.io/interpretable-ml-book

[62] A. Barredo Arrieta *et al.*, "Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, no. December 2019, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012.

[63] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A Review of Machine Learning Interpretability Methods," *Entropy*, vol. 23, no. 1, 2021, doi: 10.3390/e23010018.

[64] D. C. Sanakkayala *et al.*, "Explainable AI for Bearing Fault Prognosis Using Deep Learning Techniques," *Micromachines*, vol. 13, no. 9, 2022, doi: 10.3390/mi13091471.

[65] E. Brusa, L. Cibrario, C. Delprete, L. Gianpio, and D. Maggio, "Explainable AI for Machine Fault Diagnosis : Understanding Features ' Contribution in Machine Learning Models for Industrial Condition Monitoring," *Appl. Sci.*, 2023.

[66] D. Solís-Martín, J. Galán-Páez, and J. Borrego-Díaz, "On the Soundness of XAI in Prognostics and Health Management (PHM)," *Inf.*, vol. 14, no. 5, 2023, doi: 10.3390/info14050256.

**MS. SHREYAS GAWDE** has completed Masters in Engineering in Industrial Automation from Goa College of Engineering, Farmagudi and is currently pursuing PhD from Symbiosis International (Deemed University). She is also an Assistant Professor in the Department of Electronics at Goa University. Her research interest includes interdisciplinary areas such as Industry 4.0, Artificial Intelligence, Predictive Maintenance.

**DR. SHRUTI PATIL** has been an industry professional in the past, currently associated with Symbiosis Institute of Technology as an Associate professor and HOD at AI/ML Department. She has completed her M.Tech in Computer Science and Ph.D in the domain of Data Privacy from Pune University. She has 3 years of industry experience and 10+ years of academic experience. Her research areas include applied artificial intelligence, natural language processing, acoustic AI, adversarial machine learning, data privacy, digital twin applications, GANS, multimodal data analysis. She is currently working in the application domains of healthcare, sentiment analysis, emotion detection and machine simulation via which she is also guiding several UG, PG and PhD students as a domain expert. She has published research articles in reputed international conferences and Scopus/ web of science indexed journals, books. Dr Shruti has also been invited to deliver talks related with her expertise in various prestigious institutions.

**DR. SATISH KUMAR** is an Associate Professor in the Department of Robotics and Automation at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune India. He did his Master degree (M.Tech) and Doctoral degree (PhD) in 2013 and 2020 from K Visvesvaraya Technological University, Belgaum, Karnataka, India. He has 8+ years of experience in teaching, research and industries. He has his expertise in application of Artificial Intelligence into a variety of crucial manufacturing processes, including quality enhancement, process improvement, and optimisation. He is now working on several industrial 4.0 initiatives, including digital twin smart manufacturing, remaining usable life estimation, and others. He has authored more than 30 Scopus/WoS indexed international/national journal and conferences publications. He has recently published a patent on "Micro-oxidation Coating device". One student was recently awarded PhD. under his supervision and is currently guiding 5 Ph.D. research scholars.

**DR. POOJA KAMAT** is an Assistant Professor in the AI/ML department at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India. She has completed her Ph.D. in the domain of Applied AI for Industry 4.0. She has guided many undergraduate and postgraduate engineering students. Her expertise includes Machine Learning & Deep Learning, Cloud Computing, Predictive Analytics, and Agile Software Development. She has published more than 40+ research papers in top-indexed journals and conferences. She has been invited to conduct guest lectures and workshops for many Applied AI use cases.

**DR. KETAN KOTECHA** is a Professor at Computer Science & Engineering Department; Head at Symbiosis Centre for Applied Artificial Intelligence (SCAAI); Dean at Faculty of Engineering, Symbiosis International (Deemed University); Director at Symbiosis Institute of Technology and a Chief Executive Officer (CEO) at Symbiosis Centre for Entrepreneurship and Innovation. His 25 years of extraordinary career saw him serving in the finest of the engineering colleges in various higher technical education leadership positions. He has received various awards, guided a number of projects as well as PhD students. He has published widely in a number of excellent peer-reviewed journals on various topics ranging from education policies, teaching learning practices and AI for all. A researcher- teacher of Deep learning, his interest areas are Artificial Intelligence, Computer Algorithms, Machine Learning, Deep Learning Higher Order Thinking Skills, Critical Thinking and Ethics & Values.

**DR. SULTAN ALFARHOODA** is an Assistant Professor in the Department of Computer Science at King Saud University (KSU). Since joining KSU in 2007, he has made several contributions to the field of computer science through his research and publications. Dr. Alfarhood holds a PhD in Computer Science from the University of Arkansas and has published several research papers on cutting-edge topics such as Machine Learning, Recommender Systems, Linked Open Data, and Text Mining. His work includes proposing innovative approaches and techniques to enhance the accuracy and effectiveness of recommender systems and sentiment analysis.