IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 25 March 2024, accepted 14 April 2024, date of publication 18 April 2024, date of current version 29 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3391130

SURVEY

Explainable Predictive Maintenance: A Survey of Current Methods, Challenges and Opportunities

LOGAN CUMMINS[1], (Member, IEEE), ALEXANDER SOMMERS[1], (Member, IEEE), SOMAYEH BAKHTIARI RAMEZANI[1], (Member, IEEE), SUDIP MITTAL[1], (Member, IEEE), JOSEPH JABOUR[2], MARIA SEALE[2], AND SHAHRAM RAHIMI[1], (Member, IEEE)
[1]Department of Computer Science and Engineering, Mississippi State University, Starkville, MS 39762, USA
[2]U.S. Army Engineer Research and Development Center (ERDC), Vicksburg, MS 39180, USA

Corresponding author: Logan Cummins (nlc123@cavs.msstate.edu)

This work was supported by Mississippi State University funded by the U.S. Department of Defense (DoD) High Performance Computing Modernization Program, through the U.S. Army Engineering Research and Development Center (ERDC) under Grant W912HZ21C0014.

ABSTRACT Predictive maintenance is a well studied collection of techniques that aims to prolong the life of a mechanical system by using artificial intelligence and machine learning to predict the optimal time to perform maintenance. The methods allow maintainers of systems and hardware to reduce financial and time costs of upkeep. As these methods are adopted for more serious and potentially life-threatening applications, the human operators need trust the predictive system. This attracts the field of Explainable AI (XAI) to introduce explainability and interpretability into the predictive system. XAI brings methods to the field of predictive maintenance that can amplify trust in the users while maintaining well-performing systems. This survey on explainable predictive maintenance (XPM) discusses and presents the current methods of XAI as applied to predictive maintenance while following the *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (PRISMA) 2020 guidelines. We categorize the different XPM methods into groups that follow the XAI literature. Additionally, we include current challenges and a discussion on future research directions in XPM.

INDEX TERMS Explainable artificial intelligence (XAI), predictive maintenance, industry 4.0, industry 5.0, interpretable machine learning, PRISMA.

I. INTRODUCTION

The history of technological advancements within the past couple of hundred years is well documented. These centuries and decades can be categorized into what is described as revolutions, i.e. Industrial Revolutions [1]. The most recent of these is agreed to be known as the fourth industrial revolution or Industry 4.0 [1], [2], [3], [4].

Industry 4.0 is categorized by bridging the gap between machinery through hardware and software connectivity [5]. This revolution is characterized by the inclusion of human-machine interfaces, AI, and internet of things technologies [5]. Through these technologies, we can become more automated and efficient with new challenges that come with big data and cyber-physical systems. One of the problems created from this revolution has centered around the optimization of mechanical systems.

One method of optimizing mechanical systems is to minimize the downtime the system may suffer from due to break-downs and repairs. To tackle this level of optimization, researchers of Industry 4.0 have developed the field of predictive maintenance (PdM). PdM encompasses many different problems in the field of maintenance, but an overarching representation of PdM involves monitoring the system as it is in the present and alerting for any potential problems such as a specific anomaly or time until failure [1], [6]. While this problem that exists in the cyber-physical realm has been well studied from the perspective of deep learning models, statistical models, and more, the people that get impacted by these systems have had considerably

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

less attention. This change of focus leads us into the fifth industrial revolution or Industry 5.0.

While the mechanical systems were the focus of the fourth industrial revolution, human-centered challenges have become the focus of the fifth revolution. As described by Leng et al. [2], humans must be important in the processes related to these important decision-making systems. Nahavandi et al. [4] illustrates Industry 5.0 in the realm of a factory line. The human performs a task that is assisted by an artificial intelligent agent that can increase the productivity of the human.

As these systems are moving the focus away from mechanics and towards humans, a different area must be brought to the forefront. The way to address human-centered processes can be derived from the fields of eXplainable AI (XAI) and Interpretable Machine Learning (iML). XAI and iML are extensively researched from multiple fields on a wide array of problems including the various problems in PdM. Our article's main contribution involves using the *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* statement to organize the XAI and iML works applied to PdM. We also describe and categorize the different methods, note challenges found in PdM and provide key aspects to keep the field of Explainable Predictive Maintenance (XPM) moving forward.

The article is organized in the following manner. In Section II, important information surrounding explainability, Interpretable Machine Learning, and predictive maintenance are described. Section III describes the literature search performed including identification, screening, and inclusion. In Sections IV,V and VI, the results of the literature review are categorized and discussed in detail. Section VII discusses challenges in the field that remain to be addressed, and Section VIII provides our closing remarks.

II. BACKGROUND

To accommodate readers of varying backgrounds, we briefly explain a couple of key topics needed for understanding the importance of this research, namely Explainable Artificial Intelligence (XAI), Interpretable Machine Learning (iML), and Predictive Maintenance (PdM). We will also discuss the distinction between XAI and iML to inform the readers of the perspective with which we evaluated the literature.

A. EXPLAINABILITY AND INTERPRETABILITY IN ARTIFICIAL INTELLIGENCE

The fine distinction between *explainability* and *interpretability* in the context of AI and ML has raised considerable debate [7]. While several researchers argue that the terms are synonymous, viewing them as interchangeable to simplify discussions [8], [9], [10], [11], others assert that they capture distinct concepts [12], [13], [14], [15], [16], [17], [18], [19]. Interestingly, a third perspective points out that one term is a subset of the other, adding another layer to the discourse [20], [21], [22].

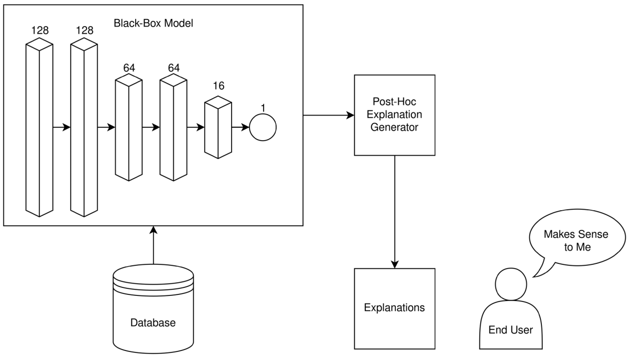


FIGURE 1. Visualization of XAI Design Cycle.

To ensure clarity and coherence in this article, we consider that *explainability* and *interpretability* are related yet distinct. While there exists a certain degree of overlap, they emphasize different facets of machine learning.

1) EXPLAINABLE ARTIFICIAL INTELLIGENCE

The rapidly growing field of eXplainable Artificial Intelligence (XAI) aims to demystify AI systems by clarifying their reasoning mechanisms and subsequent outputs [7]. XAI methodologies can typically be classified based on features such as the scope of explanation—whether global or local—and the techniques employed for generating explanations, like feature perturbation. A unifying theme across these methods is the endeavor to interpret the workings of an already-trained model. As Sokol et al. succinctly put it, *explainability is for the model's output* [19]. From a more analytical standpoint, XAI predominantly encompasses post-hoc strategies to shed light on otherwise opaque, black-box models [16]. This paradigm is illustrated in Figure 1, where a model's explanations are constructed to enhance user comprehension.

a: MODEL-AGNOSTIC AND MODEL-SPECIFIC

Explainable methods can be categorized based on their suitability for addressing various types of black-box models. Methods that are applicable to models regardless of their architecture are called *model-agnostic*. Common methods that fall into this category are Shapley Additive Explanations (SHAP) [23] and Local Interpretable Model-agnostic Explanations (LIME) [24]. These methods and additional model-agnostic methods are described in Section V-A. The opposite of these methods are known as *model-specific*. Model-specific methods such as Class Activation Mapping (CAM) [25] for Convolutional Neural Networks (CNNs) are designed to take advantage of the architecture already to provide explainability. These methods and others are described in Section V-B.

b: LOCAL EXPLANATIONS AND GLOBAL EXPLANATIONS

Another way of classifying explainable methods is by the scope of the explaination. These scopes are commonly described as either *local* or *global*. Local explanations aim

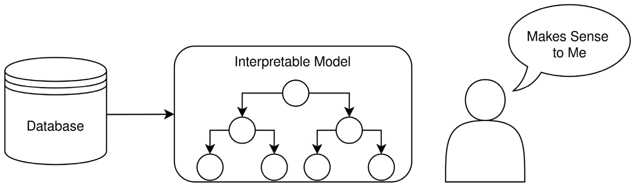


FIGURE 2. Visualization of interpretable ML Design Cycle.

at explaining the model's behavior for a single data point. Global explanations provide reasoning that represents the model's behavior for any data point.

c: XAI EXAMPLE

To give a concrete example of XAI, a researcher may want to use a Long Short-Term Memory neural network for time-series analysis due to its temporal modeling capabilities [1], [6]. Common deep learning models like this one are not commonly interpretable, so to make it explainable, the researcher might consider using a simpler model, i.e., linear regression, decision tree, etc., to serve as a surrogate for post-hoc explanations. These explanations would then be presented to the user/developer/stakeholder to better explain the behavior of an inherent black-box architecture.

2) INTERPRETABLE MACHINE LEARNING

Interpretable Machine Learning (iML) describes ML models that are referred to as *white-* or *gray-boxes* [12], and their interpretability is enforced by architectural or functional constraints. Between the two, architectural constraints make models simple enough to understand, while physical constraints attempt to cast the model's computations in terms of real-world features. While XAI focuses on the model's output, iML focuses on the model itself [19]. This has also been stated as *intrinsic interpretability* as to separate it from *post-hoc explainability* methods [22], [26]. As follows, this article will equate iML with models that are intrinsically interpretable through methods of structural constraints, physical bindings, etc. This can be seen in Figure 2, where there is no need for translating the model through an explainable method.

For a concrete example, a researcher may have a problem that could benefit from a simple logistic regression classifier. With such a simple architecture, the network itself would be interpretable as it would be clear what inputs affect what outputs. One could also extrapolate the overarching equation if the network is simple enough. This illustrates inherent interpretability.

B. PREDICTIVE MAINTENANCE

Predictive maintenance (PdM) is a subcategory of prognostics and health management (PHM) that has seen widespread attention in recent years [1], [22], [27], [28]. PdM utilizes AI and previous failure information from mechanical systems to predict a fault or downtime in the future [1], [6], [29]. PdM is implemented with a variety of tools, including anomaly detection, fault diagnosis and prognosis [22], [28].

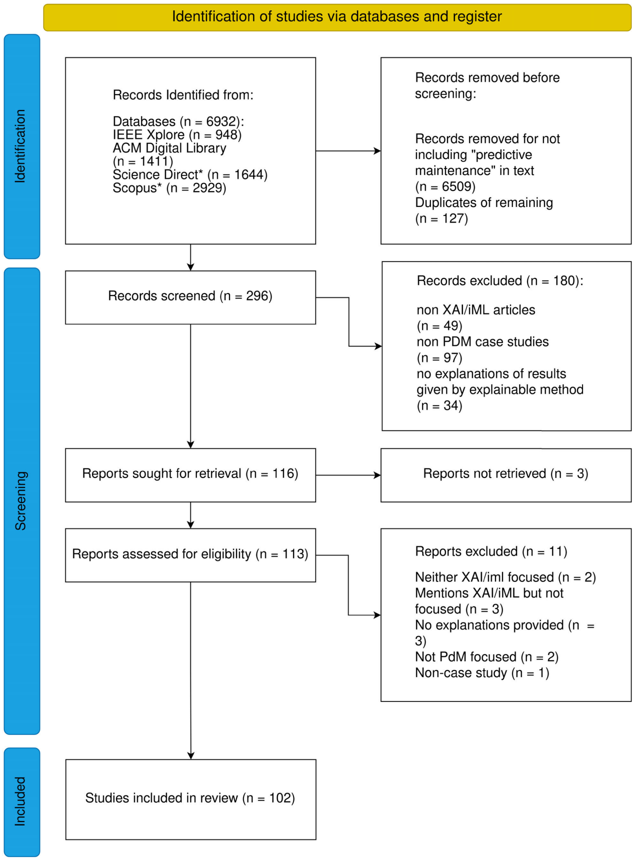

Identification of studies via databases and register
Identification
Records Identified from:
Databases (n = 6932):
IEEE Xplore (n = 948)
ACM Digital Library (n = 1411)
Science Direct* (n = 1644)
Scopus* (n = 2929)
Records removed before screening:
Records removed for not including "predictive maintenance" in text (n = 6509)
Duplicates of remaining (n = 127)
Screening
Records screened (n = 296)
Records excluded (n = 180):
non XAI/iML articles (n = 49)
non PDM case studies (n = 97)
no explanations of results given by explainable method (n = 34)
Reports sought for retrieval (n = 116)
Reports not retrieved (n = 3)
Reports assessed for eligibility (n = 113)
Reports excluded (n = 11)
Neither XAI/iml focused (n = 2)
Mentions XAI/iML but not focused (n = 3)
No explanations provided (n = 3)
Not PdM focused (n = 2)
Non-case study (n = 1)
Included
Studies included in review (n = 102)

FIGURE 3. PRISMA Search.

Anomaly detection and fault diagnosis have a very distinct difference. Whereas anomaly detection aims at determining whether a fault occured or not, fault diagnosis aims to identify the cause of a fault [28], [30]. This means that anomaly detection can be thought of as a binary classification problem, and fault diagnosis can be thought of as an extension of anomaly detection to a multi-classification problem. Finally, prognosis deals with predicting the remaining useful life (RUL) or time until failure [1], [6], [28]. This puts prognosis in the domain of regression problems. Now that these terms are defined and categorized into their different problems, we can discuss the PRISMA compliant systematic search that we performed.

III. SYSTEMATIC SEARCH

We utilized the *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (PRISMA) 2020 statement [31], [32] to layout a systemized methodology of performing a literature review. The full process can be seen in Fig. 3.

A. IDENTIFICATION

In identifying the potential databases, we focused on popular computer science publishers as well as general scientific publishers. We utilized the following databases for literature searches: IEEE Xplore, ACM Digital Library, ScienceDirect and Scopus, all of which were accessed on June 21, 2023. To capture as much as we could, we searched titles, keywords, and abstracts with two ideas in mind: *XAI and iML* and *PdM*.

In the former case we used *explainable OR interpretable OR xai* to capture the first grouping of papers. This should gather papers with common phrases like *explainable artificial intelligence, explainable machine learning, interpretable ML, XAI, etc.* To capture the PdM aspect, we provided more explicit words so as to represent the research area better. We used *prognos* OR diagnos* OR RUL OR remaining useful life OR predictive maintenance OR detection.* This would capture ideas such as *prognosis, prognostics, diagnosis, diagnostics, detection, etc.*

In research, words like *prognosis* and *diagnosis* appear in medically related articles. This makes sense as many can attest that they would go to their physician for a diagnosis. To minimize the inclusion of medical literature, ScienceDirect and Scopus were set to look at Engineering and Computer Science related articles only. Even with this selection, the initial pool of research was 6932 articles.

This narrowing down of papers was not as effective as we initially expected as only the titles, keywords, and abstracts were checked. Prior to removing duplicates, we also removed articles that did not mention *predictive maintenance* inside of the article. After removing those papers and duplicates, the initial screening started with 296 articles.

B. EXCLUSION CRITERIA AND SCREENING

Our initial screenings involved skimming through the abstracts, main objectives, conclusions, and images of the articles. These initial screenings utilized the following exclusion criteria:

1) Neither XAI nor iML are a main focus of the article.
2) Articles are not PdM case studies.
3) No explanation or interpretation is provided.

The need for the first tow criteria is easily apparent. Many articles would mention one of the search terms from XAI/iML, but they would not fall into this category of work (n = 49). This would mainly emerge as using the words *explainable* or *interpretable* in a sentence of the abstract. Similarly, to the PdM case studies, many articles mention *diagnosis*, and such, in a sentence without it being the focus of the article (n = 97). However, the third criterion needs a more in-depth explanation.

When stating that an architecture is interpretable or explainable, a certain expectation is implanted in the reader's mind. This applies to any concept whether it be computer science related or not. One of the expectations that we agreed upon was providing proof of interpretability or explainability. This would necessitate the explanation from the explainable method or the inherent interpretation of the interpretable model. With this expectation in mind, a few articles (n = 34) were removed before in-depth screening due to a mention of an explanatory method without any output of the said method. This finalized a screening population of 116 articles which were sought for retrieval. Three were not retrieved by our resources. Upon further examination, those three articles seem to lead to dead URLs.

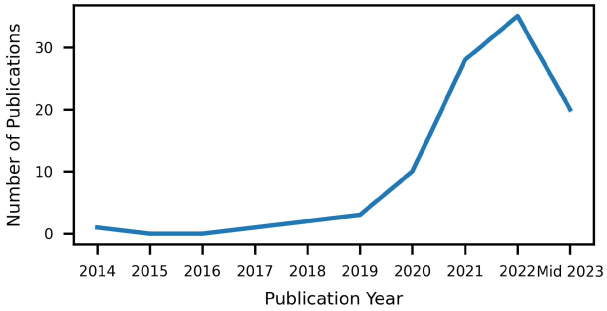


FIGURE 4. Articles published per year in our inclusion results.

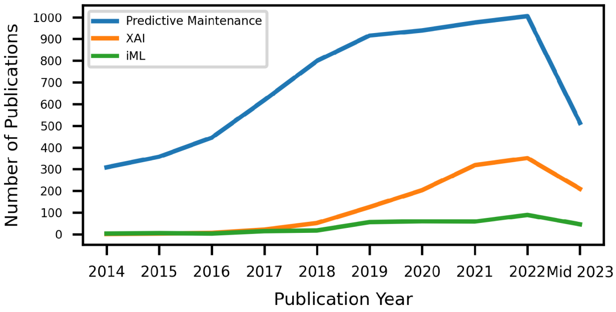


FIGURE 5. Google Search Trend for PdM, XAI, and iML from our article years.

For final assessment of eligibility, all of the resources were read. Many of the articles that were excluded were not available outside of a small preview. Of the remaining 113 articles, 11 were excluded for the following reasons:

- Three mention XAI/iML in the abstract but do not utilize any methods that we could find.
- Two were neither XAI nor iML. These mention search terms in the abstracts, but do not build on them.
- Three offer no interpretations of their interpretive method.
- Two mention PdM in the abstract but do not focus on PdM in an experiment.
- One was not a case study.

C. INCLUSION

After careful review of the articles, we finalized a population of 102 articles. Our findings and these articles are now discussed in Section IV.

IV. SEARCH RESULTS

To paint an overarching picture of our results, Fig. 4 shows a break-down of our inclusion population grouped by year. This shows a clear increasing trajectory in publications that can be explained by a few potential factors. Firstly, the popularity of predictive maintenance continues to increase, as shown in [1] and in Fig. 5, as we move to a big-data centric world in industry. This provides more opportunities

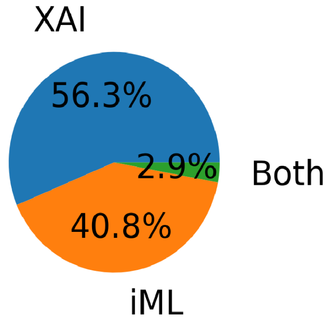


FIGURE 6. Distribution of XAI and iML in the search results.

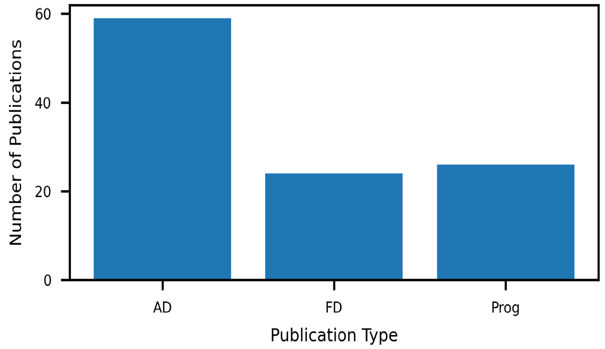


FIGURE 7. Papers per Anomaly Detection (AD), Fault Diagnosis (FD), and Prognosis (Prog).

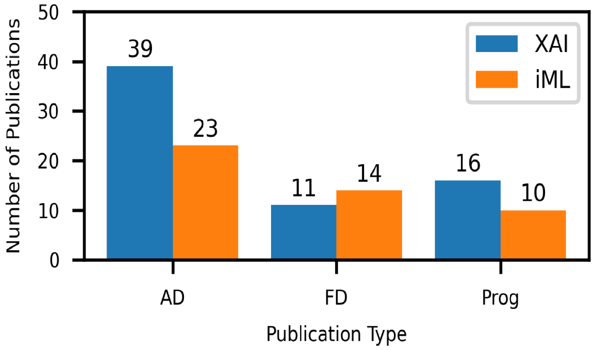


FIGURE 8. Split between XAI and iML per category of predictive maintenance.

to implement these very large and very complex neural architectures for making important decisions. The importance of these decisions leads to a second reason for increasing importance, trust.

Many articles discuss the importance of increasing the *trust* of the users in the model while decreasing the *bias* in black-box models [33], [34], [35], [36]. Rojat et al. define trust as achieved once a model can effectively explain its decisions to a person [18]. This would necessitate some sort of explainable or inherently interpretable architecture that could give the users insight. Furthermore, Vollert et al. [22] even state that trust is a *prerequisite* for a successful data-driven application.

Looking at Fig. 6, our findings reflect the idea that XAI is slightly more popular than iML in PdM. One potential reason could be the desire to make use of the benefits from complex models. Many of the articles utilize architectures such as Deep Convolutional Neural Networks [37] or Long Short-term Memory Neural Networks [38] due to their high performance in the application. With the inherent black-box nature of these models, these researchers need post-hoc explainable methods. This desire for XAI over iML seems to be affecting specific PdM tasks more than others.

The articles are categorized according to PdM task in Fig. 7, and those are further distinguished into XAI and iML within tasks in Fig. 8. Our article population reflects *anomaly detection* as the main task that utilizes XAI and iML. Fault diagnosis and prognosis are virtually the same in number of articles published within this population; however, Fig. 8 shows that the interest in XAI and iML are reversed in these groups. Succinctly, prognosis focuses on XAI, while diagnosis focuses on iML. We now describe the many methods that were applied to the varying datasets seen in Table 1. These methods are split between section V for XAI methods and section VI for iML methods. Additionally, specific articles of interest can be found in Table 4.

V. EXPLAINABLE AI IN PREDICTIVE MAINTENANCE

XAI in predictive maintenance captures a wide range of methods that can be categorized in several ways. To not repeat information, the methods are broken up into three sub-sections: model-agnostic, model-specific, and combination. The taxonomy that distinguishes the model-agnostic methods from the model-specific methods can be seen in Fig. 9.

A. MODEL-AGNOSTIC

This section describes the explainable methods in our population, seen in Table 2, that could be applied to any architecture. These methods are colloquially known as *model-agnostic* explainable methods [149]. These methods found in this section can be applied to any architecture and consist of SHAP in Section V-A1, LIME in Section V-A2 and additional related methods.

1) SHAPLEY ADDITIVE EXPLANATIONS (SHAP).

SHAP values were introduced by Lundberge et al. as a unified measure of feature importance [23]. SHAP is based on three properties that are shared with classical Shapley value estimation: local accuracy, missingness, and consistency. Local accuracy refers to the ability of the simplified input to *at least* match the output of the input from the data. Missingness refers to the features that are missing from the simplified input. Succinctly, this states that if a feature is not useful to the explanation, then it is not useful to the model. Finally, consistency brings the idea that the importance of a feature should stay the same or increase regardless of the other features.

By far, SHAP is the most used method seen in our sample. Moreover, SHAP is one of the few methods that has been

TABLE 1. Datasets from the literature search.

Datasets	Most Common Methods	Articles
Bearings and PRONOSTIA [39], [40]	Local Interpretable Model-agnostic Explanations, Interpretable Filters and Fuzzy Knowledge	[33], [37], [41]–[55]
Vehicles or vehicle subsystem	Attention and Local Interpretable Model-agnostic Explanations	[56]–[68]
CMAPSS [69]	Shapley Additive Explanations and Rule Based	[35]–[37], [70]–[80]
General Machine Faults and Failures [81]	Feature Importance, Shapley Additive Explanation, Class Activation Mapping and Local Interpretable Modle-agnostic Explanation	[42], [48], [82]–[87]
Trains	Feature Importance	[34], [88]–[93]
Gearboxes [94]	Shapley Additive Explanations and Interpretable Filters	[42], [45], [48], [95]
Artificial Dataset	Local Interpretable Modle-agnostic Explanation, Shapley Additive Explanation, Counterfactual and Surrogate	[44], [96], [97]
Hot or Cold Rolling Steel	Shapley Additive Explanations	[72], [96], [98]
Maritime	Feature Importance and Shapley Additive Explanations	[99]–[101]
Mechanical Pump	Physical Constraints, Knowledge-based and Sparse Networks	[102]–[104]
Hard Drives [105]	Shapley Additive Explanations, Rule Based, Local Interpretable Modle-agnostic Explanation and Decision Tree	[38], [70], [106]
Lithium-ion Batteries [107]	Layer-wise Relevance Propagation	[37], [108], [109]
Wind Turbines [110]	Autoencoder-based Anomaly Root Cause Analysis and Sparse Networks	[111], [112]
Amusement Park Rides	Depth-based Isolation Forrest Feature Importance and Accelerated Model-agnostic Explanations	[113], [114]
Particle Accelerators	Layer-wise Relevance Propagation and Feature importance	[115], [116]
Chemical plant	Shapley Additive Explanations	[117], [118]
Semi-conductors [119]	Shapley Additive Explanations and Knowledge-based	[120], [121]
Aircraft	Fuzzy	[52], [122]
Air Conditioners	Attention	[56]
Tennessee Eastman Process [123]	Rule-based	[70]
Compacting Machines	Accelerated Model-agnostic Explanations	[114]
UCI Machine Learning Repository [124]	Mahalanobis-Taguchi System	[125]
Transducers	Fuzzy	[126]
Heaters	Fault Tree	[127]
Computer Numerical Control data	Depth-based Isolation Forrest Feature Importance	[128]
Textiles	Visualization	[129]
Plastic Extruders	Shapley Additive Explanations	[130]
Press Machine	Shapley Additive Explanations	[131]
Coal Machinery	Shapley Additive Explanations	[132]
Refrigerators	Attention	[133]
Gas Compressors	Shapley Additive Explanations	[134]
Hydraulic Systems	Shapley Additive Explanations	[135]
Iron Making Furnaces	Signal Temporal Logic	[136]
Cutting Tools	Feature Importance	[137]
Power Lines [138]	Feature Importance	[139]
Communication Equipment	Surrogate	[140]
Water Pump	Fuzzy	[141]
Oil Drilling Equipment	Knowledge-based	[142]
Solenoid operated valves	Physical Constraints	[143]
Coal Conveyors	Digital Twin	[144]
Temperature Monitoring Devices	k-Nearest Neighbors	[145]
Distillation Unit	Rule-Based Interpretation	[146]
Water Pipes [147]	Statistical Model	[148]

applied to the problems of anomaly detection [72], [99], [117], [118], [131], [132], fault diagnosis [130], [134], and prognosis [75], [77], [120]. This is likely due to its wide versatility as a model-agnostic method that can provide global explanations.

Steurtewagen et al. [134] created a framework for fault diagnosis that consists of three parts: data collection, prognosis, and diagnosis. Importantly, in the data collection phase, they received the reports that were associated with the faults. The prognosis section used an XGBoost algorithm

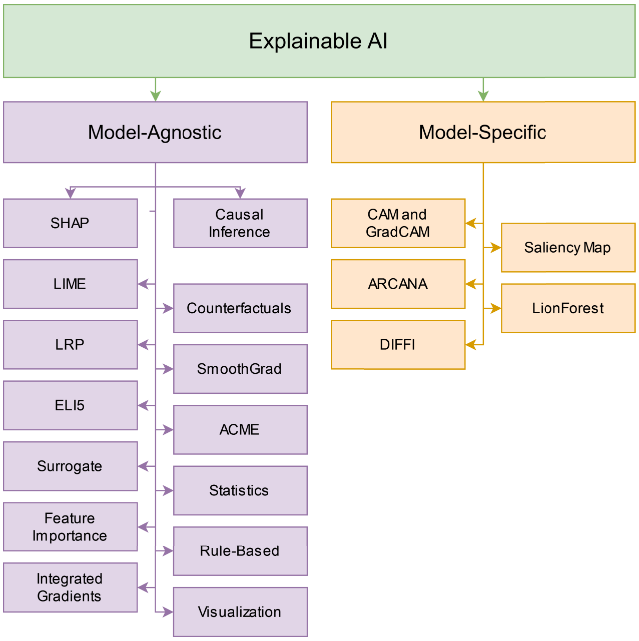


FIGURE 9. **XAI taxonomy.**

to detect a fault occurring. The diagnosis utilized SHAP to determine the features that are important to the output of XGBoost. These features are validated using the reports that accompany the fault.

Choi et al. [118] proposed a method for explainable unsupervised anomaly detection to predict system shutdowns for chemical processes. Their method consisted of what they call a period-independent framework and a period-integrated framework. The period-independent framework searched for the best anomaly detection model and applied the explainable method. In the period-integrated framework, they applied real-time information to the model chosen from the previous framework. They found that the isolation forest provided the best results in the period-independent framework based on the number of unplanned shutdowns detected, and they utilized show SHAP as an effective way of measuring root cause analysis.

Gashi et al. [120] conducted predictive maintenance on a multi-component system. Their objective was to model interdependencies and assess the significance of the interdependencies. Prior to training their Random Forest model, they used visual exploration to study interdependencies. They used two methods to justify the use of interdependencies: statistics and XAI. They used chi-squared testing to show that the performance of a model with interdependencies is better ($p < 0.001$). When applying SHAP to the random forest, they showed that the interdependency variables were usually among the top explainer features. This adds validity to SHAP as an explainable method in terms of the accuracy of its explanations.

Keleko et al. [135] utilized a fully connected deep neural network for predicting degradation states of a hydraulic system. Their model was able to predict different health states for five different internal components of the hydraulic system with high precision, recall and F1-score. To apply explainability, they utilized DeepSHAP, a mixture of DeepLIFT and Shapley values. This mixture allows for a better SHAP-based approach for deep neural networks as it is able to tailor the calculation of Shapley values to a deep neural architecture.

2) LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS (LIME)

LIME was introduced by Ribeiro et al. as a way of explaining any model using a local representation around the prediction [24]. This is done by sampling around the given input data and training a linear model with the sampled data. In doing this, they can generate an explanation that is faithful to that prediction while using only information gained from the original model.

Protopapadakis et al. [35] computed the RUL as applied to the CMAPSS turbofan dataset. They initially attempted to perform RUL prediction with two models, a random forest and a deep neural network. They found the random forest to perform poorly, which would lead to poor explanations. Their deep neural network achieved high performance, so they applied LIME. They compared two LIME explanations, one for early life and one for late life with a specific fault. They found that LIME was able to label the important features for failures that reflected the physical faults. Additionally, they showed that LIME would have a more difficult time labeling the important features when it was applied to segments with no faults as anything could occur in the future.

Allah Bukhsh et al. [93] discussed multiple tree-based classifiers for predicting the need for maintenance events, i.e., anomaly detection, for train switches. From their pool of tree-based classifiers, including decision tree, random forest, and gradient boosted tree, they identified gradient boosted tree as the most accurate amongst the models when predicting if a problem would occur. In a separate test, they had the same models predict specific types of anomalies. In this experiment, random forest outperformed the rest. For interpretability, they implemented LIME to learn from the outputs of the random forest. The researchers intend that the output from LIME will help establish trust in the model for domain experts and decision makers

3) FEATURE IMPORTANCE

Feature importance refers to the idea that some of the input features have more influence on the output than others. For example, when determining if an image is a dog, the background that has no pixels of the dog would potentially be less important than the pixels with the dog. Feature importance is typically assessed using techniques like SHAP and LIME, but various approaches exist in the literature.

Many researchers have applied different methods of feature importance calculations. Bakdi et al. [101] tackled predictive maintenance for ship propulsion systems. They combined balanced random forest models and multi-instance learning to achieve a high true positive rate which was then

TABLE 2. Explainable methods from the literature.

Method	Articles
Shapley Additive Explanations (V-A1)	[72], [96], [99], [117], [118], [130], [132] [36]–[38], [42], [66], [75]–[77], [120], [131], [134], [135]
Local Interpretable Model-agnostic Explanations (V-A2)	[35]–[38], [44], [50], [51], [54], [61], [66], [76], [84]
Feature Importance (V-A3)	[34], [54], [67], [85], [86], [93], [101], [115], [137], [139]
Layer-wise Relevance Propagation (V-A4)	[37], [44], [68], [87], [109], [116]
Rule-based (V-A5)	[65], [70], [71], [73]
Class Activation Mapping (CAM) and Gradient-weighted CAM (V-B1)	[37], [44], [48], [63]
Surrogate (V-A6)	[82], [90], [96], [140]
Visualization (V-A9)	[47], [74], [100], [129]
Depth-based Isolation Forrest Feature Importance (V-B2)	[42], [113], [128]
Integrated Gradients (V-A7)	[95], [131]
Causal Inference (V-A8)	[89]
Accelerated Model-agnostic Explanations (V-A10)	[114]
Statistics (V-A11)	[59]
Smooth Gradients (V-A12)	[131]
Counterfactuals (V-A13)	[98]
LionForests (V-B3)	[97]
Explain Like I'm 5 (V-A14)	[51]
Saliency Maps (V-B4)	[37]
Autoencoder-based Anomaly Root Cause Analysis (V-B5)	[111]

explained via Gini feature importance. Schmetz et al. [137] also applied Gini feature importance to verify a Tree Interpreter [150] for their random forest classifier.

Other researchers have ranked their features in different ways. Manco et al. [34] performed fault prediction to train systems where they ranked time steps by how anomalous they were within a time window. This ranking was performed by mixture modeling of the prior probability of the trend with the probability of the trend being normal behavior. Marcato et al. [115] applied anomaly detection to particle accelerators where permutation-based feature importance to guide further model development.

Finally, Voronov et al. [67] and Ghasemkhani et al. [86] each proposed different methods of calculating feature importance that tackle different problems. Voronov et al. proposed a forest-based variable selector called Variable Depth Distribution (VDD) that addressed the issue of variable interdependencies through clustering of features. The important features appeared in multiple clusters. Ghasemkhani et al. developed Balanced K-Star to deal with the imbalance problem commonly found in predictive maintenance. To add explainability, they applied chi-square to determine the important features in the machine failure.

4) LAYER-WISE RELEVANCE PROPAGATION (LRP)

LRP was introduced by Bach et al. [151] as an explainable method that assumes that a classifier can be decomposed into several layers of computation. LRP works with the concept of a relevance score that measures how important a feature is to an output. LRP works by extrapolating the relevance to the input layer by moving backwards through the architecture starting at the output layer. The importance of an input feature can then be measured as a summation of features it impacts through the architecture.

LRP falls into the category of model-agnostic which can be seen in the use-cases in the literature. Felsberger et al. [116] applied LRP to multiple architectures including kNN, random forest, and CNN-based models. Through LRP, they found that the CNN architectures were learning important features which led to higher performance. Han et al. [68] performed fault diagnosis for motors using the notable model LeNet [152]. Through the use of LRP, they were able to bring explainability to a notable architecture.

Wang et al. [109] proposed a method of using explainability as a method of driving the training process. They utilized LRP to calculate feature importance for the training data. The importance calculations were embedded for optimizing the model's performance. They introduced this explainability-driven approach to the problem of aging batteries, and showed its superb accuracy when compared to a data-driven approach.

Grezmak et al. [87] proposed utilizing LRP as a method of incorporating explainability to the problem of fault diagnosis of machine failure. First, they apply Continuous Wavelet Transform to change the time series information to a multiscale time-frequency images. They train a convolutional neural network (CNN) using these images to be able to classify one of four potential faults that would be occurring. This is followed up by their LRP implementation for added explainability. They found LRP able to show unique elements in the time-frequency images that could map to each class of fault. The most interesting finding occurred when comparing CNNs trained on the pure time series and discrete-fourier transformed data. LRP was not able to show consistent explanations for items of the same class for these two CNNs even though all of the trained models performed high with accuracies. They argue this is due to the generalizability of the time-frequency domain transformation which shows an interesting use case of explainable methods: verifying the generalizability of a model or data transformation method.

5) RULE-BASED EXPLAINERS

Rule-based explainers use a combination of the black-box model and the training data to create a series of IF-THEN rules. These rules are generally created using combinatorial logic (ANDs, ORs, and NOTs) to combine the features in the IF portion of the rules. The THEN portion of the rules are populated by the result from the model, usually a class or a predicted value. The rules are then presented as explanations or may be used as a replacement for the black-box model itself.

Even in rule-based explainers, there are numerous methods that have been used. Wu et al. [71] proposed the K-PdM (KPI-oriented PDM) framework, a cluster-based HMM based on key performance indicators (KPIs). A KPI is a vector of one feature of fine-grained deterioration, and a combination of

KPIs reflect the health of a machine. The health was modeled as an HMM for each KPI. These HMMs were converted into a rule-based reasoning system for explainability.

Brunello et al. [70], [73] showed twice that temporal logic can be used in anomaly detection. Firstly, they showed that linear temporal logic could be added to an online system for monitoring failures [73]. They again showed that temporal logic could be used in a different approach to the same problem. Brunello et al. [70] created syntax trees that utilized bounded signal temporal logic statement. The trees were altered using an evolutionary approach to predict failure in Blackblaze Hard Drive [105], Tennessee Eastman Process [123], and CMAPSS [69] datasets, commononly used datasets for PdM of hard drives, electrical processes and turbofans. This method led to great performance with rule-based explanations.

Ribeiro et al. [65] applied XAI to the online learning process using a Long Short-term Memory AutoEncoder (LSTM-AE) for modeling public transport faults. Simultaneously, the authors' system learned regression rules that explained the outputs of the model. While their system was learning to map the anomalies, the output of their model was fed into Adaptive Model Rules (AMRules), a stream rule learning algorithm. They applied their method to four public transport datasets, and they output their global and local rule-based explanations given used in their system.

6) SURROGATE MODELS

Surrogate models are simpler models that are used to represent more complex models. These surrogate models generally take the form of simple decision trees and linear/logistic regression models. The simplistic nature of these models makes them interpretable; however, their use has their interpretability as an explainable method for a black-box model.

When utilizing a surrogate model as an explainable method, the surrogate model must be inherently interpretable as a way of allowing an explanation to be gathered from the main model. Glock et al. [82] utilized two ARIMA models to explain a random forest model. One ARIMA model learned the same data as the random forest, and the second ARIMA model learned the residual errors from the random forest. While the random forest is not explainable, the two ARIMA models could show what the random forest could and could not learn.

Zhang et al. [140] proposed an alarm-fault association rule extraction based on feature importance and decision trees. Their process started with a weighted-random forest. Feature selection was performed to gather the important features in the abnormal state. These features were used to create a series of C4.5 decision trees that model different features. Once their random forest was trained and predicted a fault, the decision tree with the highest accuracy could be used to extrapolate an explanation of the fault.

Errandonea et al. [90] tested XAI on edge computing with all possible models in H2O.ai's AutoML to perform their fault diagnosis. After determining the optimal architectures, they trained a decision tree surrogate model to add explainability to their autoML process. By optimizing hardware and accuracy, they showed that explainable predictive maintenance could theoretically occur on edge computing devices.

7) INTEGRATED GRADIENTS

Integrated gradients was introduced by Sundararajan et al. [153] to attribute the prediction of a deep architecture to its input features. They introduce two axioms, sensitivity and implementation invariance, to build their explainable method. Sensitivity is achieved if *for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution.* Implementation invariance means *attributions are always identical for two functionally equivalent networks.* With these axioms in mind, the integrated gradients are calculated via small summations through the layers' gradients.

Hajgato et al. [95] introduced the PredMaX framework for predictive maintenance which identified sensitive machine parts and clustered time periods. It works in two steps: a deep convolutional autoencoder was applied to the data, and clustering was performed on the latent space in the autoencoder. From the clusters, they showed which channels contribute to the transition from normal to abnormal. Additionally, the integrated gradients technique was used to extract the relevant sensor channels for a malfunctioning machine part.

8) CAUSAL INFERENCE

Causality goes beyond the notion of statistics dependencies as it shows a true relationship between two or more variables [154]. Causality can be measured in *causal strength* which measures the change in distribution of n-1 variables when one variable has been changed [154]. Causality is not an easy quality to analyze as it can only be truly discovered by repeated observations of a phenomenon occurring given an event; however, causal inference has been a method of XAI that some researchers have utilized.

Trilla et al. [89] designed an anomaly detection framework based around a denoising variational autoencoder (VAE) and an MLP. They extracted intra-subsystem and inter-subsystem patterns by making the time series data into voxels. The VAE generalized the embeddings. Finally the MLP was used to create a smooth diagnosis probabilistic function. They applied their method on a locomotion dataset and utilized causal inference via the Peter-Clark algorithm to answer the question "Did the VAE learn cause-effect relationships?" They found that the VAE could at best be described as modeling a correlation relationship, but this limitation was mainly attributed to limited data availability.

9) VISUALIZATION

Visualization techniques do not take any one specific form. Generally, these visualizations take the form of visualizing

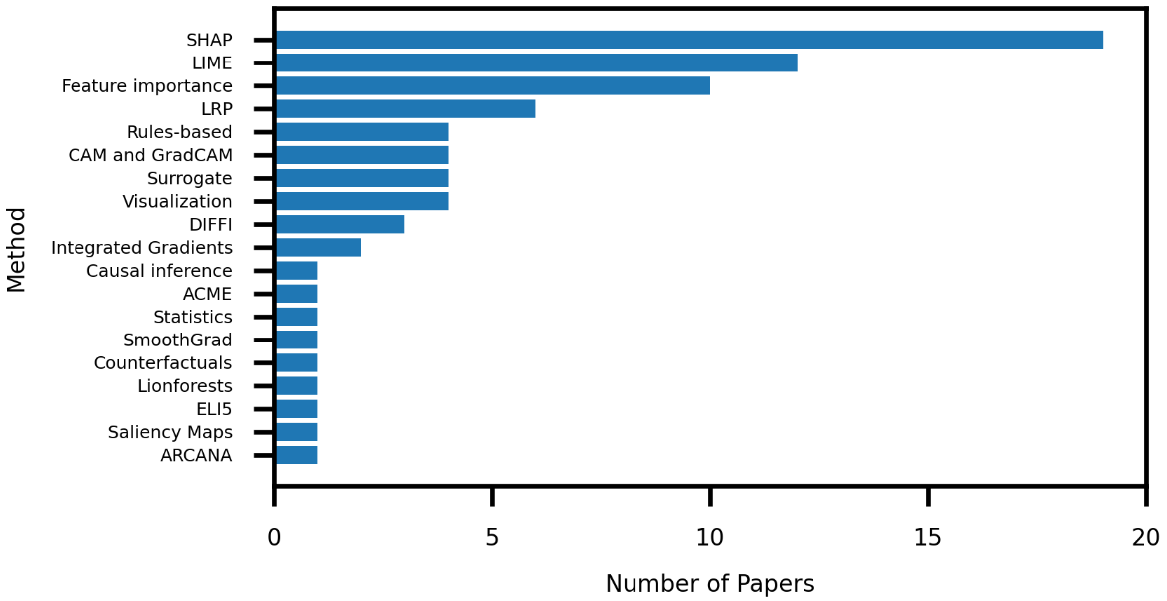


FIGURE 10. Use of XAI methods.

weights; however they may also take the form of visualizing specific examples. Whatever the case, these methods benefit the users by providing an image enlightens the user to the inner workings of the architecture.

Visualizations can be utilized in many ways for explainability. Michalowska et al. [100] use visualizations to compare healthy and anomalous data. Costa et al. [74] utilized visualizations coupled with a recurrent variational encoder. They show that the latent space created by the encoder can add explainability. When input data with similar RULs pass through the encoder, they show the latent spaces are similar for those with similar RULs.

Xin et al. [47] aimed to address bearing fault diagnosis via a novel model named logarithmic-short-time Fourier transform modified self-calibrated residual network (log-STFT-MSCResNet). The STFT extracts time-frequency features from raw signals to retain physical meaning of fault signatures which are visualized for explainability. The MSCResNet is used to enlarge the receptive field without introducing more parameters. With the combination of the two, they aim to have high accuracy even under unknown working conditions. They compared their model to popular models such as LSTM and ResNet18. log-STFT-MSCResNet performed among the best even under unknown working conditions, had a small number of features and had a shorter training time than the others.

10) ACCELERATED MODEL-AGNOSTIC EXPLANATIONS (ACME)

ACME was introduced by Dandolo et al. [155] as a method of quickly generating local and global feature importance measures based on perturbations of the data. For global explanations, they take a vector that holds the mean of each feature through the entire dataset; this is known as the baseline vector. Then a variable-quantile matrix is created that holds the different quantiles of the features. This matrix is used to gather predictions that would represent each quantile. The global feature importance is finally calculated for each feature by computing the standardized effect over each quantile. To get a local explanation, the baseline vector is replaced with the specific data point that is meant for explaining.

Anello et al. [114] applied ACME to the problem of anomaly detection to compare it to SHAP. They utilized isolation forest to detect anomalies as it is commonly used for detecting outliers or anomalies. An anomaly score was used as a label for the time series to represent the problem as a regression task which allows ACME to be applied. After applying SHAP and ACME to a roller coaster dataset and a compacting machine dataset, they found a drastic speed up by using ACME with all of the data while SHAP would be slower even with access to 30% of the data.

11) STATISTICS

As a method of explaination applied to the problem of predictive maintenance, statistical tests can be used to compare the distribution of the features between different classes.

Fan et al. [59] developed ML methods that take advantage of physics knowledge for added interpretability. Their case study was fault detection of leak-related faults in vehicle air systems. They applied three physics equations to their data that would model the air leakage. Moreover, they used that data in the training data of their kNN and MLP models. Results showed that the physics-assisted models to outperform the non-assisted models.

12) SMOOTH GRADIENTS (SMOOTHGRAD)

SmoothGrad was developed by Smilkov et al. [156] to produce a gradient-based sensitivity map. The intuition behind SmoothGrad involves differentiating the predicting model with respect to the input. This derivative creates a sensitivity map that represents how much difference a change in each pixel of the input would make to the classification [156]. Moreover, this sensitivity map can ideally show regions that are key to the prediction. Serradilla et al. [131] utilized this method in combination with others; therefore, the information surrounding this work can be found in V-C.

13) COUNTERFACTUALS

Counterfactuals were introduced by Wachter et al. [157] to provide statements of the differences needed to gain the desirable outcome. This method also works by providing an explanation for the output of the model, but this extra capability makes counterfactuals very unique in realm of XAI methods.

Jakubowski et al. [98] developed a predictive maintenance solution for an industrial cold rolling operation. They utilize a semi-supervised algorithm based on the Physics-Informed Auto-Encoder (PIAE). This architecture was physics-informed by applying a list of equations at the beginning of their input data. The output of the equations was appended to the input data of their AE. Their model proved to be more accurate than a base AE. While PIAE has some interpretable aspects already, they applied counterfactuals as an explainability method to show the important features from their algorithm's decisions.

14) EXPLAIN LIKE I'M 5 (ELI5)

ELI5 is a popular method from Github [158] maintained by the user TeamHG-Memex and 15 other contributors. This Python library focuses on explaining the weights of a model which also serves as a method for calculating feature importance. While maintaining original methods, ELI5 also provides other explainability method implementations. Serradilla et al. [51] utilized this method; however, their work is presented in V-C as they applied multiple methods.

B. MODEL-SPECIFIC

This section describes the explainable methods in our population that base the explanations on the properties of the architecture it intends to explain. These methods are known as *model-specific* [149]. Here we discuss methods that take advantage of the architecture for generating explanations such as CAM and GradCAM in Section V-B1, DIFFI in Section V-B2 and more.

1) CLASS ACTIVATION MAPPING (CAM) AND GRADIENT-WEIGHTED CAM (GRADCAM)

CAM was introduced by Zhou et al. [25] as a method of global explainability for convolutional neural networks (CNN). The map that is created indicates the image regions that are used by the CNN to identify the target category. CAM does this by utilizing a global average pooling (GAP) layer in the CNN architecture which outputs the spatial average of the feature map of the final layer. The pixels with higher values are associated with the pixels in the image associated with the class label. Additionally, Selvaraju et al. [159] extend CAM to GradCAM by using the gradient information going into the last convolutional layer to understand the importance of the features.

GradCAM has been validated through different studies via comparison and metrics. Mey et al. [44] focuses on the plausibility of XAI for explaining a CNN. They investigated GradCAM, LRP and LIME as methods of explaining a CNN for anomaly detection. They found non-distinguishable features highlighted by LRP, and they found unimportant features highlighted by LIME. GradCAM was able to highlight the important features that they labeled prior to CNN training. This could point towards model-specific methods outperforming model-agnostic methods when applicable.

Solis-Martin et al. [37] present a comparison on LIME, SHAP, LRP, Image-Specific Class Saliency (Saliency Maps) and GradCAM as applied to predictive maintenance datasets such as CMAPSS and batteries. They identify eight metrics for comparison: identity, separability, stability, selectivity, coherence, completeness, congruence and acumen, an evaluation proposed by the authors. When comparing the different methods as applied to a CNN architecture, GradCAM performed the best in regards to the nine metrics.

Oh et al. [63] propose a fault detection and diagnosis framework that consists of a 1D-CNN for fault detection, class activation maps for fault diagnosis (explainable method) and VAE for implementing user feedback. The CNN utilizes a GAP layer as the output layer due to its ability to maintain the temporal information. This also allows them to use CAM as an explainable method as opposed to GradCAM. The VAE is utilized with the principle of Garbage-In, Garbage-Out logic to minimize the amount of false positives and negatives that would be presented to the users. To verify their method, they apply it to the Ford Motor dataset which is a vehicle engine dataset that contains an amount of noisy data. They show that their model is accurate even in noisy data, and they show that the VAE increases their accuracy. They also show via CAM that the anomalous data is linearly separable, which is found in the VAE.

2) DEPTH-BASED ISOLATION FORREST FEATURE IMPORTANCE (DIFFI)

DIFFI was introduced by Carletti et al. [160] as an explanable method for isolation forests. Isolation forests are an ensemble of isolation trees which learn outliers by isolating them from the inliers. DIFFI relies on two hypotheses to define feature importance where a feature must: induce the isolation of anomalous data points at small depth (i.e., close to the root) and produce a higher imbalance on anomalous data points while being useless on regular points [160]. These hypotheses would allow explanations for anomalous data which would allow for explanations of outliers or faulty data.

Berno et al. [113] performed anomaly detection for automated rides at entertainment parks. They introduced the idea of providing extra focus specific features by splitting their data into a multivariate set and many univariate sets based on a prior knowledge. They utilized isloation forest to model the multivariate time series with DIFFI explaining the output. They modeled the univariate time series with a Growing When Required (GWR) neural gas network. The multivariate analysis was used for determining anomalies within most of the variables, and the explanations were used to rank the features causing the anomaly.

Lorenti et al. [128] designed an unsupervised interpretable anomaly detection pipeline known as Continuous Unsupervised Anomaly Detection on Machining Operations (CUAD-MO). CUAD-MO consists of 4 parts: data segmentation and feature extraction, unsupervised feature selection via Forward Selection Component Analysis (FSCA), anomaly detection via Isolation Forest, and post-hoc explainability via DIFFI. Their feature extraction consisted of adding basic statistics and higher order moments of the signals such as Kurtosis. FSCA iteratively selects features to maximize the amount of variance explained. Finally, the Isolation Forest is used to detect outliers which are handled as faulty events. These are explained via DIFFI. They applied their method to 2 years of computer numerical control data resulting in a 67% precision rate.

3) LIONFORESTS

LionForests were introduced by Mollas et al. [161] as a local explanation method specifically for random forests. Their method follows these steps: estimating the minimum number of paths for the accurate answer, reducing the paths through association rules, clustering, random selection or distribution-based selection, extracting the feature-ranges, categorical handling of features, composing the interpretation, and visualizing the feature ranges. The outputs of their method are the interpretations in the form of IF-THEN rules and visualizations of the features.

Mylonas et al. [97] aimed to alleviate the non-explainable nature of random forest by applying an expanded version of LionForests to fault diagnosis. They expanded LionForests into the realm of multi-label classification by applying three different strategies: single label, predicted labelset, and label subsets. Single label aims at explaining every individual prediction (local); predicted labelset aims at explaining all predictions (global); and label subsets aim at explaining based on frequently appearing subsets of predictions. With their expansion, their attention is focused on multiple machine failure datasets, but specifically the AI4I dataset [162]. They utilized accuracy metrics such as precision, and they provided metrics for their explanations such as length of explanations and coverage of data. One of the more notable elements of their work involves comparing their XAI algorithm to other algorithms, namely global and local surrogates and Anchors.

4) SALIENCY MAPS

Saliency maps were introduced by Simonyan et al. [163] as a method for explaining CNN outputs. Given an input and a model, saliency maps rank the pixels of the input based on their influence on the output of the model. This is done by approximating the output with a linear function in the neighborhood of the input by using the derivative of the scoring function with respect to the input. This approximation is the saliency map. Solis-Martin et al. [37] utilized saliency maps in a comparison experiment where they found GradCAM to be best in their use-case. More information about this experiment can be found in Section V-B1.

5) AUTOENCODER-BASED ANOMALY ROOT CAUSE ANALYSIS (ARCANA)

ARCANA was introduced by Roelofs et al. [111]. They noticed that autoencoders were a popular method of detecting anomalies in their target domain, wind turbines; however by themselves, autoencoders are not interpretable. To overcome this lack of interpretability, they implement ARCANA as a way of explaining the cause of the reconstruction error of an autoencoder. ARCANA works by minimizing a loss function that is based on reconstruction. As opposed to measuring the difference between the output of the autoencoder and the input, they add this bias vector to the input data as to have a *corrected input*. Moreover, the bias shows "incorrect" features based on the output; therefore, the bias would explain the behavior of the autoencoder by showing which features are making the output anomalous.

Roelofs et al. [111] also utilize their method for anomaly detection and root cause analysis for wind turbines. They verify that ARCANA provides the most important feature causing the issues with their wind turbines. This method is done by firstly measuring the features reconstruction error. When performing ARCANA, the feature that shows the most importance is the same feature with the largest error. They then show that even when the feature does not appear in the reconstruction error, ARCANA is able to find feature importance in sensors that are applicable to known anomalies.

C. COMBINATION OF METHODS

This section describes the works that used multiple explainability methods. Some of these works were utilized to just note the differences between the different explainable methods. Other works compared the methods as to determine the better method. This section reviews the works that combine multiple methods without aiming to declare one method as better than another.

Utilizing multiple explainable methods can be used in a stacked manner or in a simultaneous manner. The stacked manner involves using explainable methods sequentially. In Jakubowski et al. [96] they created a quasi-autoencoder for explainable anomaly detection. A surrogate model of

XGBoost was used as a way of simplifying the original model. They achieved a high R^2 score using this XGBoost model while adding explainability via TreeExplainer (SHAP).

More commonly, a simultaneous utilization of explainable methods appears in the literature where the authors obtain multiple explanations from different methods. Khan et al. [36] found the best architecture for their problem of RUL prediction amongst: random forest, SVM, gradient boosting, elastic net GLM and an MLP regressor. After seeing the MLP regressor to have the best performance, they used LIME and SHAP to explain the output. LIME and SHAP did not have the same explanations, but they had similar explanations. Similarly, Jakubowski et al. [76] performed an experiment testing five architectures and using SHAP and LIME as explainers. The found that SHAP and LIME had different explanations throughout the different neural architectures suggesting a fidelity concern between architectures.

Like the prior two, Serradilla et al. [51] performed remaining useful life prediction on a bushings testbed. They tested six different models and determined random forest regressor to be the best. They then utilize two explainability methods (ELI5 and LIME) to show global and local feature importance of driving model development. Additionally in future work, Serradilla et al. [131] utilized a combination of SHAP, Integrated Gradients and SmoothGrad to explain the connection between the variables and the loss of their deep architecture for anomaly diagnostics. Brito et al. [42] performed a large experiment that applied many unsupervised learning algorithms for fault detection and fault diagnosis. They showed that Local-DIFFI and SHAP seemed to be mostly in agreement about the explanation for the model's output, but they did not move further in asking which is better.

Ferraro et al. [38] focused on analyzing the effectiveness of explainability methods on the predictions of a recurrent neural network based model for RUL prediction. Notably, the model performed well, but the focus was on the explainable methods SHAP and LIME. A quantitative analysis was performed using three metrics: identity, stability and separability. This showed: (1) LIME was unable to give identical explanations for identical instances; (2) LIME more than SHAP gave similar explanations to instances in the same class; and (3) LIME and SHAP were able to give different explanations for instances in different classes.

Li et al. [66] aimed at integrating explainability into an AutoML environment used for vehicle data. They tested four different AutoML platforms: AutoSklearn, TPOT, H_2O, and AutoKeras. They performed two different experiments where they provided different subsections of their dataset with both resulting in TPOT performing the best in accuracy. Finally, they apply LIME and SHAP to the resulting model to explain a local sample and the whole model. Their work results in a defined workflow for an automatic predictive maintenance system that includes explainability.

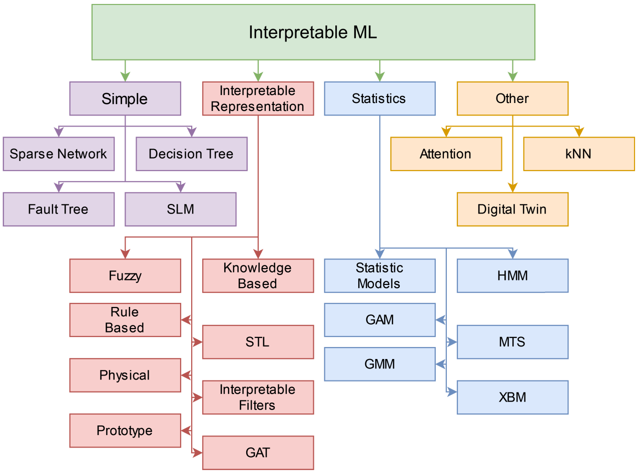


FIGURE 11. Interpretable ML taxonomy.

VI. INTERPRETABLE ML IN PREDICTIVE MAINTENANCE

Interpretable machine learning (iML) encompasses many methods whose inner-workings are understandable without requiring a post-hoc method for explanation generation. These methods can be interpreted by the target audience without the need of separate methods to serve as a translator between the model and the person. iML methods namely consist of architectures that can have human-readable outputs such as rule-based systems, simple visual representations such as decision trees and simple networks or physical mappings that are intelligible to the user. The overarching taxonomy can be seen in Fig.11.

A. SIMPLE ARCHITECTURES

What is denoted as a "simple" architecture is really just an architecture that is small in the number of weights or a tree like architecture. A small number of weights indicates a straightforward understanding of the model's performance as there are no layers one has to decipher. Additionally, tree-based architectures are simplistic in nature as one is able to visualize a tree structure and follow the reasoning of a small enough tree. This section consists of Sparse Networks, Section VI-A1, Decision Trees, Section VI-A2, Fault Trees, Section VI-A3, and Symbolic Life Models, Section VI-A4.

1) SPARSE NETWORKS

Sparse networks are neural networks that are limited in their architecture. Large deep neural networks are inherently blackbox models; however, interpretable whitebox models can take the form of very simple neural network models such as linear regression or logistic regression models. As the models are simple, the impacts of the input features can be seen as they are propagated through the network.

Beretta et al. [112] utilized two different models for predictive maintenance: a gradient-boosting regressor to model the normal data and an isolation forest to model the fault data. The output of these are merged with a mean average of the temperature readings to create a score of failure. The authors praise the simplicity of the algorithms as the source of interpretability in their method.

Pu et al. [46] explored a new frequency domain space they call the restricted sparse frequency domain space (RSFDS) for rolling bearing faults. The RSFDS breaks down the features into a space that is made of real and imaginary points. This space is able to visualize boundaries that have physical meanings to the faults. They use a simple two-layer neural network to these points, and they achieve high performance equal to that of a CNN-LSTM with less memory and CPU usage.

Langone et al. [104] proposed a model for interpretable anomaly prediction based on a logistic regression model with elastic net regularization. Their method is made of 3 steps: data preparation, learning and refinement of the prediction model. In the data preparation phase, they categorize the data using included statistics, apply windowing to the data, and finally mark the windows as either being anomalous or not. The learning phase consists of learning the relevant features from the windowed data. This includes considering the feature distributions across failures and non-failures and measuring the distance according to the Kolmogorov-Smirnov metric. The refinement of prediction model phase consists of the training and utilization of the logistic regression model. Coupled with elastic net regularization, this model selected a smaller subset of the original data and captures the variable correlations. They applied their method to a plunger pump in a chemical plant and produced relative good and consistent scores.

2) DECISION TREES

Decision trees encompass both classification and regression trees that date back to the first regression tree algorithm proposed by Morgan and Songquist [164]. Decision trees create a tree-based architecture where each set of children of each node is split using a feature. To produce an output, a decision tree algorithm starts at the root of the tree and proceeds down the tree by evaluating the feature that is used for splitting. The output corresponds to the final leaf node that the decision trees reaches on its path.

Li et al. [92] perform failure prediction with a combination of two methods. Firstly, they perform alarm prediction,a specific failure prediction, using an SVM based model. To make the output human-readable, they perform an exhaustive search among the feature space and visualize the first two principle components. With the visualization they are able to make some basic rules that predict the alarm. For failure prediction, specific failures besides general alarm, they use a decision tree. This decision tree can be translated directly into rules as their model is highly interpretable.

Amram et al. [106] utilized two types of decision trees, optimal classification trees [165] and optimal survival trees [166]. Their goals included predicting the RUL of long-term health of hard drives, predicting RUL of the short-term health of hard drives, predicting failure classification in short-term health of the hard drives and performing similar experiments with limited information. Their results showed that they could gather better results using separate models for the tasks as opposed to using one model. They also showed the interpretable methods shared many of the important features for the different tasks.

Panda et al. [57] aimed at tackling the problem of commercial vehicle predictive maintenance by designing an interpretable ML framework. To simplify their problem, they solely looked at the air compressor system. By looking at the air compressor system, they ran a broad experiment that analyzed different configurations of models and data. The C5.0 with boosting model performed the best, and the inclusion of Diagnostic Trouble Codes with the sensor data raised the performance metrics.

Simmons et al. [139] argued that the dynamics of a time-series are in themselves discriminative of health or failure. Additionally, the dynamics are interpretable because they are derived directly from the information. These ideas were translated into the data mining domain by creating features that represent shorter time series in the temporal, spatial, and mixed domains. The features went through a rank-based selection process which gathered features that were statistically different between classes. These features were used to train a Light Gradient Boosting Machine (LightGBM) which is a type of gradient boosting decision tree introduced by Ke et al. [167]. This method allows for constant monitoring of feature importance during training which can be used for interpreting the results.

Matzka et al. [83] provided two main findings. Firstly, they provided a synthetic dataset for predictive maintenance that others in the community can use. Secondly, they utilized a bagged tree ensemble where each decision tree was trained on different combinations of features. They found that their ensemble could correctly identify three of the four failure modes consistently. Additionally, they provided the user with the features that deviate the most from healthy as to provide only the most useful information.

3) FAULT TREES

Fault trees were introduced by H.A. Watson at Bell Labs in 1961 [168]. Fault trees were introduced as an understandable model that can learn complex systems and perform root cause analysis. They are tree-like structures that are created using different types of nodes: basic events, gate events, condition events, and transfer events. Basic events are the nodes that represent either a failure event or a normal operating event. Gate events are the logic combining nodes and consists of AND, OR, Inhibit, Priority and Exclusive OR. Condition events represent conditions that must occur for a gate event to occur. Transfer events are nodes that point to somewhere else in the tree. With all of these gates, fault trees are able to learn root causes for different faults that can occur in a system.

Verkuil et al. [127] noticed that fault trees are made via human intervention. With the idea of automating the process,

TABLE 3. Interpretable methods from the literature.

Method	Articles
Attention (VI-D1)	[56], [58], [64], [78], [133]
Fuzzy (VI-B1)	[52], [53], [126], [141]
Knowledge-based (VI-B2)	[103], [121], [142]
Sparse Networks (VI-A1)	[46], [104], [112]
Interpretable Filters (VI-B3)	[45], [49], [60]
Physical Constraints (VI-B4)	[55], [102], [143]
Statistical Model (VI-C1)	[41], [148]
Decision Tree (VI-A2)	[57], [106]
Fault Tree (VI-A3)	[79], [127]
Graph Attention Networks (VI-B5)	[88]
Gaussian Mixture Model (VI-C3)	[108]
Explainable Boosting Machine (VI-C6)	[76]
Hidden Markov Model (VI-C2)	[80]
Prototype (VI-B6)	[62], [122]
Signal Temporal Logic (VI-B7)	[136]
Digital Twin (VI-D2)	[144]
Symbolic Life Model (VI-A4)	[33]
Generalized Additive Model (VI-C4)	[43]
Mahalanobis-Taguchi System (VI-C5)	[125]
k-Nearest Neighbors (VI-D3)	[145]
Rule-based Interpretations (VI-B8)	[146]

they applied the C4.5 tree combined with LIFT to create fault trees for domestic heaters. C4.5 is used to learn the failure thresholds of the sensor data. LIFT creates fault trees in an iterative process using the learned features. While they do not provide a performance metric, they note that their method cannot be optimal for the reasons of oversimplifying the problem and using a greedy heuristic. However, domain experts weighed in on the explanations provided in a positive manner.

Waghen et al. [79] utilized fault trees to perform interpretable time causality analysis. Their methodology consisted of building multiple logic trees for each subset of data. These logic trees were aggregated into one fault tree representing the multiple trees. They performed interpretatable time cause analysis by going through each variable in the fault tree. By traversing the fault tree, they were able to extrapolate rules that can model the causality through time towards faults.

4) SYMBOLIC LIFE MODEL (SLM)

Symbolic life models aim to alleviate the black box effect by modeling the process learned by mapping relationships and results. Symbolic life models are a form of symbolic regression based on genetic programming. This method creates a tree representation of an equation where the nodes are an input, a mathematical expression or a number. The output of the tree given an input is found by traversing the tree and performing the mathematical expressions as nodes are expanded. The genetic algorithm is used to perform crossovers and mutations based on the different mathematical functions and numbers where the goal is to maximize the tree's performance on a given dataset. For more detailed information, we recommend Augusto and Barbosa [169].

Ding et al. [33] proposed the use of symbolic life models, specifically dynamic structure-adaptive symbolic approach (DSASA), as a way of modeling RUL. DSASA combines the evolving methods of symbolic life models with the structure of adaption methods. An initial symbolic life model is created from a genetic programming algorithm and run-to-failure data. This is followed by the dynamic adjustment to the life models based on the performance on real-time information. This creates groups of improved models that can all be used for prediction. The life models are interpretable as they are simple models that perform based on the physical constraints.

B. INTERPRETABLE REPRESENTATIONS

An interpretable representation is the idea that a model or architecture is able to represent its knowledge or the training/testing data in an interpretable fashion. This may consist of representing the decision or data using rule structures such as Fuzzy Knowledge, Section VI-B1, or Rule-based structures like Signal Temporal Logic and Rule-based Interpretations, Section VI-B7 and Section VI-B8 respectively. Additionally, they may use grounding in their model to represent simple sin waves, as in Section VI-B3 Interpretable Filters, or real life processes, Section VI-B4 Physical Constraints. This section goes over many different ways to take a difficult process, such as predictive maintenance, to an simpler, more interpretable dimension.

1) FUZZY-BASED

Fuzzy logic was introduced by Zadeh [170] as a way of understanding the approximate mode of reasoning as opposed to the exact. Following this approximate model of understanding, all knowledge would come with a degree of confidence as opposed to a statement being 100% in a category. This adds some interesting and useful components to machine learning as these in-between categories can be utilized in a way that is different from having all information fall strictly into one category.

Fuzzy-based methods apply fuzzy logic in different ways. Lughofer et al. [53] and Kothamasu et al. [52] used type 1 fuzzy logic. Lughofer et al. proposed a framework of representation learning based on transfer of fuzzy classifiers. The transfer learning matched the distributions between the source data and the target task using fuzzy rule activation. This was done by feeding the model all of the source data and the healthy data from the target domain. Through this training, the model classified unseen healthy and unhealthy data from the target task. Their model did not outperform all black box models; however, it was in the upper ranks of performance while bringing interpretability to the user.

Additionally, Kothamasu et al. [52] presented a Mamdani neuro-fuzzy modeling approach for two use cases, bearing fault detection and aircraft engine fault diagnosis. They chose this model as it has the characteristics of being adaptive, flexible, lucid, and robust. Their model consists of five layers: input, linguistic term input, rules, linguistic terms output, and defuzzification. As the rules can become undistinguisable through training, they utilized Kullback-Leibler mean information to refine the rules.

Fuzzy-based methods can also take the form of higher-order fuzzy logic as seen by Upasane et al. [126], [141].

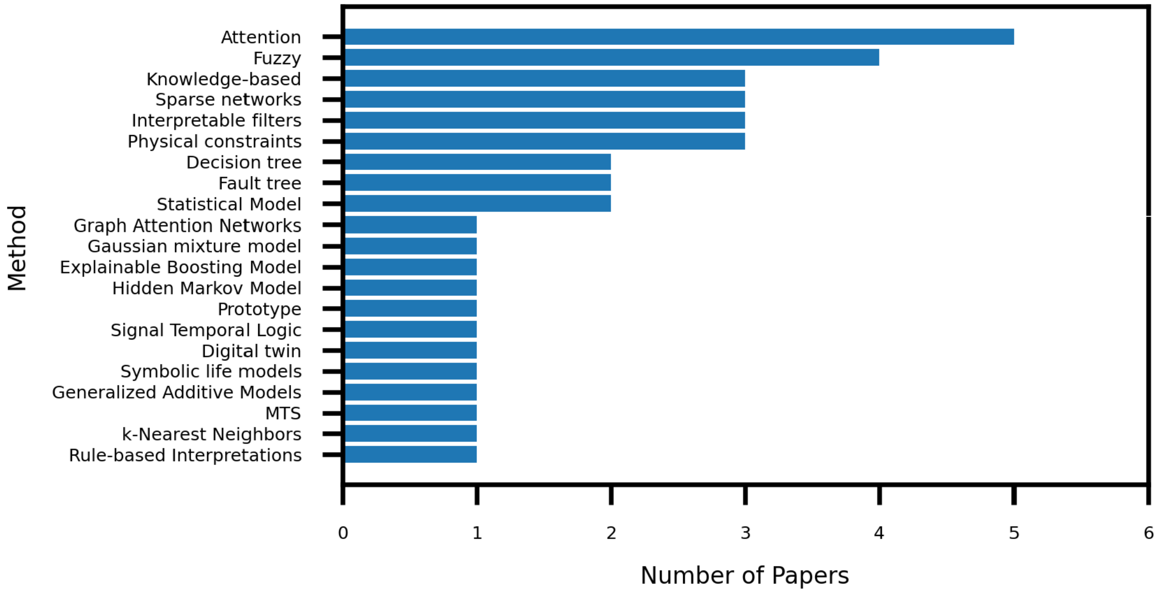


FIGURE 12. Use of iML methods.

They proposed a type 2 fuzzy logic system for fault prediction to allow interpretability [126]. Additionally, the Big-Bang Big-Crunch (BB-BC) evolutionary algorithm was used for optimizing the number of antecedents of their fuzzy logic system. This was optimized for minimizing the RMSE of their system. Their system was able to get a very low RMSE with 100 rules and six antecedents per rule.

Upasane et al. [141] extended their previous work [126] to include most of the faults that can occur as well as proposing an explainable framework. While maintaining accuracy with more faults is noteworthy, the experiment's measurement of users' trust was quite unique compared to the literature. They observed that 80% of the respondents agreed or strongly agreed with having trust in the interpretable system. This trust is attributed to the explainable framework and interpretable nature of their architecture; moreover, the interface is noted to provide helpful insights to the users that would minimize downtime of the assets.

2) KNOWLEDGE-BASED

In this paper, *knowledge-based* approaches include methods such as knowledge-graphs, knowledge-based systems, etc. Knowledge-based approaches focus on a symbolic representation of the data that one can find in a source of data. These representations consist of connections between different features where the links take the form of a link when discussing graphs or production rules when discussing production systems. These methods produce interpretation by providing these connections within the features, usually in the form of natural language.

Xia et al. [142] proposed a maintenance-oriented knowledge graph to apply for predictive maintenance of oil drilling equipment. Once they had the maintenance-oriented knowledge graph, an attention-based compressed relational graph convolutional network (ACRGCN) was used to predict solutions for different faults by predicting links between knowledge. This method also explained faults due to its knowledge-graph that maps different symptoms and maintenance requirements. Even though knowledge-graphs have inherent interpretability, they created a question-answer system that allowed the user to query the graph.

Salido et al. [103] created a fuzzy diagnosis system based on knowledge-based networks (KBN) and genetic algorithms (GA). The KBN constructed fuzzy rules using neural learning where the input is the features and the following layers are OR neurons and AND neurons. To determine the optimal number of neurons, they used a GA. Importantly in their GA, they added a metric to measure simplicity of their rules by making more concise rules. With their architecture, they could 1) detect a fault and 2) explain the fault using an IF-THEN rule which can be used as a method of root cause analysis.

Cao et al. [121] created an approach based on knowledge-based systems for anomaly prediction. Their method is broken into three parts: pruning of chronicle rule base, integration of expert rules, and predictive maintenance. Pruning of chronicle rule base consists of mining the rules with frequent chronicle mining, translating the rules into SWRL rules, and using accuracy (how many true rules) and coverage (how many true encompassing rules) to select the best quality rules. The integration of expert rules involved receiving input from the experts and placing the same restrictions on their rules. Finally, the rules were used for anomaly prediction of semiconductors.

Steenwinckel et al. [91] produce a framework for anomaly detection and root cause analysis. They introduce Fused-AI Interpretable Anomaly Generation System (FLAGS) which combines knowledge-driven and data-driven techniques to gain the benefits of both and negate the detriments. Their framework is made of 3 parts: anomaly detection, fault recognition and root cause analysis to create semantic and interpretable anomalies and faults, a dashboard to capture user feedback, and an optimization of the anomaly detection and root cause analysis through fused AI and user feedback. This is done by creating a semantic database of rules through knowledge graphs and Matrix Profile, displaying the rules for user feedback and utilizing the feedback to optimize the models. They showed high accuracy when applied to a train dataset, and they illustrated the adaptability of their method by utilizing input given from the train operators.

3) INTERPRETABLE FILTERS

Interpretable filters are a concept that brings specific waveforms to a CNN architecture as a way of showing what signals are being learned. As explained in Ravanelli and Bengio [171], the first layer of a CNN appears to be important for waveform-based CNNs. In using these interpretable filters that take the form of common waveforms, one can begin to understand the behavior of the CNN if one understands the behavior of the waveform.

Li et al. [45] aimed to improve CNN-based methods for PHM by addressing the black box problem. They proposed the Continuous Wavelet Convolution (CWC) layer which is designed to make the first layer of a CNN interpretable. It does this by using a library of filters that have physical meanings which are convolved on the input signal. These convolutions can be traversed along the series and projected into a two-dimensional time and scale dimension. Its performance was compared with a CNN with different wavelets, and their findings were two-fold. Firstly, the performance of the CNN with a CWC layer showed better performance than a CNN without. Lastly, the CWC learned a well-defined waveform while the one without learned what looked to be a noisy and uninterpretable representation.

Li et al. [45] built on their previous work by examining compound faults. They designed an interpretable framework called wavelet capsule network (WavCapsNet) which utilizes backward tracking. This network has 1) interpretable meaning from the wavelet kernel convolutional layer, 2) capsule layers that allow decoupling of the compound fault, and 3) backward tracking which helps interpret output by focusing on the relationships between the features and health conditions. Not only was their framework able to achieve high accuracy on all conditions, including compound faults, but also they showed that the backward tracking method can decouple the capsule layers effectively.

Ben et al. [49] proposed a new architecture, SincNet, that trains directly on the raw vibration signals to diagnose bearing faults. Their architecture utilized interpretable digital filters for CNN architectures. They reduced the number of trainable parameters and extracted meaningful representations by having the predefined functions serve as the convolution. When comparing the performance to a CNN, the SincNet had a higher precision and reached convergence faster.

4) PHYSICAL CONSTRAINTS

Physical constraints are used to bring real-life limitations to the data-driven models. This can be in the form of mapping the input and output of the architectures to physical components, or more commonly, utilizing known physics information or equations about the real-life system in the architecture of their model in some way.

The methods of applying physical constraints can be seen in different forms, namely model-based approaches and physics-informed approaches, which need to be differentiated. Model-based approaches are created to model a system without the training of a network with the data provided, separate from data-based models [143]. These model-based approaches have physical constraints as they have to model the mathematical properties of the system. Physics-informed models aim to combine model-based and data-driven approaches by attaching the mathematical properties of the system to the data in data-driven approaches [102].

Tod et al. [143] implemented a first-principle model-based approach to assess the health of solenoid operated valves. Compared to other first-principle models, their improved model takes other degradation effects into account, namely shading ring degradation and mechanical wear. The method extracts three condition indicators which allows them to detect problematic signals that can be directly mapped to physical components through their model.

Wang et al. [55] performed fault diagnostics of wind turbines. Their method was an online method that detected issues with bearings. Coupled with equations that represent the physical aspects of the bearings, they detected issues surrounding clearance of the bearings with high interpretability. Their interpretation specifically showed the different frequencies around the physical features of the bearings.

Xu et al. [102] propose the physics-constraint variational neural network (PCVNN) as applied to external gear pumps. The PCVNN is physics-informed asymmetric autoencoder where the encoder is a stacked CNN, BiLSTM, Attention network while the decoder is a generative physical model. This would allow for an NN to learn the data, and it would allow the physical model to represent the learned patterns in a way that is consistent with the physics of the problem.

5) GRAPH ATTENTION NETWORKS (GATS)

GATs were introduced by Velivckovic et al. [172] as a way of combining self-attention layers with graph-structured data. This is done by applying attention layers where nodes can attend whole neighborhoods of previous graph nodes. While this comes with many benefits, the main two come from the benefits that other architectures gain from attention mechanisms and the retraction of needing prior knowledge of the graph structure.

Liu et al. [88] designed a framework for fault detection based around the Graph Convolutional Network and Graph Attention Networks. They propose the Causal-GAT. Causal-GAT is comprised of two parts: causal graph construction

TABLE 4. Examples of articles from sample population.

Title	Objective	Contribution
Impact of Interdependencies: Multi-Component System Perspective Toward Predictive Maintenance Based on Machine Learning and XAI [120]	Perform predictive maintenance by modeling interdependencies and test their importance	Showed with statistical significance that interdependency modeling increases performance and understandability of a model
Explainable and Interpretable AI-Assisted Remaining Useful Life Estimation for Aeroengines [35]	Compute RUL of the CMAPSS turbofan dataset with LIME explaining the performance	Showed that LIME performed poorly when applied to segments with no faults but performed well when labeling features with failing sequences
Explainability-driven Model Improvement for SOH Estimation of Lithium-ion Battery [109]	Perform predictive maintenance by embedding explanations into the training loop	Introduced the idea of explainability-driven training for predictive maintenance
Online Anomaly Explanation: A Case Study on Predictive Maintenance [65]	Apply XAI methods to the online learning process	Showed that local and global explanations could be added into the online learning paradigm
Explaining a Random Forest with the Difference of Two ARIMA Models in an Industrial Fault Detection Scenario [82]	Utilize two ARIMA surrogate models to explain the capabilities of a random forest model	Introduced a method of sandwiching a model between two surrogates to show where a model fails to perform well
Edge Intelligence-based Proposal for Onboard Catenary Stagger Amplitude Diagnosis [90]	Test XAI on edge computing for fault diagnosis	Provided a method of performing XAI in an edge computing example coupled with AutoML libraries
Explainable AI Algorithms for Vibration Data-based Fault Detection: Use Case-adapted Methods and Critical Evaluation [44]	Discover the plausibility of XAI methods explaining the output of CNN architectures	LRP showed non-distinguishable features, LIME showed unimportant features, and GradCAM showed the important features
On the Soundness of XAI in Prognostics and Health Management (PHM) [37]	Compare different XAI methods for the CMAPSS and lithium-ion battery dataset	Showed different metrics for comparing explanations generated by different XAI methods and showed GradCAM to perform the best on CNN architectures
Interpreting Remaining Useful Life Estimations Combining Explainable Artificial Intelligence and Domain Knowledge in Industrial Machinery [51]	Perform RUL of bushings through multiple different models and explanatory methods	Showed the importance of applying global and local explanations to interpret performances of models from all aspects
Evaluating Explainable Artificial Intelligence Tools for Hard Disk Drive Predictive Maintenance [38]	Analyze the effectiveness of explainability methods for recurrent neural network based models for RUL prediction	Utilized three metrics to compare explanations from LIME and SHAP and showed where each of them shine over the others
Automatic and Interpretable Predictive Maintenance System [66]	Aimed to integrate explainability into an AutoML environment	Defined a workflow for an automatic explainable predictive maintenance system
DTCEncoder: A Swiss Army Knife Architecture for DTC Exploration, Prediction, Search and Model interpretation [64]	Perform fault detection by classifying DTCs	Designed the DTCEncoder that utilizes an attention mechanism to provide an interpretable latent space as to why the a DTC is output
Deep Multi-Instance Contrastive Learning with Dual Attention for Anomaly Precursor Detection [133]	Perform anomaly detection and anomaly precursor detection	Performed anomaly precursor detection through multi-instance learning with verified explanations through domain experts
A Type-2 Fuzzy Based Explainable AI System for Predictive Maintenance Within the Water Pumping Industry [141]	Utilize an evolutionary algorithm to optimize their fuzzy logic system for fault prediction	Used a type 2 fuzzy logic system and evolutionary optimization to generate fuzzy rules for fault prediction
Waveletkernelnet: An Interpretable Deep Neural Network for Industrial Intelligent Diagnosis [45]	Improve CNN-based methods for PHM	Designed the Continuous Wavelet Convolution to add physical interpretations to the first layer of CNN architectures
Restricted Sparse Networks for Rolling Bearing Fault Diagnosis, [46]	Perform fault detection using a sparse network	Explored the Restricted-Sparse Frequency Domain Space and used the transform into this space to train a two-layer network that performs equal to a CNN-LSTM
Interpretable and Steerable Sequence Learning via Prototypes [62]	Construct a deep learning model with built-in interpretability for fault diagnosis via DTCs	Introduced Prototype Sequence Network (ProSeNet) which uses prototype similarity in the training of the network and justified the interpretability of their approach via a user study on Amazon MTurk
Causal and Interpretable Rules for Time Series Analysis [146]	Perform predictive maintenance while utilizing causal rules for explanations	Designed Case-crossover APriori algorithm for predictive maintenance which showed both higher performance occurs when having rules that are additive and subtractive to an output

and DC-Attention for extracting features and detection. The causal graph construction uses causal discovery methods and/or prior expertise to encode monitoring variables into a directed acyclic graph. The Disentangled Causal Attention (DC-Attention) aggregates the causal variables for generating representations of the effect variables. The DC-Attention outputs the system status (faulty or not faulty). They then utilize a custom loss function that calculates the distance between the current support of representations and its theoretically disentangled support.

6) PROTOTYPE LEARNING

Prototype learning, as described by Ming et al. [62], is a form of case-based reasoning that determines the output of an input by comparison to a representative example. Determining the best prototypes is a problem itself, but the interpretability it bring is apparent. The output of a specified input would be similar to its most similar prototype's output; therefore, the reason that the input data has a certain output is due to the output of a very similar piece of data. This brings interpretability via comparison to the prototype.

Ming et al. [62] used the concept of prototype learning to construct a deep learning model with built-in interpretability. They introduced the prototype sequence network (ProSeNet) for a multi-class classification problem of fault diagnosis via diagnostic trouble codes. The model consists of a sequence encoder that is based on a recurrent architecture. The hidden state is fed into a prototype layer that determines how similar the hidden state is to prototypes in the form of a similarity vector. The network then outputs a prediction probability for the different classes based on the similarity vector. Interpretability can be conceived via the prototypes that are most similar to the input. They justified the interpretability of their model by using Amazon MTurk and surveying the users about the interpretability. They also studied how the input of human knowledge would affect the interpretability. They showed that including the human feedback improved the interpretability of their network in a post-study of different Amazon MTurk users.

7) SIGNAL TEMPORAL LOGIC (STL)

Introduced by Maler and Nichovic [173], STL as a type of temporal logic that is used for *dense-time real-valued signals*. STL is defined as predicates over atomic propositions. These STL rules are formed by applying Boolean filters for these atomic propositions that transforms a signal into a Boolean signal. This involves considering: the filter that is being applied, the length of the signal, the sampling of the signal and any additional desired samples. We refer the reader to Maler and Nichovic [173] for an example.

Chen et al. [136] performed fault diagnosis on a furnace using internet-of-things, reinforcement learning, and signal temporal logic. Their algorithm takes in the STL grammar and labeled input data, and it outputs an optimal STL formula. The agent chooses a formula from the agenda and adds it to a chart based on the current policy. The evaluator evaluates the performance of the formula on the input. The learner updates the policy function according to the performance. The agenda is updated based on the formulas in the chart. They utilize an MDP to construct the agenda-based formulas while the reinforcement learning solves the problem. They apply their method to multiple faults demonstrating good robustness results, fast runtimes, and statistically significant performances.

8) RULE-BASED INTERPRETATIONS

Similar to rule-based explainers presented in V-A, rule-based interpretations involve utilizing rules that are learned from the data. Unlike the rule-based explainers, rule-based interpretations remove the black-box from the problem. This allows the rules to be directly learn from the information as opposed to learning from the black-box model and the data.

Dhaou et al. [146] proposed a novel approach that combines case-crossover research design with Apriori data mining. This combination resulted in the Case-crossover APriori (CAP) algorithm for association and causal rules explanation. The case-crossover design describes the way of setting up the problem. They ignored the group of data where nothing goes wrong, and they focused on the subjects that have the class change. In the case of predictive maintenance, a class change would be from healthy to failure data. The case-crossover design looks at the period prior to class change as the control group, and it looks at moments before the class change as the case period. These data points are combined with Association Rule Mining APriori to extract causal rules. These causal rules can be both additive (predictive of truth) and subtractive (predictive of falsehood). Their results show that both additive and subtractive rules help with performance, and they show their algorithm to outperform random forest on the same problem.

C. STATISTICS

Statistics in a wide-area with many applications that allow us to evaluate our hypotheses in a meaningful and common way [174]. These methods allow us to compare data distributions and more using tried and true ways that have been around since no later than 1900, in terms of Pearson's chi-squared test [175].These methods utilize actual tests from the realm of statistics in Section VI-C1, or they utilize methods that build upon these methods such as Hidden Markov Models and Generalized Additive Models in Section VI-C2 and Section VI-C4 respectively.

1) STATISTICAL METHODS

Statistical methods are used for explaining by analyzing different features along different classes using statistical tests, such as Student's t-test [176], Pearson's chi-squared test [175], etc.

Yao et al. [41] proposed a framework with interpretable and automatic approaches that consisted of solely statistical processing. Their method proposed kurtosis-energy metric to define key sub-bands, a new health index of these sub-bands, a joint statistical alarm and fault identification strategy. Additionally, they proposed a health phase segmentation strategy for health phase assessment and degradation pattern analysis. This method involved analyzing the data on the time-frequency domain and suppressing the disturbing components such as noise. This analysis was able to help form the sub-bands for monitoring the current state. If it fails statistical tests, then an anomaly is detected. They tested their method on the PHM 2012 rolling bearing dataset, and they reported very low false positives.

As there are typically no sensors within wastewater pipe blockages, Castle et al. [148] propose a statistical approach with the historical data. To perform reliability analysis, synonymous to anomaly detection, of the wastewater pipes, they apply two different statistical methods: the frequentist Cox Proportional Hazards Model (Cox PHM) and Bayesian Weibull Proportional Hazards Model (Weibull PHM) The Cox PHM [177] is a statistical model that compares failure rates between units while providing information about what variables influence failure events. The Weibull PHM utilizes a Lasso regression to define the priors calculations, and they use a Markov Chain Monte Carlo procedure to update the posterior distributions of their covariates. They were able to use these approaches to find statistically significant differences in water pipes with features that indicate blockages and those that do not. With these statistical results, they were able to easily interpret these results in addition to their reliability analysis.

2) HIDDEN MARKOV MODEL (HMM)

HMMs were introduced by Baum and Petrie [178] and can be described as a statistical state-space algorithm [29]. HMMs represent the learning as a statistical process that transitions between states, and HMMs represent the output as separate states that extend from the transitional states. HMMs, as a statistical process, can discern hidden states from the data that may not be readily apparent. They are also capable of learning combinations of sensor data, leveraging confounding variables, and executing dimensionality reduction to simplify the complexity of the data. [80].

Abbas et al. [80] combined the input-output HMM with reinforcement learning to make interpretable maintenance decisions. Their hierarchical method consisted of two steps. The input-output HMM filters the data and detects failure states. Once the failure state was detected, the deep reinforcement learning agent learned a policy for maintenance based on the failures. The first challenge of this approach involves representing predictive maintenance as a reinforcement learning problem. This is done by representing the potential actions as hold, repair, or replace, creating a reward function based on holding, early replacement and replacement after failure, and measuring the cost based on these reward functions. The HMM is used for interpreting the output of their model by observing the features that led the model into detecting a failure state.

3) GAUSSIAN MIXTURE MODEL (GMM)

As described by Reynolds [179], Gaussian mixture models (GMMs) is a probability density function designed as a weighted sum of Gaussian component densities. The component densities are created using the mean vector and covariance matrix of the data while the mixture weights are estimated. GMMs are commonly used due to their capability of representing information via a discrete set of Gaussian functions to improve modeling of larger distributions. These models can be labeled as interpretable as the models directly represent the distributions of the features. These models can then be directly used to explain the features.

Csalodi et al. [108] performed survival analysis via a Weibull distribution by representing the operation signals as a Gaussian mixture models and the parameters of the Weibull model via clustering. Specifically, their method used an expectation-maximization algorithm which consists of two parts. The expectation step determined the probability that a data point belongs to any cluster given the survival time and parameters while assuming the clustering is correct. The maximization step updated the parameters for the Gaussian mixture models and the Weibell distribution to better represent the data. When applying their method to lithium-ion batteries, they represented distributions of unhealthy batteries quite accurately while healthy batteries were less well-represented. This occurred due to the large category of healthy data which was harder to represent in one small model while the unhealthy data could be easily represented when isolated.

4) GENERALIZED ADDITIVE MODEL (GAM)

Introduced by Hastie and Tibshirani [180], GAMs are a way of estimating a function by summing a list of nonlinear functions in an iterative manner as to become better with accurate local models as opposed to an overarching global model. These local models are smoothed using a series of smoothing functions. Additionally, these local models are independent of one another as they are trained using single features. These local models allow for interpretability as well as importance related to their impacts on the outcome of the GAM.

Yang et al. [43] introduced the Noise-Aware Sparse Gaussian Process as a way of solving the scalability and noise sensitivity issues of normal Guassian Processes. Based on their NASGP algorithm, they developed an interpretable GAM that uses additive kernels and individual features. They applied their method to the IEEE PHM 2012 data challenge in forms of RUL prediction and fault diagnosis. Their method performed well in comparison to other methods and allowed a level of interpretability.

5) MAHALANOBIS-TAGUCHI SYSTEM (MTS)

MTS was introduced by Taguchi and Jugulum [181] as a diagnosis and forecasting method. This method bases its discriminative power on the Mahalanobis distance calculation; this method cannot feasibly work if the classes cannot be distinguished this way. The feature space is reduced via orthogonal arrays and signal-to-noise ratios. The orthogonal array contains different subsets of the features. The signal-to-noise ratio measures the abnormality of the feature. Finally, the Mahalanobis distance is maximized by only including the features whose signal-to-noise ratio increases the distance. This maximized distance can be seen as the reason for a diagnosis, which is determined by the features that are used to calculate the distance.

Scott et al. [125] introduced use of the Mahalanobis-Taguchi system for fault detection. MTS utilizes Mahalanobis distance, orthogonal arrays, and signal-to-noise ratios for multivariate diagnosis and prediction. The Mahalanobis space represents the stable operations andyields the difference of an observation from stable. The orthogonal arrays and signal-to-noise ratio is used to diagnose or identify variables responsible for the fault. This method was able to detect roughly 75% of the faults tested.

6) EXPLAINABLE BOOSTING MACHINE (EBM)

EBM were introduced by Nori et al. [182] as a *glassbox model*, another term for interpretable model, with similar accuracy to that of state-of-the-art blackbox algorithms. EBM is a type of generalized additive model that learns each feature's function using techniques such as bagging. Additionally, it can detect interactions between features and include those pairs of terms by learning functions of combinations of features. Because of its nature as an additive model, the features can be explained by their impact on the outcome. Jakubowski et al. [76] utilized this method, but they focused more on the XAI approaches. Their work is described in Section V-C.

D. OTHER METHODS

The methods found here do not fit cleanly within the other categories in interpretable machine learning. These methods consist of the ever popular attention mechanism, Section VI-D1, digital twins, Section VI-D2, and k-Nearest Neighbors, Section VI-D3.

1) ATTENTION

Attention was introduced by Vaswani et al. [183] as a method of natural language processing. This attention module gets extended to introduce the transformer architecture that has led to many famous models such as GPT. The weights from the attention modules can be visualized to allow interpretation of the aspects the architecture is focusing.

Xia et al. [58] and Hafeez et al. [64] tackled interpretable fault diagnosis in two separate ways. Xia et al. looked at hierarchical attention by grouping the features by systems and subsystems. They utilized BiLSTM encoders with attention to obtain important features where the attention components added interpretability. Hafeez et al. created an architecture known as the DTCEncoder to learn low level representations of multivariate sequences with attention. It utilized the Diagnostic Trouble Codes (DTC) commonly found in predictive maintenance problems as a class label for fault diagnosis. Dense layers were used to translate the encoded latent space from DTCEncoder into a probability distribution for the different DTCs. The latent space was learned using attention mechanisms and could be used to add interpretability of why the network output the DTC.

For interpretable fault prediction, Wang et al. [56] proposed a two-stage method based on anomaly detection and anomaly accumulation. The anomaly detection module was made using a CT-GAN to train a discriminator on limited data, i.e., faults. The anomaly scores from the CT-GAN were fed into the anomaly accumulation module based on an Attention-LSTM. This modeled the temporal dependencies of the anomaly scores while the attention mechanism was used to give importance to different anomalies at different time steps. Their model outperforms models such as SVM and LSTM on prediction and DTW on classification.

Xu et al. [133] was not only interested in anomaly detection, but also anomaly precursor detection, early symptoms of an upcoming anomaly. They argued that detecting precursors is useful for early prediction of anomalies to better understand when and what kind of anomaly will occur. They proposed Multi-instance Contrastive learning approach with Dual Attention (MCDA) to tackle the problem of anomaly precursor detection. MCDA combined multi-instance learning and tensorized LSTM with time-dependent correlation to learn the precursors. Additionally, the dual-attention module produced their interpretable results. This approach had high accuracy results, and their attention mechanism provided variables which are explanatory for the results. Importantly, they verified these explanations with domain experts.

2) DIGITAL TWIN

Digital twins originated in 2002 as described by Grieves and Vickers [184] as a way of creating a digital construct that describes a physical system. Moreover, digital twins consist of two systems: a physical system that is represented by the asset and a digital system that holds the information about the physical system. Using digital twins, one can observe the performance of the physical system without having the physically observe the asset.

Mahmoodian et al. [144] proposed the use of a digital twin to monitor the infrastructure of a conveyor. Their digital twin consists of taking in real data from different sensors and simulating the data. This data is compared to the real time data to ensure the data is consistent. Their digital twin can display the different information as well as receive input from the users to rate the explanations given. If it is seen as not valid, the digital twin can run simulations surrounding that data to increase its accuracy.

3) K-NEAREST NEIGHBORS (KNN)

Originally introduced in 1951 by Fix and Hodges [185], kNN is a supervised learning algorithm that is based on grouping input data with the k most similar other pieces of input data. It represents the input data as a large feature space. The output of some input data is represented by its place in the feature space in relation to the k closest other data points. Small k values lead to less consideration for the output value of the input data; however, it also leads to a more specific output. Larger k values lead to considering more values when determining the output; however, too large k values will make the output less meaningful.

Konovalenko et al. [145] used a modified kNN algorithm for generating decision support of temperature alarms. They tackled three problems associated with kNN: (1) the difficulty associated with sparse regions; (2) the blindness to class boundaries leading to misclassifications; and (3) sensitivity to class overlap. These problems were addressed by adding principles of local similarity and neighborhood homogeneity. Local similarity refers to the idea that a new data is closer to training samples with the same class label. Neighborhood homogeneity is the idea that new data falls into a neighborhood where the class label represents the majority. This method is interpretable through its ability to separate classes of data on a small dimensional graph.

VII. CHALLENGES AND RESEARCH DIRECTIONS OF EXPLAINABLE PREDICTIVE MAINTENANCE

XAI and iML have been successfully utilized in predictive maintenance on many accounts. Researchers have shown that these methods can add to a prediction in a way that can be used for root cause analysis, validation of faults, etc. The main focus of much of the research focuses on adding explainability to a complex and unexplainable problem. While an important aspect of this field of study, there are multiple facets to the problem that generally go under-represented.

A. PURPOSE OF THE EXPLANATIONS

All explanations serve one overarching purpose: produce reasons that make the model's functioning understandable. This information transfer has taken form in visualizations of data distributions, visualizations of feature importance graphs, predictive rules, etc; however, the information is not specific to a target audience. To echo Neupane et al. [15], *"explanations are not being designed around stakeholders"*. Not only are the explanations not being designed for stakeholders, but also many explanations do not have a target audience outside of the implicit audience of the model's designer.

Barredo Arrieta et al. [186] provides a list of potential audiences XAI can target. While they go into more detail, some potential target audiences, especially for predictive maintenance, could be the data scientists and developers creating the predictive system, the project managers and stakeholders in the project, or even the mechanics working on the physical systems. These different people may need different types of explanations ranging from more explanations relating to the physical and time domains to higher level abstract information.

B. EVALUATION OF THE EXPLANATIONS

In the literature presented above, there are over ten different evaluation metrics for the performance of the machine learning algorithms, including RMSE, MAPE, FP, etc. This shows that the field has collectively come to an agreement on how we should measure performance in a meaningful way. The evaluation of the explanations has not received the same attention as the performance of the algorithm even though work has been done in defining these different metrics, some of which are seen in Table 5.

TABLE 5. Explanation evaluation metrics from [187], [188], [189], [190], and [191].

Metrics	Viewpoint	Description
D	Objective	Difference between the model's performance and the explanation's performance
R	Objective	Number of rules in explanations
F	Objective	Number of features in explanation
S	Objective	The stability of the explanation
Sensitivity	Objective	Measure the degree in which explanations are affected by small changes to the test points
Robustness	Objective	Similar inputs should have similar explanations
Monotonicity	Objective	Feature attributions should be monotonic; otherwise, the correct importance is not calculated
Explanation correctness	Objective	Sensitivity and Fidelity
Fidelity	Objective	Explanations correctly describe the model; features and their attribution are correlated
Generalizability	Objective	How much one explanation informs about others
Trust	Subjective	Measured through user questionnaires
Effectiveness	Subjective	Measures the usefulness of the explanations
Satisfaction	Subjective	Ease of use

Miller [187] provides one of the most in-depth descriptions of various people's needs regarding explanations. Miller has provided many theoretical representations for explanation including scientific explanations and data explanations. They also provide much more information including levels of explanation that could be applicable to different types of users, structures of explanations that could impact the power of the explanations, and more.

Coroama and Groza [188] present 37 different metrics for measuring the effectiveness of an explanation. The methods range from objective to subjective types. Each method includes the property it measures and whether there is a systemic implementation.

Sisk et al. [189] present the case for human-centered evaluations and objective evaluations for explainable methods. Their human-centered evaluations aim at partitioning the users based on their wants from explainable systems. The objective metrics provided involve many aspects of the explanations including number of rules and number of features.

Kadir et al. [190] propose a taxonomy of XAI evaluations as they appeared in the literature. They identified 28 different metrics through their literature search. These metrics are broken down into a taxonomy of how the analysis is performed. An example would be sensitivity analysis for local explanations. Sensitivity analysis is broken down into the removal of features and the addition of features. Each of these categories then includes many methods that were used.

Hoffman et al. [191] express the importance of high quality explanations in XAI. If explanations are received well and are valid, a user would be better equipped to trust and use a system that employes the XAI process. This allows for

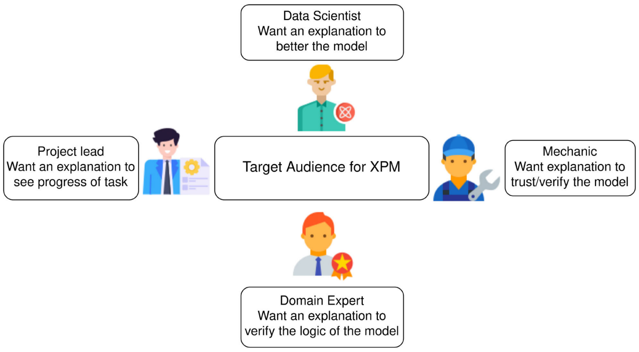

Data Scientist
Want an explanation to better the model
Project lead
Want an explanation to see progress of task
Target Audience for XPM
Mechanic
Want explanation to trust/verify the model
Domain Expert
Want an explanation to verify the logic of the model

FIGURE 13. Potential audiences of explainable predictive maintenance. Icons taken from[1].

multiple areas of evaluations including the *goodness* of the explanation, the *satisfaction* the explanations provided to the users, the *comprehension* of the user, the *curiosity* that motivates the user, the *trust and reliance* the user has with the AI, and the *performance* of the human-XAI system. They provide methods for measuring these metrics that are readily available.

Lastly, the addition of these metrics only add to the field of XPM. When we compare two models based on their RMSE, we can come to concrete conclusions about the use of these two models; one of the two models will be superior. The addition of explanation metrics, even ones as simple as the objective measures found in Table 5, gives us as researchers another axis to compare our models. This may show that certain explainable methods perform best with certain models, and it may show us that certain explainable methods are better applied to certain cases, especially when audiences may be different.

C. ADDITION OF HUMAN INVOLVEMENT

The target audience of an explainable system is a human subject whether a data scientist, a stakeholder, an engineer, or other. Addressing the needs of different types of users of an explainable system is an important area of research that is currently lacking. As seen in Fig. 13, different people on the same task have different goals and desires from predictive maintenance. While compensating for these differences would be difficult, we suggest a way to accomplish this, together with the resulting benefits.

First, a target audience for the explainable system should be identified, ensuring that a sample population of statistically significant size is used. This would allow the researchers and developers to gather specific user requirements. Second, one should present the information to this sample population as a way of evaluating the explanations generated by a proposed method. This would also allow the researchers to apply evaluation metrics such as effectiveness and trust. Finally, explanation methods should be altered to the user's needs as opposed to an assumed need of the user. This would bring many benefits to the XAI field as a whole. These include: making more quality metrics available, allowing researchers to discern which information is more or less useful, and bringing more attention to customizable explanations via the type of user. These would push the field of XAI forward as well as push the field of predictive maintenance forward towards a human-AI teaming environment.

D. STUDY LIMITATIONS

This study focuses on a small amount of potential XAI and iML literature. While this survey reflects the work done as applied to predictive maintenance, it does not reflect many of the applied XAI and iML algorithms that exist. It also does not reflect all of the applicable ML algorithms developed within the context of predictive maintenance. While we do not see this as a detriment to the article presented, we do note that there are a number of popular methods of which the reader may be aware that are not present.

VIII. CONCLUSION

Over the last decade, predictive maintenance has occupied a considerable presence in the field of machine learning research. As we move towards complex mechanical systems with interdependencies that we struggle to explain, predictive maintenance allows us to break down the mysticality of what could potentially go wrong in the system. Many of these approaches move us closer to understanding the system while building a new system that we need to comprehend. Explainable predictive maintenance and interpretable predictive maintenance aim at breaking down these new walls to bring us closer to a clear understanding of the mechanical system.

In this review, we provided a wide range of methods that are being used to tackle the problem of explainability. These methods are broken down in XAI and iML approaches. In our writing, XAI was broken-up into model-agnostic approaches like SHAP, LIME and LRP, and model-specific approaches like GradCAM and DIFFI. iML approaches all apply different methods of applying inherently interpretable models to the problem of predictive maintenance.

Additionally, we have brought the attention to over 40 methods that can be utilized to bring explainability and interpretability to predictive maintenance tasks. We cannot make statements about the quality of the explanations generated by these differnet methods. As can be seen in our review, many methods have been applied to the same datasets where very few were used in a comparative analysis. This leaves us lacking in a recommendation for an overarching best approach to solve the problem of explainability in predictive maintenance. We provide the different methods coupled with their datasets, so for researchers beginning with turbofans for example, beginning with Shapley Additive Explanations or Rule-based approaches would be the best place to start, as seen in Table 1.

Our systematic review showed some weak points in the field that can be addressed. Namely, there is a lack of utilization of metrics of explanations in predictive maintenance. The

[1] https://icons8.com/

field of XAI has shown a number of metrics that do not even need to show the explanations to the target audience of the explainable systems. We provided a list of potential metrics found in the literature that can be applied to this domain.

Lastly, we provided a short description of how humans can be brought into the evaluation of explainable and interpretable methods. After defining the target audience, researchers can gather a statistically significant sized sample of that audience. Providing the explanations to that sample would give feedback and allow the field to push towards human-specified explanations.

ACKNOWLEDGMENT

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Army ERDC or the U.S. DoD. The authors would like to thank Mississippi State University's Predictive Analytics and Technology Integration (PATENT) Laboratory for its support.

REFERENCES

[1] S. B. Ramezani, L. Cummins, B. Killen, R. Carley, A. Amirlatifi, S. Rahimi, M. Seale, and L. Bian, ''Scalability, explainability and performance of data-driven algorithms in predicting the remaining useful life: A comprehensive review,'' *IEEE Access*, vol. 11, pp. 41741–41769, 2023.

[2] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, and L. Wang, ''Industry 5.0: Prospect and retrospect,'' *J. Manuf. Syst.*, vol. 65, pp. 279–295, Oct. 2022.

[3] F. Longo, A. Padovano, and S. Umbrello, ''Value-oriented and ethical technology engineering in Industry 5.0: A human-centric perspective for the design of the factory of the future,'' *Appl. Sci.*, vol. 10, no. 12, p. 4182, Jun. 2020.

[4] S. Nahavandi, ''Industry 5.0—A human-centric solution,'' *Sustainability*, vol. 11, no. 16, p. 4371, Aug. 2019.

[5] A. Lavopa and M. Delera. (2021). *What is the Fourth Industrial Revolution? | Industrial Analytics Platform*. [Online]. Available: https://iap.unido.org/articles/what-fourth-industrial-revolution

[6] L. Cummins, B. Killen, K. Thomas, P. Barrett, S. Rahimi, and M. Seale, ''Deep learning approaches to remaining useful life prediction: A survey,'' in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–9.

[7] T. Speith, ''A review of taxonomies of explainable artificial intelligence (XAI) methods,'' in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Jun. 2022, pp. 2239–2250.

[8] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, ''Evaluating the quality of machine learning explanations: A survey on methods and metrics,'' *Electronics*, vol. 10, no. 5, p. 593, Mar. 2021.

[9] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert, ''From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI,'' *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–42, Dec. 2023.

[10] Y. Rong, T. Leemann, T.-t. Nguyen, L. Fiedler, P. Qian, V. Unhelkar, T. Seidel, G. Kasneci, and E. Kasneci, ''Towards human-centered explainable AI: A survey of user studies for model explanations,'' 2022, *arXiv:2210.11584*.

[11] J. Sharma, M. L. Mittal, and G. Soni, ''Condition-based maintenance using machine learning and role of interpretability: A review,'' *Int. J. Syst. Assurance Eng. Manage.*, pp. 1–16, Dec. 2022.

[12] R. Marcinkevič s and J. E. Vogt, ''Interpretable and explainable machine learning: A methods-centric overview with concrete examples,'' *WIREs Data Mining Knowl. Discovery*, vol. 13, no. 3, p. e1493, May 2023.

[13] M.-A. Clinciu and H. Hastie, ''A survey of explainable AI terminology,'' in *Proc. 1st Workshop Interact. Natural Language Technol. Explainable Artif. Intell.*, 2019, pp. 8–13.

[14] S. S. Kim, E. A. Watkins, O. Russakovsky, R. Fong, and A. Monroy-Hernández, '''Help me help the AI': Understanding how explainability can support human-AI interaction,'' in *Proc. Conf. Human Factors Comput. Syst.*, 2023, pp. 1–17.

[15] S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, and M. Seale, ''Explainable intrusion detection systems (X-IDS): A survey of current methods, challenges, and opportunities,'' *IEEE Access*, vol. 10, pp. 112392–112415, 2022.

[16] A. K. M. Nor, S. R. Pedapati, M. Muhammad, and V. Leiva, ''Overview of explainable artificial intelligence for prognostic and health management of industrial assets based on preferred reporting items for systematic reviews and meta-analyses,'' *Sensors*, vol. 21, no. 23, p. 8020, Dec. 2021.

[17] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, ''Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence,'' *Inf. Fusion*, vol. 99, Nov. 2023, Art. no. 101805.

[18] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, ''Explainable artificial intelligence (XAI) on TimeSeries data: A survey,'' 2021, *arXiv:2104.00950*.

[19] K. Sokol and P. Flach, ''Explainability is in the mind of the beholder: Establishing the foundations of explainable artificial intelligence,'' 2021, *arXiv:2112.14466*.

[20] T. A. J. Schoonderwoerd, W. Jorritsma, M. A. Neerincx, and K. van den Bosch, ''Human-centered XAI: Developing design patterns for explanations of clinical decision support systems,'' *Int. J. Hum.-Comput. Stud.*, vol. 154, Oct. 2021, Art. no. 102684.

[21] P. Lopes, E. Silva, C. Braga, T. Oliveira, and L. Rosado, ''XAI systems evaluation: A review of human and computer-centred methods,'' *Appl. Sci.*, vol. 12, no. 19, p. 9423, Sep. 2022.

[22] S. Vollert, M. Atzmueller, and A. Theissler, ''Interpretable machine learning: A brief survey from the predictive maintenance perspective,'' in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2021, pp. 01–08.

[23] S. M. Lundberg and S.-I. Lee, ''A unified approach to interpreting model predictions,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, '''Why should I trust you?' explaining the predictions of any classifier,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, ''Learning deep features for discriminative localization,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[26] C. Molnar, *Interpretable Machine Learning*. Morrisville, NC, USA: Lulu.com, 2020.

[27] S. B. Ramezani, L. Cummins, B. Killen, R. Carley, S. Rahimi, and M. Seale, ''Similarity based methods for faulty pattern detection in predictive maintenance,'' in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2021, pp. 207–213.

[28] Y. Wen, M. Fashiar Rahman, H. Xu, and T.-L.-B. Tseng, ''Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective,'' *Measurement*, vol. 187, Jan. 2022, Art. no. 110276.

[29] S. B. Ramezani, B. Killen, L. Cummins, S. Rahimi, A. Amirlatifi, and M. Seale, ''A survey of HMM-based algorithms in machinery fault prediction,'' in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–9.

[30] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, ''Prognostics and health management: A review on data driven approaches,'' *Math. Problems Eng.*, vol. 2015, pp. 1–17, Aug. 2015.

[31] M. J. Page et al., ''PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews,'' *BMJ*, vol. 372, p. n160, Mar. 2021.

[32] N. R. Haddaway, M. J. Page, C. C. Pritchard, and L. A. McGuinness, ''PRISMA2020: An R package and shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and open synthesis,'' *Campbell Systematic Rev.*, vol. 18, no. 2, p. e1230, Jun. 2022.

[33] P. Ding, M. Jia, and H. Wang, ''A dynamic structure-adaptive symbolic approach for slewing bearings' life prediction under variable working conditions,'' *Struct. Health Monitor.*, vol. 20, no. 1, pp. 273–302, Jan. 2021.

[34] G. Manco, E. Ritacco, P. Rullo, L. Gallucci, W. Astill, D. Kimber, and M. Antonelli, "Fault detection and explanation through big data analysis on sensor streams," *Expert Syst. Appl.*, vol. 87, pp. 141–156, Nov. 2017.

[35] G. Protopapadakis, A. Apostolidis, and A. I. Kalfas, "Explainable and interpretable AI-assisted remaining useful life estimation for aeroengines," in *Proc. ASME Turbo Expo., Turbomachinery Tech. Conf. Expo.*, Oct. 2022, Art. no. V002T05A002.

[36] T. Khan, K. Ahmad, J. Khan, I. Khan, and N. Ahmad, "An explainable regression framework for predicting remaining useful life of machines," in *Proc. 27th Int. Conf. Autom. Comput. (ICAC)*, Sep. 2022, pp. 1–6.

[37] D. Solís-Martín, J. Galán-Páez, and J. Borrego-Díaz, "On the soundness of XAI in prognostics and health management (PHM)," *Information*, vol. 14, no. 5, p. 256, Apr. 2023.

[38] 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, no. 7, pp. 7279–7314, Jul. 2023.

[39] P. Nectoux et al., "PRONOSTIA: An experimental platform for bearings accelerated degradation tests," in *Proc. IEEE Int. Conf. Prognostics Health Manag.*, Jun. 2012, pp. 1–8.

[40] H. Qiu, J. Lee, J. Lin, and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics," *J. Sound Vibrat.*, vol. 289, nos. 4–5, pp. 1066–1090, Feb. 2006.

[41] R. Yao, H. Jiang, C. Yang, H. Zhu, and C. Liu, "An integrated framework via key-spectrum entropy and statistical properties for bearing dynamic health monitoring and performance degradation assessment," *Mech. Syst. Signal Process.*, vol. 187, Mar. 2023, Art. no. 109955.

[42] L. C. Brito, G. A. Susto, J. N. Brito, and M. A. V. Duarte, "An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery," *Mech. Syst. Signal Process.*, vol. 163, Jan. 2022, Art. no. 108105.

[43] J. Yang, Z. Yue, and Y. Yuan, "Noise-aware sparse Gaussian processes and application to reliable industrial machinery health monitoring," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 5995–6005, Apr. 2023.

[44] O. Mey and D. Neufeld, "Explainable AI algorithms for vibration data-based fault detection: Use case-adadpted methods and critical evaluation," *Sensors*, vol. 22, no. 23, p. 9037, Nov. 2022.

[45] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, and R. X. Gao, "WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2302–2312, Apr. 2022.

[46] H. Pu, K. Zhang, and Y. An, "Restricted sparse networks for rolling bearing fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 1–11, Feb. 2023.

[47] G. Xin, Z. Li, L. Jia, Q. Zhong, H. Dong, N. Hamzaoui, and J. Antoni, "Fault diagnosis of wheelset bearings in high-speed trains using logarithmic short-time Fourier transform and modified self-calibrated residual network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7285–7295, Oct. 2022.

[48] L. C. Brito, G. A. Susto, J. N. Brito, and M. A. V. Duarte, "Fault diagnosis using eXplainable AI: A transfer learning-based approach for rotating machinery exploiting augmented synthetic data," *Expert Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120860.

[49] F. Ben Abid, M. Sallem, and A. Braham, "An end-to-end bearing fault diagnosis and severity assessment with interpretable deep learning," *J. Electr. Syst.*, vol. 18, no. 4, pp. 1–19, Jul. 2022.

[50] D. C. Sanakkayala, V. Varadarajan, N. Kumar, Karan, G. Soni, P. Kamat, S. Kumar, S. Patil, and K. Kotecha, "Explainable AI for bearing fault prognosis using deep learning techniques," *Micromachines*, vol. 13, no. 9, p. 1471, Sep. 2022.

[51] O. Serradilla, E. Zugasti, C. Cernuda, A. Aranburu, J. R. de Okariz, and U. Zurutuza, "Interpreting remaining useful life estimations combining explainable artificial intelligence and domain knowledge in industrial machinery," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2020, pp. 1–8.

[52] R. Kothamasu and S. H. Huang, "Adaptive Mamdani fuzzy model for condition-based maintenance," *Fuzzy Sets Syst.*, vol. 158, no. 24, pp. 2715–2733, Dec. 2007.

[53] E. Lughofer, P. Zorn, and E. Marth, "Transfer learning of fuzzy classifiers for optimized joint representation of simulated and measured data in anomaly detection of motor phase currents," *Appl. Soft Comput.*, vol. 124, Jul. 2022, Art. no. 109013.

[54] A. L. Alfeo, M. G. C. A. Cimino, and G. Vaglini, "Degradation stage classification via interpretable feature learning," *J. Manuf. Syst.*, vol. 62, pp. 972–983, Jan. 2022.

[55] J. Wang, M. Xu, C. Zhang, B. Huang, and F. Gu, "Online bearing clearance monitoring based on an accurate vibration analysis," *Energies*, vol. 13, no. 2, p. 389, Jan. 2020.

[56] W. Wang, Z. Peng, S. Wang, H. Li, M. Liu, L. Xue, and N. Zhang, "IFP-ADAC: A two-stage interpretable fault prediction model for multivariate time series," in *Proc. 22nd IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2021, pp. 29–38.

[57] C. Panda and T. R. Singh, "ML-based vehicle downtime reduction: A case of air compressor failure detection," *Eng. Appl. Artif. Intell.*, vol. 122, Jun. 2023, Art. no. 106031.

[58] S. Xia, X. Zhou, H. Shi, S. Li, and C. Xu, "A fault diagnosis method with multi-source data fusion based on hierarchical attention for AUV," *Ocean Eng.*, vol. 266, Dec. 2022, Art. no. 112595.

[59] Y. Fan, H. Sarmadi, and S. Nowaczyk, "Incorporating physics-based models into data driven approaches for air leak detection in city buses," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2022, pp. 438–450.

[60] W. Li, H. Lan, J. Chen, K. Feng, and R. Huang, "WavCapsNet: An interpretable intelligent compound fault diagnosis method by backward tracking," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[61] G. B. Jang and S. B. Cho, "Anomaly detection of 2.4 l diesel engine using one-class SVM with variational autoencoder," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2019, vol. 11, no. 1, pp. 1–17.

[62] Y. Ming, P. Xu, H. Qu, and L. Ren, "Interpretable and steerable sequence learning via prototypes," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 903–913.

[63] C. Oh, J. Moon, and J. Jeong, "Explainable process monitoring based on class activation map: Garbage in, garbage out," in *Communications in Computer and Information Science*. Cham, Switzerland: Springer, 2020, pp. 93–105.

[64] A. B. Hafeez, E. Alonso, and A. Riaz, "DTCEncoder: A Swiss army knife architecture for DTC exploration, prediction, search and model interpretation," in *Proc. 21st IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2022, pp. 519–524.

[65] R. P. Ribeiro, S. M. Mastelini, N. Davari, E. Aminian, B. Veloso, and J. Gama, "Online anomaly explanation: A case study on predictive maintenance," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2022, pp. 383–399.

[66] X. Li, Y. Sun, and W. Yu, *Automatic and Interpretable Predictive Maintenance System*. Warrendale, PA, USA: SAE, Apr. 2021.

[67] S. Voronov, D. Jung, and E. Frisk, "A forest-based algorithm for selecting informative variables using variable depth distribution," *Eng. Appl. Artif. Intell.*, vol. 97, Jan. 2021, Art. no. 104073.

[68] J.-H. Han, S.-U. Park, and S.-K. Hong, "A study on the effectiveness of current data in motor mechanical fault diagnosis using XAI," *J. Electr. Eng. Technol.*, vol. 17, no. 6, pp. 3329–3335, Aug. 2022.

[69] A. Saxena and K. Goebel, *Turbofan Engine Degradation Simulation Data Set*, vol. 18. Washington, DC, USA: NASA Ames prognostics Data Repository, 2008.

[70] A. Brunello, D. D. Monica, A. Montanari, N. Saccomanno, and A. Urgolo, "Monitors that learn from failures: Pairing STL and genetic programming," *IEEE Access*, vol. 11, pp. 57349–57364, 2023.

[71] Z. Wu, H. Luo, Y. Yang, P. Lv, X. Zhu, Y. Ji, and B. Wu, "K-PdM: KPI-oriented machinery deterioration estimation framework for predictive maintenance using cluster-based hidden Markov model," *IEEE Access*, vol. 6, pp. 41676–41687, 2018.

[72] J. Jakubowski, P. Stanisz, S. Bobek, and G. J. Nalepa, "Anomaly detection in asset degradation process using variational autoencoder and explanations," *Sensors*, vol. 22, no. 1, p. 291, Dec. 2021.

[73] A. Brunello, D. Della Monica, A. Montanari, and A. Urgolo, "Learning how to monitor: Pairing monitoring and learning for online system verification," in *Proc. OVERLAY*, 2020, pp. 83–88.

[74] N. Costa and L. Sánchez, "Variational encoding approach for interpretable assessment of remaining useful life estimation," *Rel. Eng. Syst. Saf.*, vol. 222, Jun. 2022, Art. no. 108353.

[75] M. Sayed-Mouchaweh and L. Rajaoarisoa, "Explainable decision support tool for IoT predictive maintenance within the context of Industry 4.0," in *Proc. 21st IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2022, pp. 1492–1497.

[76] J. Jakubowski, P. Stanisz, S. Bobek, and G. J. Nalepa, "Performance of explainable AI methods in asset failure prediction," in *Computational Science—ICCS*. Cham, Switzerland: Springer, 2022, pp. 472–485.

[77] E. Kononov, A. Klyuev, and M. Tashkinov, "Prediction of technical state of mechanical systems based on interpretive neural network model," *Sensors*, vol. 23, no. 4, p. 1892, Feb. 2023.

[78] T. Jing, P. Zheng, L. Xia, and T. Liu, "Transformer-based hierarchical latent space VAE for interpretable remaining useful life prediction," *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101781.

[79] K. Waghen and M.-S. Ouali, "A data-driven fault tree for a time causality analysis in an aging system," *Algorithms*, vol. 15, no. 6, p. 178, May 2022.

[80] A. N. Abbas, G. C. Chasparis, and J. D. Kelleher, "Interpretable input-output hidden Markov model-based deep reinforcement learning for the predictive maintenance of turbofan engines," in *Big Data Analytics and Knowledge Discovery*. Cham, Switzerland: Springer, 2022, pp. 133–148.

[81] J. Brito and R. Pederiva, "Using artificial intelligence tools to detect problems in induction motors," in *Proc. 1st Int. Conf. Soft Comput. Intell. Syst.*, vol. 1, 2002, pp. 1–6.

[82] A.-C. Glock, "Explaining a random forest with the difference of two ARIMA models in an industrial fault detection scenario," *Proc. Comput. Sci.*, vol. 180, pp. 476–481, Jul. 2021.

[83] S. Matzka, "Explainable artificial intelligence for predictive maintenance applications," in *Proc. 3rd Int. Conf. Artif. Intell. Industries (AI4I)*, Sep. 2020, pp. 69–74.

[84] A. Torcianti and S. Matzka, "Explainable artificial intelligence for predictive maintenance applications using a local surrogate model," in *Proc. 4th Int. Conf. Artif. Intell. Industries (AI4I)*, Sep. 2021, pp. 86–88.

[85] Y. Remil, A. Bendimerad, M. Plantevit, C. Robardet, and M. Kaytoue, "Interpretable summaries of black box incident triaging with subgroup discovery," in *Proc. IEEE 8th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2021, pp. 1–10.

[86] B. Ghasemkhani, O. Aktas, and D. Birant, "Balanced K-star: An explainable machine learning method for Internet-of-Things-enabled predictive maintenance in manufacturing," *Machines*, vol. 11, no. 3, p. 322, Feb. 2023.

[87] J. Grezmak, J. Zhang, P. Wang, K. A. Loparo, and R. X. Gao, "Interpretable convolutional neural network through layer-wise relevance propagation for machine fault diagnosis," *IEEE Sensors J.*, vol. 20, no. 6, pp. 3172–3181, Mar. 2020.

[88] J. Liu, S. Zheng, and C. Wang, "Causal graph attention network with disentangled representations for complex systems fault detection," *Rel. Eng. Syst. Saf.*, vol. 235, Jul. 2023, Art. no. 109232.

[89] A. Trilla, N. Mijatovic, and X. Vilasis-Cardona, "Unsupervised probabilistic anomaly detection over nominal subsystem events through a hierarchical variational autoencoder," *Int. J. Prognostics Health Manage.*, vol. 14, no. 1, pp. 1–14, May 2023.

[90] I. Errandonea, P. Ciáurriz, U. Alvarado, S. Beltrán, and S. Arrizabalaga, "Edge intelligence-based proposal for onboard catenary stagger amplitude diagnosis," *Comput. Ind.*, vol. 144, Jan. 2023, Art. no. 103781.

[91] B. Steenwinckel, D. De Paepe, S. Vanden Hautte, P. Heyvaert, M. Bentefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van Hoecke, and F. Ongenae, "FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning," *Future Gener. Comput. Syst.*, vol. 116, pp. 30–48, Mar. 2021.

[92] H. Li, D. Parikh, Q. He, B. Qian, Z. Li, D. Fang, and A. Hampapur, "Improving rail network velocity: A machine learning approach to predictive maintenance," *Transp. Res. Part C, Emerg. Technol.*, vol. 45, pp. 17–26, Aug. 2014.

[93] Z. A. Bukhsh, A. Saeed, I. Stipanovic, and A. G. Doree, "Predictive maintenance using tree-based classification techniques: A case of railway switches," *Transp. Res. Part C, Emerg. Technol.*, vol. 101, pp. 35–54, Apr. 2019.

[94] P. Cao, S. Zhang, and J. Tang, "Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning," *IEEE Access*, vol. 6, pp. 26241–26253, 2018.

[95] G. Hajgató, R. Wéber, B. Szilágyi, B. Tóthpál, B. Gyires-Tóth, and C. Hós, "PredMaX: Predictive maintenance with explainable deep convolutional autoencoders," *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101778.

[96] J. Jakubowski, P. Stanisz, S. Bobek, and G. J. Nalepa, "Explainable anomaly detection for hot-rolling industrial process," in *Proc. IEEE 8th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2021, pp. 1–10.

[97] N. Mylonas, I. Mollas, N. Bassiliades, and G. Tsoumakas, "Local multi-label explanations for random forest," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2022, pp. 369–384.

[98] J. Jakubowski, P. Stanisz, S. Bobek, and G. J. Nalepa, "Roll wear prediction in strip cold rolling with physics-informed autoencoder and counterfactual explanations," in *Proc. IEEE 9th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2022, pp. 1–10.

[99] D. Kim, G. Antariksa, M. P. Handayani, S. Lee, and J. Lee, "Explainable anomaly detection framework for maritime main engine sensor data," *Sensors*, vol. 21, no. 15, p. 5200, Jul. 2021.

[100] K. Michałowska, S. Riemer-Sørensen, C. Sterud, and O. M. Hjellset, "Anomaly detection with unknown anomalies: Application to maritime machinery," *IFAC-PapersOnLine*, vol. 54, no. 16, pp. 105–111, 2021.

[101] A. Bakdi, N. B. Kristensen, and M. Stakkeland, "Multiple instance learning with random forest for event logs analysis and predictive maintenance in ship electric propulsion system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7718–7728, Nov. 2022.

[102] W. Xu, Z. Zhou, T. Li, C. Sun, X. Chen, and R. Yan, "Physics-constraint variational neural network for wear state assessment of external gear pump," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 4, no. 1, pp. 1–11, May 2022.

[103] J. M. F. Salido and S. Murakami, "A comparison of two learning mechanisms for the automatic design of fuzzy diagnosis systems for rotating machinery," *Appl. Soft Comput.*, vol. 4, no. 4, pp. 413–422, Sep. 2004.

[104] R. Langone, A. Cuzzocrea, and N. Skantzos, "Interpretable anomaly prediction: Predicting anomalous behavior in Industry 4.0 settings via regularized logistic regression tools," *Data Knowl. Eng.*, vol. 130, Nov. 2020, Art. no. 101850.

[105] A. Klein. (Jul. 2021). *Hard Drive Failure Rates: A Look At Drive Reliability*. [Online]. Available: https://www.backblaze.com/blog/backblaze-hard-drive-stats-q1-2020/

[106] M. Amram, J. Dunn, J. J. Toledano, and Y. D. Zhuo, "Interpretable predictive maintenance for hard drives," *Mach. Learn. Appl.*, vol. 5, Sep. 2021, Art. no. 100042.

[107] P. M. Attia, A. Grover, N. Jin, K. A. Severson, T. M. Markov, Y.-H. Liao, M. H. Chen, B. Cheong, N. Perkins, Z. Yang, P. K. Herring, M. Aykol, S. J. Harris, R. D. Braatz, S. Ermon, and W. C. Chueh, "Closed-loop optimization of fast-charging protocols for batteries with machine learning," *Nature*, vol. 578, no. 7795, pp. 397–402, Feb. 2020.

[108] R. Csalódi, Z. Bagyura, and J. Abonyi, "Mixture of survival analysis models-cluster-weighted Weibull distributions," *IEEE Access*, vol. 9, pp. 152288–152299, 2021.

[109] F. Wang, Z. Zhao, Z. Zhai, Z. Shang, R. Yan, and X. Chen, "Explainability-driven model improvement for SOH estimation of lithium-ion battery," *Rel. Eng. Syst. Saf.*, vol. 232, Apr. 2023, Art. no. 109046.

[110] K. S. Hansen, N. Vasiljevic, and S. A. Sarensen. (May 2021). *Wind Farm Measurements*. [Online]. Available: https://data.dtu.dk/collections/Wind_Farm_measurements/5405418/3

[111] C. M. A. Roelofs, M.-A. Lutz, S. Faulstich, and S. Vogt, "Autoencoder-based anomaly root cause analysis for wind turbines," *Energy AI*, vol. 4, Jun. 2021, Art. no. 100065.

[112] M. Beretta, A. Julian, J. Sepulveda, J. Cusidó, and O. Porro, "An ensemble learning solution for predictive maintenance of wind turbines main bearing," *Sensors*, vol. 21, no. 4, p. 1512, Feb. 2021.

[113] M. Berno, M. Canil, N. Chiarello, L. Piazzon, F. Berti, F. Ferrari, A. Zaupa, N. Ferro, M. Rossi, and G. A. Susto, "A machine learning-based approach for advanced monitoring of automated equipment for the entertainment industry," in *Proc. IEEE Int. Workshop Metrol. Ind. 4.0 IoT*, Jun. 2021, pp. 386–391.

[114] E. Anello, C. Masiero, F. Ferro, F. Ferrari, B. Mukaj, A. Beghi, and G. A. Susto, "Anomaly detection for the industrial Internet of Things: An unsupervised approach for fast root cause analysis," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2022, pp. 1366–1371.

[115] D. Marcato, G. Arena, D. Bortolato, F. Gelain, V. Martinelli, E. Munaron, M. Roetta, G. Savarese, and G. A. Susto, "Machine learning-based anomaly detection for particle accelerators," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2021, pp. 240–246.

[116] L. Felsberger, A. Apollonio, T. Cartier-Michaud, A. Müller, B. Todd, and D. Kranzlmüller, "Explainable deep learning for fault prognostics in complex systems: A particle accelerator use-case," in *Lecture Notes in Computer Science*. Cham, Switzerland: Springer, 2020, pp. 139–158.

[117] P. Bellini, D. Cenni, L. A. I. Palesi, P. Nesi, and G. Pantaleo, "A deep learning approach for short term prediction of industrial plant working status," in *Proc. IEEE 7th Int. Conf. Big Data Comput. Service Appl.*, Aug. 2021, pp. 9–16.

[118] H. Choi, D. Kim, J. Kim, J. Kim, and P. Kang, "Explainable anomaly detection framework for predictive maintenance in manufacturing systems," *Appl. Soft Comput.*, vol. 125, Aug. 2022, Art. no. 109147.

[119] M. McCann and A. Johnston. (2008). *SECOM*. [Online]. Available: https://archive.ics.uci.edu/dataset/179/secom

[120] M. Gashi, B. Mutlu, and S. Thalmann, "Impact of interdependencies: Multi-component system perspective toward predictive maintenance based on machine learning and XAI," *Appl. Sci.*, vol. 13, no. 5, p. 3088, Feb. 2023.

[121] Q. Cao, C. Zanni-Merk, A. Samet, F. D. B. de Beuvron, and C. Reich, "Using rule quality measures for rule base refinement in knowledge-based predictive maintenance systems," *Cybern. Syst.*, vol. 51, no. 2, pp. 161–176, Feb. 2020.

[122] V. M. Janakiraman, "Explaining aviation safety incidents using deep temporal multiple instance learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 406–415.

[123] I. Katser, V. Kozitsin, V. Lobachev, and I. Maksimov, "Unsupervised offline changepoint detection ensembles," *Appl. Sci.*, vol. 11, no. 9, p. 4280, May 2021.

[124] D. Dua and C. Graff. (2019). *Uci Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[125] K. Scott, D. Kakde, S. Peredriy, and A. Chaudhuri, "Computational enhancements to the mahalanobis-taguchi system to improve fault detection and diagnostics," in *Proc. Annu. Rel. Maintainability Symp. (RAMS)*, Jan. 2023, pp. 1–7.

[126] S. J. Upasane, H. Hagras, M. H. Anisi, S. Savill, I. Taylor, and K. Manousakis, "A big bang-big crunch type-2 fuzzy logic system for explainable predictive maintenance," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2021, pp. 1–8.

[127] B. Verkuil, C. E. Budde, and D. Bucur, "Automated fault tree learning from continuous-valued sensor data: A case study on domestic heaters," 2022, *arXiv:2203.07374*.

[128] L. Lorenti, G. De Rossi, A. Annoni, S. Rigutto, and G. A. Susto, "CUAD-Mo: Continuos unsupervised anomaly detection on machining operations," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2022, pp. 881–886.

[129] B. A. U. Olimov, K. C. Veluvolu, A. Paul, and J. Kim, "UzADL: Anomaly detection and localization using graph Laplacian matrix-based unsupervised learning method," *Comput. Ind. Eng.*, vol. 171, Sep. 2022, Art. no. 108313.

[130] A. Lourenço, M. Fernandes, A. Canito, A. Almeida, and G. Marreiros, "Using an explainable machine learning approach to minimize opportunistic maintenance interventions," in *Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst.*, 2022, pp. 41–54.

[131] O. Serradilla, E. Zugasti, J. Ramirez de Okariz, J. Rodriguez, and U. Zurutuza, "Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data," *Appl. Sci.*, vol. 11, no. 16, p. 7376, Aug. 2021.

[132] M. Hermansa, M. Kozielski, M. Michalak, K. Szczyrba, Ł. Wróbel, and M. Sikora, "Sensor-based predictive maintenance with reduction of false alarms—A case study in heavy industry," *Sensors*, vol. 22, no. 1, p. 226, Dec. 2021.

[133] D. Xu, W. Cheng, J. Ni, D. Luo, M. Natsumeda, D. Song, B. Zong, H. Chen, and X. Zhang, "Deep multi-instance contrastive learning with dual attention for anomaly precursor detection," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2021, pp. 91–99.

[134] B. Steurtewagen and D. Van den Poel, "Adding interpretability to predictive maintenance by machine learning on sensor data," *Comput. Chem. Eng.*, vol. 152, Sep. 2021, Art. no. 107381.

[135] A. T. Keleko, B. Kamsu-Foguem, R. H. Ngouna, and A. Tongne, "Health condition monitoring of a complex hydraulic system using deep neural network and DeepSHAP explainable XAI," *Adv. Eng. Softw.*, vol. 175, Jan. 2023, Art. no. 103339.

[136] G. Chen, M. Liu, and Z. Kong, "Temporal-logic-based semantic fault diagnosis with time-series data from industrial Internet of Things," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4393–4403, May 2021.

[137] A. Schmetz, C. Vahl, Z. Zhen, D. Reibert, S. Mayer, D. Zontar, J. Garcke, and C. Brecher, "Decision support by interpretable machine learning in acoustic emission based cutting tool wear prediction," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2021, pp. 629–633.

[138] T. V. A. Howard, S. Dane. (2018). *VSB Power Line Fault Detection*. [Online]. Available: https://kaggle.com/competitions/vsb-power-line-fault-detection

[139] S. Simmons, L. Jarvis, D. Dempsey, and A. W. Kempa-Liehr, "Data mining on extremely long time-series," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2021, pp. 1057–1066.

[140] Y. Zhang, P. Wang, K. Liang, Y. He, and S. Ma, "An alarm and fault association rule extraction method for power equipment based on explainable decision tree," in *Proc. 11th Int. Conf. Power Energy Syst. (ICPES)*, Dec. 2021, pp. 442–446.

[141] 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.*, vol. 5, no. 2, pp. 1–14, Aug. 2023.

[142] L. Xia, Y. Liang, J. Leng, and P. Zheng, "Maintenance planning recommendation of complex industrial equipment based on knowledge graph and graph neural network," *Rel. Eng. Syst. Saf.*, vol. 232, Apr. 2023, Art. no. 109068.

[143] G. Tod, A. P. Ompusunggu, and E. Hostens, "An improved first-principle model of AC powered solenoid operated valves for maintenance applications," *ISA Trans.*, vol. 135, pp. 551–566, Apr. 2023.

[144] M. Mahmoodian, F. Shahrivar, S. Setunge, and S. Mazaheri, "Development of digital twin for intelligent maintenance of civil infrastructure," *Sustainability*, vol. 14, no. 14, p. 8664, Jul. 2022.

[145] I. Konovalenko and A. Ludwig, "Generating decision support for alarm processing in cold supply chains using a hybrid k-NN algorithm," *Expert Syst. Appl.*, vol. 190, Mar. 2022, Art. no. 116208.

[146] A. Dhaou, A. Bertoncello, S. Gourvénec, J. Garnier, and E. Le Pennec, "Causal and interpretable rules for time series analysis," *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, pp. 2764–2772, Aug. 2021.

[147] *Water Pipe (WCORP-002)*. Accessed: Nov. 2023. [Online]. Available: https://catalogue.data.wa.gov.au/dataset/water-pipe-wcorp-002

[148] P. Castle, J. Ham, M. Hodkiewicz, and A. Polpo, "Interpretable survival models for predictive maintenance," in *Proc. 30th Eur. Saf. Rel. Conf. 15th Probabilistic Saf. Assessment Manage. Conf.*, 2020, pp. 3392–3399.

[149] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers Big Data*, vol. 4, Jul. 2021, Art. no. 688969.

[150] A. Saabas. (2014). *Interpreting Random Forests*. [Online]. Available: http://blog.datadive.net/interpreting-random-forests/

[151] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[152] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[153] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3319–3328.

[154] D. Janzing, D. Balduzzi, M. Grosse-Wentrup, and B. Schölkopf, "Quantifying causal influences," *Ann. Statist.*, vol. 41, no. 5, pp. 2324–2358, Oct. 2013.

[155] D. Dandolo, C. Masiero, M. Carletti, D. Dalle Pezze, and G. A. Susto, "AcME—Accelerated model-agnostic explanations: Fast whitening of the machine-learning black box," *Expert Syst. Appl.*, vol. 214, Mar. 2023, Art. no. 119115.

[156] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," 2017, *arXiv:1706.03825*.

[157] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *SSRN Electron. J.*, vol. 1, pp. 1–11, Nov. 2017.

[158] TeamHG-Memex. *Teamhg-memex/eli5: A Library for Debugging/inspecting Machine Learning Classifiers and Explaining Their Predictions*. Accessed: Nov. 2023. [Online]. Available: https://github.com/TeamHG-Memex/eli5

[159] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020.

[160] M. Carletti, M. Terzi, and G. A. Susto, ‘‘Interpretable anomaly detection with DIFFI: Depth-based feature importance of isolation forest,’’ *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105730.

[161] I. Mollas, N. Bassiliades, and G. Tsoumakas, ‘‘Conclusive local interpretation rules for random forests,’’ *Data Mining Knowl. Discovery*, vol. 36, no. 4, pp. 1521–1574, Jul. 2022.

[162] (2020). *AI4I 2020 Predictive Maintenance Dataset*. [Online]. Available: https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset

[163] K. Simonyan, A. Vedaldi, and A. Zisserman, ‘‘Deep inside convolutional networks: Visualising image classification models and saliency maps,’’ 2013, *arXiv:1312.6034*.

[164] J. N. Morgan and J. A. Sonquist, ‘‘Problems in the analysis of survey data, and a proposal,’’ *J. Amer. Stat. Assoc.*, vol. 58, no. 302, p. 415, Jun. 1963.

[165] D. Bertsimas and J. Dunn, ‘‘Optimal classification trees,’’ *Mach. Learn.*, vol. 106, no. 7, pp. 1039–1082, Jul. 2017.

[166] D. Bertsimas, J. Dunn, E. Gibson, and A. Orfanoudaki, ‘‘Optimal survival trees,’’ *Mach. Learn.*, vol. 111, no. 8, pp. 2951–3023, Aug. 2022.

[167] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, ‘‘Lightgbm: A highly efficient gradient boosting decision tree,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–18.

[168] H. A. Watson et al., *Launch Control Safety Study*. Murray Hill, NY, USA: Bell labs, 1961.

[169] D. A. Augusto and H. J. C. Barbosa, ‘‘Symbolic regression via genetic programming,’’ in *Proc. 6th Brazilian Symp. Neural Netw.*, vol, vol. 1, 2000, pp. 173–178.

[170] L. A. Zadeh, ‘‘Fuzzy logic,’’ *Computer*, vol. 21, no. 4, pp. 83–93, Apr. 1988.

[171] M. Ravanelli and Y. Bengio, ‘‘Speaker recognition from raw waveform with SincNet,’’ in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 1021–1028.

[172] P. Velič ković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, ‘‘Graph attention networks,’’ 2017, *arXiv:1710.10903*.

[173] O. Maler and D. Nickovic, ‘‘Monitoring temporal properties of continuous signals,’’ in *Proc. Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Cham, Switzerland: Springer, 2004, pp. 152–166.

[174] P. S. Bandyopadhyay and M. R. Forster, *Philosophy of Statistics*. Stanford, CA, USA: Stanford Univ., 2011, pp. 1–50.

[175] K. Pearson, ‘‘X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,’’ *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 50, no. 302, pp. 157–175, Jul. 1900.

[176] Student, ‘‘The probable error of a mean,’’ *Biometrika*, vol. 6, no. 1, p. 1–25, Mar. 1908.

[177] D. R. Cox, ‘‘Regression models and life-tables,’’ *J. R. Stat. Soc., B (Methodol.)*, vol. 34, no. 2, pp. 187–202, Jan. 1972.

[178] L. E. Baum and T. Petrie, ‘‘Statistical inference for probabilistic functions of finite state Markov chains,’’ *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[179] D. A. Reynolds, ‘‘Gaussian mixture models,’’ *Encyclopedia Biometrics*, vol. 741, nos. 659–663, pp. 1–14, Jun. 2009.

[180] T. Hastie and R. Tibshirani, ‘‘Generalized additive models,’’ *Stat. Sci.*, vol. 1, no. 3, pp. 297–310, Aug. 1986, doi: 10.1214/ss/1177013604.

[181] G. Taguchi, G. Taguchi, and R. Jugulum, *The Mahalanobis-Taguchi Strategy: A Pattern Technology System*. Hoboken, NJ, USA: Wiley, 2002.

[182] H. Nori, S. Jenkins, P. Koch, and R. Caruana, ‘‘InterpretML: A unified framework for machine learning interpretability,’’ 2019, *arXiv:1909.09223*.

[183] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ‘‘Attention is all you need,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–17.

[184] M. Grieves and J. Vickers, ‘‘Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,’’ in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Springer, 2017, pp. 85–113.

[185] E. Fix and J. L. Hodges, ‘‘Discriminatory analysis. Nonparametric discrimination: Consistency properties,’’ *Int. Stat. Rev. Revue Internationale De Statistique*, vol. 57, no. 3, p. 238, Dec. 1989.

[186] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, ‘‘Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,’’ *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

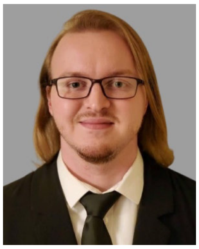
[187] T. Miller, ‘‘Explanation in artificial intelligence: Insights from the social sciences,’’ *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.

[188] L. Coroama and A. Groza, ‘‘Evaluation metrics in explainable artificial intelligence (XAI),’’ in *Proc. 2nd Int. Conf., Adv. Res. Technol., Inf., Innov. Sustain. Revis. Select. Papers, Part I*, 2022, pp. 401–413.

[189] M. Sisk, M. Majlis, C. Page, and A. Yazdinejad, ‘‘Analyzing XAI metrics: Summary of the literature review,’’ 2022, *techrxiv.21262041.v1*.

[190] M. A. Kadir, A. Mosavi, and D. Sonntag, ‘‘Assessing XAI: Unveiling evaluation metrics for local explanation, taxonomies, key concepts, and practical applications,’’ *EngrXiv preprint*, 2023.

[191] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, ‘‘Metrics for explainable AI: Challenges and prospects,’’ 2018, *arXiv:1812.04608*.



LOGAN CUMMINS (Member, IEEE) received the B.S. degree in computer science and engineering from Mississippi State University, where they are currently pursuing the Ph.D. degree in computer science (minor in cognitive science).

They are a Graduate Research Assistant with the Predictive Analytics and Technology Integration (PATENT) Laboratory in collaboration with the Institute for Systems Engineering Research. Additionally, they perform research with the Social Therapeutic and Robotic Systems (STaRS) Research Laboratory. Their research interests include explainable artificial intelligence and its applications, cognitive science, and human–computer interactions as applied to human-agent teamming. They are a member of ACM at Mississippi State University.

ALEXANDER SOMMERS (Member, IEEE) received the B.S. degree in computer science from the Saint Vincent Collage and the M.S. degree in computer science from Southern Illinois University. He is currently pursuing the Ph.D. degree in computer science with Mississippi State University with a concentration in machine learning.

He is a Graduate Research Assistant with the Predictive Analytics and Technology Integration Laboratory (PATENT Lab) in collaboration with the Institute for Systems Engineering Research. His work concerns synthetic time-series generation and remaining-useful-life prediction. His research interests include the application of machine learning to reliability engineering and lacuna discovery. He is a member of ACM.

SOMAYEH BAKHTIARI RAMEZANI (Member, IEEE) received the B.S. degree in computer engineering and the M.S. degree in information technology engineering from Iran University of Science and Technology, in 2004 and 2008, respectively. She is currently pursuing the Ph.D. degree in computer science with Mississippi State University.

She is a Graduate Research Assistant with the Predictive Analytics and Technology Integration (PATENT) Laboratory in collaboration with the Institute for Systems Engineering Research. Prior to joining Mississippi State University, in 2019, she was with several companies in the energy and healthcare sectors as an HPC Programmer and a Data Scientist. Her research interests include probabilistic modeling and optimization of dynamic systems, the application of ML, quantum computation, and time-series segmentation in the healthcare sector. She is a member of ACM, the President of the ACM-W Student Chapter at Mississippi State University, and the Chair of the IEEE-WIE AG Mississippi Section. She is a 2021 SIGHPC Computational and Data Science Fellow.

SUDIP MITTAL (Member, IEEE) received the Ph.D. degree in computer science from the University of Maryland, Baltimore County, in 2019. He is an Assistant Professor with the Department of Computer Science and Engineering, Mississippi State University. His goal is to develop the next generation of cyber defense systems that help protect various organizations and people. At Mississippi State, he leads the Secure and Trustworthy Cyberspace (SECRETS) Laboratory. He has published over 80 journals and conference papers in leading cybersecurity and AI venues. He has received funding from NSF, USAF, USACE, and various other department of defense programs. His primary research interests include cybersecurity and artificial intelligence. He is a member of ACM. He serves as a program committee member and the program chair for leading AI and cybersecurity conferences and workshops. His work has been cited in the LA times, Business Insider, WIRED, the Cyberwire, and other venues.

JOSEPH JABOUR received the B.S. degree in computer science from the University of Mississippi in 2019. He is currently pursuing the M.S. degree in computer science with Mississippi State University. He is a Computer Scientist with the Information Technology Laboratory (ITL), Engineering Research and Development Center (ERDC), Vicksburg, MS, USA, where he joined in 2019. He has pursued research in the field of artificial intelligence and machine learning. Additionally, he has performed a significant amount of work in the fields of data visualization, digital twins, and many other forms of research. He has presented at several nationally recognized conferences. He was the Vice Chair of the ERDC Association of Computing Machinery, and is currently a Facilitator of the ERDC ITL Field Training Exercise based off of leadership principles from the Echelon Front. He has received awards for his research and development, including the ERDC Award for Outstanding Innovation in Research and Development. He seeks to push past the forefront of technological development and innovation and endeavors to identify and implement solutions to our nation's leading causes of concern.

MARIA SEALE received the B.S. degree in computer science from the University of Southern Mississippi, in 1987, and the M.S. and Ph.D. degrees in computer science from Tulane University, in 1992 and 1995, respectively.

Prior to joining the Information Technology Laboratory, U.S. Army Engineer Research and Development Center (ERDC), in 2016, she held positions with the Institute for Naval Oceanography, the U.S. Naval Research Laboratory, and various private companies; and a tenured associate professorship with the University of Southern Mississippi. At ERDC, she has been involved with research in making scalable machine learning algorithms available on high-performance computing platforms and expanding the center's capabilities to manage and analyze very large data sets. Her research interests include natural language processing, machine learning, natural computing, high-performance data analytics, and prognostics and health management for engineered systems. She is a member of the Prognostics and Health Management Society, the American Society of Mechanical Engineers, and the Association of Computing Machinery.

SHAHRAM RAHIMI (Member, IEEE) is currently a Professor and the Head of the Department of Computer Science and Engineering, Mississippi State University. Prior to that, he led the Department of Computer Science, Southern Illinois University, for five years. He is also a recognized leader in the area of artificial and computational intelligence, with over 220 peer-reviewed publications and a few patents or pending patents in this area. His intelligent algorithm for patient flow optimization and hospital staffing is currently used in over 1000 emergency departments across the nation. He was named one of the top ten AI technology for healthcare, in 2018, by *HealthTech Magazine*. He has secured over $20M of federal and industry funding as a PI or a co-PI in the last 20 years. He has also organized 15 conferences and workshops in the areas of computational intelligence and multi-agent systems over the past two decades.

He is a member of the IEEE New Standards Committee in Computational Intelligence. He provides advice to staff and administration at the Federal Government on Predictive Analytics for Foreign Policy. He was a recipient of the 2016 Illinois Rising Star Award from ISBA and selected among 100s of highly qualified candidates. He has served as the Editor-in-Chief for two leading computational intelligence journals and is on the editorial board of several other journals.