A Type-2 Fuzzy Based Explainable AI System for Predictive Maintenance within the Water Pumping Industry

Shreyas J Upasane, *Student Member, IEEE,* Hani Hagras, *Fellow, IEEE,* Mohammad Hossein Anisi, *Senior Member, IEEE,* Stuart Savill, Ian Taylor and Kostas Manousakis

***Abstract* — Industrial maintenance has undergone a paradigm shift due to the emergence of artificial intelligence (AI), the Internet of Things (IoT), and cloud computing. Rather than accepting the drawbacks of reactive maintenance, leading firms worldwide are embracing "predict-and-prevent" maintenance. However, opaque box AI models are sophisticated and complex for the average user to comprehend and explain. This limits the AI employment in predictive maintenance, where it is vital to understand and evaluate the model before deployment. In addition, it's also important to comprehend the maintenance system's decisions. This paper presents a type-2 fuzzy-based Explainable AI (XAI) system for predictive maintenance within the water pumping industry. The proposed system is optimised via Big-Bang Big-Crunch (BB-BC), which maximises the model accuracy for predicting faults while maximising model interpretability. We evaluated the proposed system on water pumps using real-time data obtained by our hardware placed at real-world locations around the United Kingdom and compared our model with Type-1 Fuzzy Logic System (T1FLS), a Multi-Layer Perceptron (MLP) Neural Network, an effective deep learning method known as stacked autoencoders (SAEs) and an interpretable model like decision trees (DT). The proposed system predicted water pumping equipment failures with good accuracy (outperforming the T1FLS accuracy by 8.9% and DT by 529.2% while providing comparable results to SAEs and MLPs) and interpretability. The system predictions comprehend why a specific problem may occur, which leads to better and more informed customer visits to reduce equipment failure disturbances. It will be shown that 80.3% of water industry specialists strongly agree with the model's explanation, determining its acceptance.**

***Impact Statement* — Advances in artificial intelligence for maintenance analytics are frequently linked to advanced statistical methods that are highly sophisticated, relying on a massive amount of training data and complex models to find patterns and make predictions. This intricacy and the statistical nature of correlations between input data make them difficult to understand even for skilled users. Advanced analytics to predict failures can enhance equipment uptime by up to 20% [36]. Increased trust in the predictive maintenance system is one of the stated benefits of improving the explainability of AI systems. Maintenance engineers will have higher faith in AI outputs if they understand what led to a judgement or suggestion made by AI. Furthermore, an explainable AI system will allow the service engineer to understand what is causing a given fault, allowing ordering the correct spare part to minimise any possible stoppage of the assigned equipment. In addition, this will reduce the unneeded service trips and the associated CO_2 emissions. With increased trust in the system, extracting business insights and long-term consequences from the model enhances traditional maintenance services and converts them into innovative, effective, and efficient maintenance services. This work is one of the first real-world XAI-based systems for predictive maintenance in the water pumping industry. Although the system presented in this paper is applied to the water pumping industry, it also scales up for other similar utility-based industries, which can profoundly impact millions of humans worldwide.**

***Index Terms*— Interval type-2 fuzzy logic, IoT, predictive maintenance (PdM), Explainable AI (XAI)**

I. Introduction

Industrial systems require regular maintenance to sustain high levels of productivity. However, conventional maintenance methods, such as manual inspection by service engineers and run-to-failure maintenance, are labour intensive and inefficient when meeting service-based industries' current and future problems. The building services sector is an example of a service-based industry in which maintenance is critical. Companies in this industry consider providing innovative solutions to minimise any inconveniences their clients might experience because of equipment failures.

The adoption of proper maintenance strategies in the building services industry ensures that the equipment is operating correctly, resulting in better and more informed service visits, which in turn results in lower equipment failure costs and interruptions. This has resulted in a significant evolution of maintenance over time, owing to substantial contributions made in scientific and industrial environments. As a result, several maintenance policies have emerged. For instance, *Reactive or run-to-failure maintenance* primarily focuses on fault repairs and helps in the restoration of non-operational assets to their desired functioning state [1],[2],[3]. It is the most common type of maintenance, and it is frequently used when the equipment has a low value, is easy to replace, and has no significant impact on the business process. According to studies, reactive maintenance accounts for more than 55% of device maintenance [37]. While this is appropriate in some circumstances, it is highly costly to

S. J. Upasane, H. Hagras, M. Anisi are with the Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, U.K. (e-mail: sjupas@essex.ac.uk).
S. Savill, I. Taylor and K. Manousakis are with Aquatronic Group Management Plc, Copford, UK

operate in this manner because breakdowns occur randomly and are thus unpredictable. *Preventive maintenance* is a type of maintenance that is planned and performed regularly to ensure the continued operation of the assets. As a result, prospective time-based expected failures are avoided, and unexpected failures are reduced, hence extending the asset's life [2]. This strategy is typically used when the maintenance cost is moderate and when it is possible to complete the work outside of regular production hours. *Condition Based Maintenance* is akin to preventative maintenance, except instead of relying on pre-scheduled periods, it evaluates the actual use of the assets. This type of maintenance is typically used when maintenance costs are not too expensive, and maintenance may be completed at convenient times. These approaches, however, are not always sufficient. *Predictive maintenance (PdM)* is a way to make sure that system maintenance and operations are as efficient as possible by taking into account the current operating status of equipment and systems [4]. It is the fourth and most advanced type of maintenance using a variety of process data and robust analytical methodologies. *Predictive maintenance (PdM)* can detect faults well before emergency intervention is required [2],[3].

In most cases, this maintenance strategy is used when downtimes or maintenance costs are substantial. When maintenance chores are complex, it might also make scheduling a little easier. According to [5], predictive maintenance is a crucial component of Industry 4.0. Predictive maintenance (*PdM)*, on average, increases productivity by 25%, reduces breakdowns by 70% and saves maintenance expenses by 25% [6]. *PdM* is based on the premise that unnecessary visits can be minimized when the total failure of equipment can be forecasted, resulting in the end-user never being without a critical service.

The potential failure (P) and the functional failure (F) P–F curve, as shown in Fig. 1, clearly demonstrate the predictive maintenance capabilities. As the name suggests, the P–F curve is used to illustrate the behaviour of an asset before it fails. The curve demonstrates that asset performance or condition deteriorates with time, resulting in functional failure. When the equipment is found to be failing, it is called a potential failure. The point at which the equipment fails functionally is called functional failure. PdM helps to prevent the equipment from progressing further in a dangerous state.

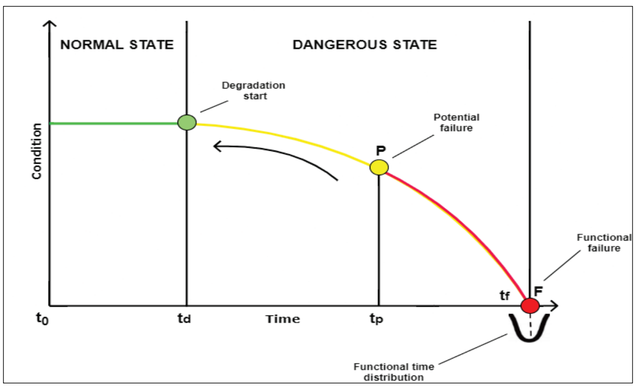


Fig. 1. Predictive maintenance in the scope of the P-F curve [7]

In recent years, deep learning algorithms have been used in predictive maintenance where [8] and [9] used Long Short-Term Memory (LSTMs) and Convolution Neural Networks (CNN) architectures to anticipate the remaining useful life of wind turbines. Research also demonstrates how to assess the degeneration of rolling bearings using LSTMs [10]. However, it compares it to other deep learning models and does not consider simpler machine learning algorithms. Senseye [11] is a system that collects data from various sources, analyses it, and notifies a designated person when an anomaly is detected, or a breakdown is imminent. The Real-time Remote Machinery Prognostics and Health Management (R2MPHM) platform [12] is another platform that collects data, processes it, and extracts features that allow identifying potential faults and determining the remaining usable life of the equipment. It uses a variety of methodologies and algorithms, including Neural Networks and Fuzzy Logic Prediction. None of these informs the reasons or causes of the abnormality or failure. In [38], the author used bagged decision trees to apply explainability for predictive maintenance dataset and concludes that the explanations provided by the decision trees tends to be instable in a considerable number of cases. This is true as any small change in the data can lead to major changes in the structure of the tree. [39] developed the QARMA (Quantitative Association Rule Mining) algorithmic family for industrial IoT predictive maintenance applications. This algorithm family focuses on the mining of quantitative rules. In terms of explainability, QARMA outperforms deep learning and ML methods; however, they generate an excessive number of rules, most of which are complex. Using LSTM, [42] developed a PdM framework. They collected data for three pumps using BAS and IoT devices as part of this framework but did not specify the associated operational parameters or detected faults. In [43], the authors evaluated the interpretability of current data-driven fault detection and diagnosis (FDD) techniques. They reviewed the data-driven FDD methodologies for constructing mechanical & electrical engineering (M&E) service failures using a three-layer Bayesian Belief Network (BBN), adapting the maintainability rules as an additional layer for FDD. However, they did not explain the actual implementation of the maintainability rules integrated FDD framework. [44], re-examined the standard procedure for fault detection and diagnosis (FDD) in electric drives by employing ML algorithms and reviewed the existing fault detection approaches. They concluded that the fundamental flaw of the data driven FDD method is the inability to interpret the results or explain the events observed in any given electric drive.

The water pumping industry is risk-averse, and one of the most significant barriers to AI adoption is the black-box nature of many deep learning techniques. Because of this, these models cannot be used in predictive maintenance, where it is necessary to comprehend and evaluate the model before it can be used. In addition, it is critical to understand the reasoning behind any given choice provided by the maintenance system [14]. Service engineers may lose faith in black-box predictive AI models and disable such solutions if they lose confidence in them. Without explainability in the model, the possibilities for introducing expert users' knowledge to supplement the model are limited, given that the end-user can't interact with the model [33]. These challenges are critical in the water pumping industry as it impedes from capturing information that service engineers have built up over time as expertise.

Most water pumping companies have standardised regular equipment checks to make sure everything is working correctly. The challenge is that most visits result in no work being accomplished, wasting engineer time and resources, and having a negative impact on the environment. However, suppose an issue with the water pumps arises. In that case, the end-users may be deprived of water or heated air (depending on the type of unit), or their manufacturing process may be put on hold till the arrival of an engineer. If the engineer isn't equipped with tools, a second visit is frequently required resulting in enormous losses for the premise associated with unneeded service trips and the coupled CO_2 emissions. Explainable AI (XAI) with predictive maintenance solves this issue by making AI models more transparent and explainable [15]. XAI should also assist the end-user in comprehending the model output and drawing conclusions from the prediction, enhancing the confidence and credibility needed in real-world operations. By integrating the XAI system, the service engineer will be able to determine what is causing a certain fault and order the appropriate spare part, minimising any potential equipment downtime.

Our previous paper [22] presented an XAI system with optimised IT2FLS that uses the BB-BC algorithm to predict one pump fault. We also discussed how the system's predictions and the rules could be used to determine why that particular fault might happen, leading to better and more informed service visits. In this paper, we extended our study within the water pumping industry to include most of the faults which can occur in pumps with real-world experiments achieved from 20 real-world sites all over the UK. In addition, we presented a novel explainable framework which introduces a novel rules post-processing engine and a unique traffic light system which is supported by decision module to generate automatic maintenance tasks for the service engineers. This was not present in the [22] where the service engineers found it difficult to comprehend the outputs from [22] when an alert was raised. We evaluated the proposed system on various pump-based systems using real-world data obtained by our hardware placed at real-world locations around the United Kingdom. The proposed system provided a highly interpretable and relatively accurate model for forecasting equipment failures in the water pumping industries. The system predictions comprehend why a specific problem may occur, which leads to better and more informed visits to customers so that disturbances caused by equipment failures are reduced.

Our paper's scientific contribution is developing and evaluating the first explainable AI-based predictive maintenance system based on Type-2 fuzzy logic within the water pumping industry. The research proposes a novel way to enhance the accuracy and explainability of the system's predictions, which is especially pertinent to the water pumping business. In predicting the likelihood of failure, our IT2FLS-based predictive system surpasses the T1FLS-based equivalent by obtaining 8.99% greater combined average accuracy over four fault test data. It also outperforms the Decision Trees in terms of accuracy, with a 529.21% improvement. Furthermore, the system produced clear and interpretable explanations for its forecasts, which can be used to improve maintenance decision-making. The system also demonstrated its practical utility by lowering downtime and maintenance costs in the water pumping industry. This is a substantial cost saving for the industry and highlights the suggested system's ability to increase the efficiency and effectiveness of maintenance operations. The study's findings show that the system is highly accurate and explainable, improving maintenance decision-making and reducing industry costs.

The rest of this paper is organised as follows; in section II, we will provide a brief overview of type-2 Fuzzy Logic Systems (FLSs). Section III will present an overview of water pumping systems and their common faults. Section IV will give an overview of the proposed predictive maintenance system. The proposed optimised type-2 fuzzy-based system for explainable predictive maintenance is presented in Section V. Section VI will present the experiments and results, while the conclusions and future work are presented in Section VII.

II. A Brief Overview on Type-2 Fuzzy Systems

Fuzzy Logic Systems (FLSs) seek to emulate human thinking rather than represent the brain's architecture, as would be the case with a neural network. Approximate rather than precise reasoning is the objective when it comes to human thinking. A collection of linguistic If-Then rules is used to represent a given behaviour in an understandable and human-readable manner [13].

The Interval Type-2 FLS (IT2FLS) is depicted in Fig. 2 contains five components: fuzzifier, rule base, inference engine, type-Reducer and a defuzzifier. One difference between type-1 FLS and the type-2 FLS illustrated in Fig. 2 is that the type-1 FLS does not have a type-Reducer and employs type-1 fuzzy sets for input and output [16][22]. The type-1 fuzzy sets have precise Membership Functions (MFs), which cannot directly handle the high levels of uncertainties encountered in real-world environments.

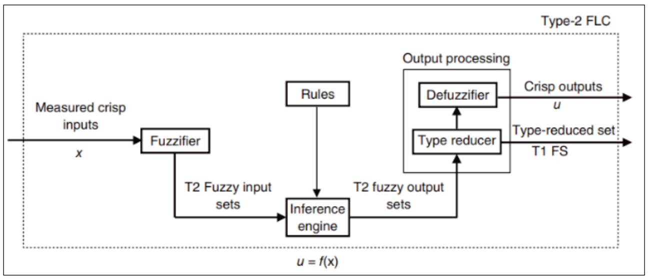


Fig. 2. Type-2 FLS

The IT2FLS works as follows: fuzzification of the crisp inputs into an input type-2 fuzzy set is performed initially. A type-2 fuzzy set has a membership function, but unlike a type-1 MF, it is three-dimensional and has a Footprint of Uncertainty (FOU). This FOU provides more design degrees of freedom to MFs, implying that type-2 fuzzy sets have a more substantial capacity for modelling and handling encountered uncertainty [17].

We employ interval type-2 fuzzy sets [23], shown in Fig. 3, to represent the inputs and outputs of the IT2FLS in this research because they are easier to compute than general type-2 fuzzy sets. As seen in Fig. 3a, the 25°C membership is a function in the primary membership domain that takes values from 0.6 to 0.8 with associated third dimension interval distribution (Fig. 3b), called a secondary MF.

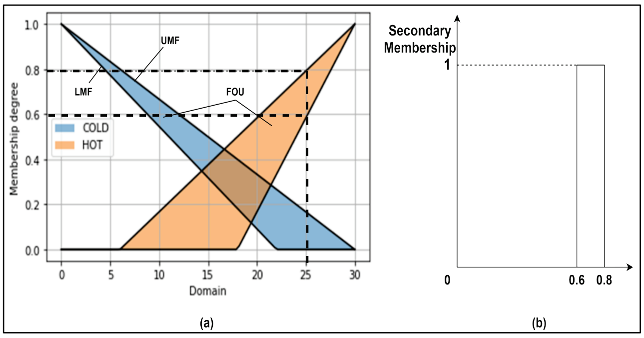


Fig. 3. Interval Type-2 Fuzzy Sets showing lower membership function (LMF) & upper membership function (UMF)

The inference engine uses the input type-2 fuzzy sets to activate the rule base and generates the output type-2 fuzzy sets once the inputs have been fuzzified. In terms of the rule base, there is no difference between a type-1 FLS and a type-2 FLS other than the fuzzy sets being interval type-2 fuzzy sets rather than type-1 fuzzy sets. Finally, the output type-2 sets generated in the preceding phases are transformed into a single number. [29],[30],[33]. It is possible to produce crisp outputs from IT2FLSs using two ways, the first of which is a traditional two-step procedure in which the output type-2 sets are transformed into type-reduced interval type-1 sets and then defuzzified. The second way is direct defuzzification, which was devised in response to the previous method's computing difficulty.

III. An Overview of the Water Pumping Systems and Their Common Faults

Water pump systems are used worldwide in various applications, from freshwater supply, grey/black water removal, and heating & cooling applications. Their exact usage may vary, but they are all required to move the pumped fluid at the required rate as efficiently as possible. All pump systems consist of a motive power source, which could be provided by either electricity or an internal combustion engine. This rotates an impellor assembly that could be of axial flow, centrifugal design, or a metering system, a positive displacement design. There are sensors and controls to establish when and how the pump should run. All these need to be operating correctly for the system to be efficient. Some applications rely on pumps running for short durations but with exact timing, whilst in others, the pump(s) may run for extended periods or even constantly but may vary their speed to match system requirements.

An inefficient or failing pump system can display problems in many forms. The outlet pressure may be incorrect, be it too high or too low. The flow rate could be false again, too high or too low. The pump could be meeting its pressure and flow requirements but be leaking, making noise, or consuming too much energy for the work being produced. Mechanically, the pump can become inefficient when cavitation (voids or bubbles form in the fluid surrounding the impellor as the localised fluid stresses become too high) occurs. Wear within the pump could cause the fluid to bypass internally, massively reducing efficiency and inducing heating.

Lots of secondary systems also rely on pump systems to operate, and a pump system operating inefficiently has direct consequences on the operation of these other systems. Excessive pressures can cause structural damage to the connected plant. Insufficient pressure or incorrect flow rate causes the connected plant to become inefficient or may even cause the pumped fluid to change state (flashing to steam), thus damaging other parts of the system.

According to [31],[32], energy and maintenance costs account for more than 50–95% of pump ownership costs. Wasted energy often results in high maintenance of the system.

IV. An Overview of the Proposed Predictive Maintenance System

The Internet of Things (IoT) enables predictive maintenance models to be developed using sensor data. A sensor translates physical quantities into electrical values [18]. Predictive maintenance algorithms take advantage of the IoT to boost the system's efficiency by capturing and monitoring this sensor data. Using the Internet of Things-enabled predictive maintenance, organisations can use real-time data, combined with service and repair history, to intelligently predict when a piece of machinery or equipment component might break down or not work as well as it should. The block diagram of the proposed predictive maintenance system is depicted in Fig. 4. The proposed IoT-enabled predictive maintenance system displays multiple integrations of sensing, capturing, communicating, and displaying the output of XAI models via an intuitive dashboard. The sensors on the pump system continuously sense system parameters such as pressure, vibration, current, and ultrasonics and transmit them to the microcontroller unit. This time-series sensor data is sent through a communication gateway to our dedicated cloud, where it is monitored, and XAI models are

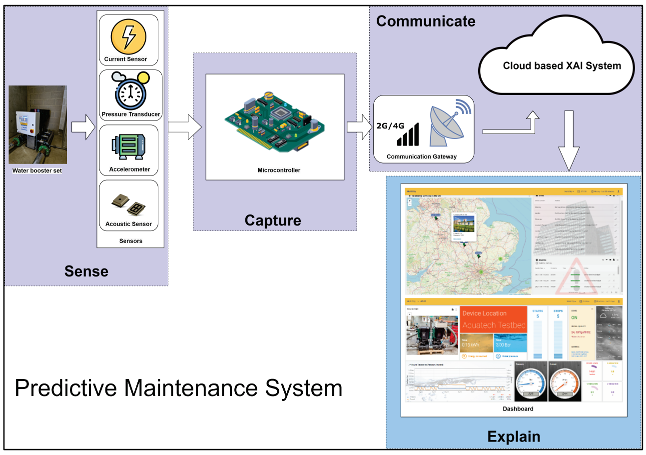


Fig. 4. Proposed PdM system

used to detect any potential issues. The user-friendly dashboard is employed to track trends and make AI model output more understandable to service engineers.

This Internet-of-Things enabled predictive maintenance system has the advantage of integrating the natural environment, such as pumping systems with sensors, onto the internet, allowing a human to monitor the equipment, thereby facilitating the human-machine interaction.

A. *Sensors & Transducers*

We are using a variety of sensors to collect operating data from the pump system for analysis. Depending on the pump system, some sensors might already be available or added as required. Alternatively, it is also possible to harvest data from the control system itself in some cases. Sensors and transducers measuring the current, pressure, vibration, and ultrasonics are all embedded in our hardware, depicted in Fig. 5 (a-d), respectively.

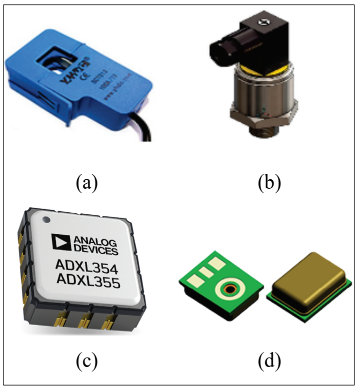


Fig. 5. (a) Current Sensor; (b) Pressure Transducer; (c) Vibration Sensor; (d) Ultrasonic Sensor

The *Current Sensor* is a simple install alternating current sensor with a maximum current detection rating of up to 50 Amperes and is suitable for various applications. The electromagnetic induction method is used to make the measurement. Analogue signals are generated from the measured output alternating current and are supplied into the microcontroller for further processing. This current sensor also calculates dissipated electricity (watts) and used energy (kWh).

The *Pressure Transducer* has a detection range of up to 25 bars. The pressure delivered to the transducer causes a diaphragm deflection, causing strain on the gauges. The tension causes a variation in electro-pressure resistance. The pressure transducer transforms this electro-pressure resistance into an analogue electric signal which is sent into the microcontroller.

The *Vibration Sensor* is a high-performance digital 3D sensor explicitly designed for Industry 4.0 applications. A whole acceleration range of $\pm 2/\pm 4/\pm 8/\pm 16g$ is possible. It has a dedicated programmable signal processing pipeline with low latency, low noise, and reliable filtering for control loop stability. The vibration sensor is connected to the microcontroller to simplify the processing of complicated signals. It assists us in measuring the slightest vibrations produced by the pump system in our experiments.

Our studies employed an ultra-compact, low-power, omnidirectional digital MEMS (*Ultrasonic Sensor*) with a capacitive sensing element and an IC interface. It uses a specific silicon micromachining method to fabricate the sensor device, emitting ultrasonic waves.

B. *Microcontroller & Communication board*

As shown in Fig. 6, a powerful RISC microcontroller board and a cellular-based connectivity board are utilised. The microcontroller processes the gathered data from the abovementioned sensors, and the cloud receives this telemetry data using cellular-based technology available on the connectivity board.

The board also includes a micro-SD card for storing off-line data logging of telemetry data, which can be utilized for data analytics on historical data.

C. *Cloud & the communication protocol*

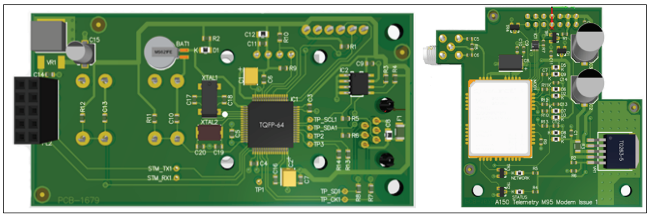


Fig. 6. Microcontroller and Communication board

The 'things' in the IoT have memory constraints, processing power constraints, and even energy consumption issues. As a result, most of the processing occurs on the server-side, which is nothing more than a powerful PC linked to the internet. This is known as a private cloud. In this research, we created our cloud (as illustrated in Fig. 7). We hosted it on the public internet to enable the hardware and communication board to communicate the essential telemetry data for processing. It comprises three components: a rule engine, a database, and a graphical user interface (GUI). Multiple communication protocols are used for data transmission on the Internet of Things and Machine-to-Machine (M2M) systems. MQTT is an open OASIS and ISO standard (ISO/IEC PRF 20922) for client-server, publish/subscribe messaging [21]. This paper utilises this standard because it is lightweight, open, straightforward, and simple to implement.

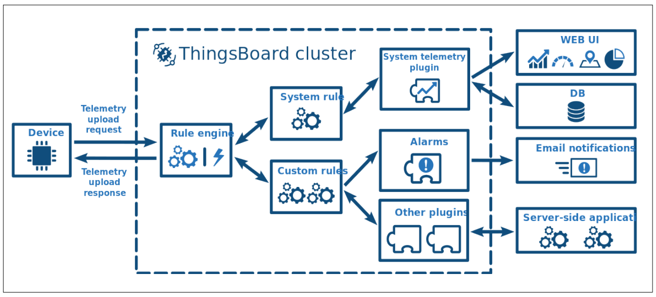


Fig. 7. Cloud Architecture [20]

V. The Proposed Optimised Type-2 Fuzzy Based System For Explainable Predictive Maintenance

The design of our intelligent explainable system for predictive maintenance is shown in this section. The data acquisition and pre-processing, off-line data analysis and learning phase, dynamic monitoring and prediction phase, explainable framework and decision support modules are explained in the following sub-sections.

A. *Data Acquisition & Pre-processing*

The data collection module initiates the system's operation by gathering sensor data from the sensors listed above. The data is acquired from all the sensors by the microcontroller for data pre-processing. The IoT data generated by sensors are essentially sequential and immense. Sometimes the sensor data values are mostly repeated, and noises with irregular values may go undetected. Therefore, it is necessary to pre-process the acquired sensor data before communicating it to further processing and to train the machine learning models. Pre-processing is a straightforward solution that is typically fast and lightweight, making it well-suited for dynamic IoT settings. The IoT data generated by sensors are essentially sequential and immense. Sometimes the sensor data values are mostly repeated, and noises with irregular values may go undetected. Therefore, it is necessary to pre-process the acquired sensor data before communicating it to further processing and to train the machine learning models. Pre-processing is a straightforward solution that is typically fast and lightweight, making it well-suited for dynamic IoT settings. In [40], the author proposes two sensor signal processing techniques – segmentation and labelling algorithm and compressive sensing technique. Both methods are for systems with limited memory and bandwidth, which does not apply to our system due to availability of advance microcontroller. Moreover, the compressive sensing technique assumes a sparse signal, which is not in our case; we use data sampling techniques based on windowing before applying the fast Fourier transform. In this paper, we have pre-processed the data from the sensors before communicating the essential telemetry data to our cloud for further processing. The vibration and ultrasonic sensor data are pre-processed using the Fast Fourier transform, while standard statistical methods pre-process the current and pressure sensor data. In particular, data from vibration and ultrasonic sensors is collected in the time domain and translated to the frequency domain using a digital signal processing algorithm such as the Fast-Fourier transform (FFT) used on-board by our advanced secondary microcontroller.

Each rotating machine has its unique vibration signature due to the combination of its distinct components. When the machine encounters unexpected activity, such as unbalance or bearing wear, the amplitude of that signature varies. The machine's entire vibration is measured in the time domain, and it is mostly the sensation we can feel when we put our hands on the machine. It is also the sum of vibrations of each component.

When performing vibration analysis in the frequency domain, it's possible to track down fault causes more efficiently before the equipment fails catastrophically. The cube is shown in Fig. 8 frequently serves as a visual representation of the time and frequency domains. The time-domain measurements track the signal's amplitude across time, whereas the frequency domain measurements track the signal's amplitude at each frequency. The Fast Fourier

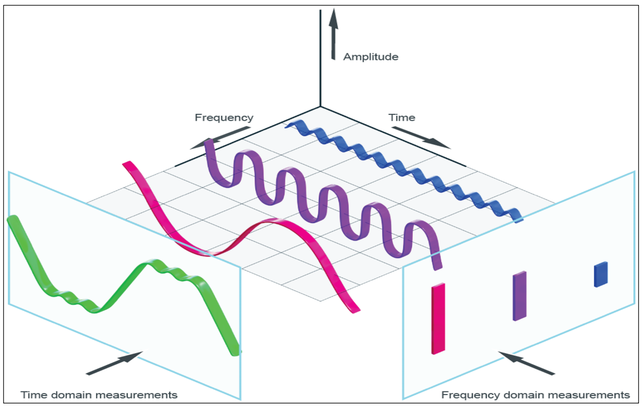


Fig. 8. Cube of time and frequency domain [19]

transform (FFT) decomposes the overall vibration signal into its constituent frequencies. Although the involved mathematics is fairly hard, our microcontroller can make the

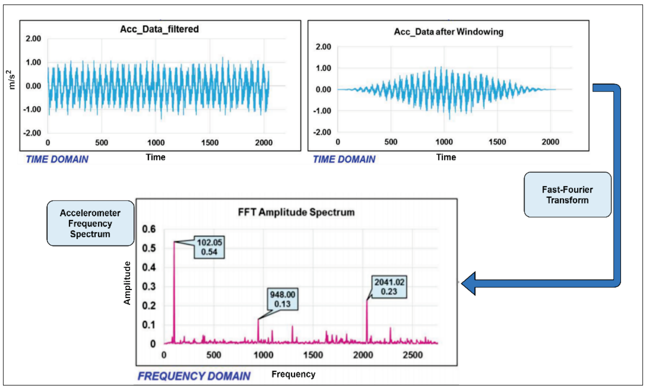


Fig. 9. Time domain to frequency domain of vibration data

calculations. Fig. 9 shows the filtered acceleration data from our vibration sensor, the overall vibration. This shows the data after windowing, and individual component's frequency domain signals obtained through FFT analysis on-board. Similarly, ultrasonic signals from our ultrasonic sensor are also processed from time-domain to frequency-domain. Each datum consisting of current, pressure, vibration and ultrasonic sensor data is then communicated to the cloud, where it is stored in a database.

B. *Off-line Data Analysis & Learning phase*

Fig.10 shows an overview of the proposed XAI based PdM. The database feeds the offline Data Analysis and learning phase module, which executes knowledge generation using advanced data analytics and cloud technologies combined with the acquired telemetry input/output data to extract the rules required for the fuzzy logic system. The original approach for extracting fuzzy rules provided in this study is based on a type-2 extension of L. X. Wang's Wang–Mendel method [23][24]. The rule extraction methodology is a one-step procedure for obtaining fuzzy rules from data. Given a collection of data pairs with various inputs and outputs,

$$\left(x^{(t)}; y^{(t)}\right), \quad t = 1,2, \ldots., N \tag{1}$$

where N is the number of data instances, $x^{(t)} \in R^n$ and $y^{(t)} \in R^k$. n *and* k are input and output variables respectively.

We derive the multiple-input-multiple-output rules that define the relationship between $y = (y_1, \ldots, y_k)$ and $x = (x_1, \ldots x_n)^T$ and take the following form:

IF x_1 is $\tilde{A}_1^l$…and x_n is $\tilde{A}_n^l$ THEN y_1 is $\tilde{B}_1^l$…and y_k is $\tilde{B}_k^l$ (2)

where $l = 1,2 \ldots, M, M$ is the number of rules and l is the index of the rules. There are V_i interval type-2 fuzzy sets $\tilde{A}_s^q$, q = 1,…, V_i, defined for each input x_s where $s = 1, \ldots, n$. There are V_o interval type-2 fuzzy sets $\tilde{B}_c^h, h = 1, \ldots, V_o$, defined for each output y_c where $c = 1, \ldots, k$.

After producing the M rules from the user data, we can get an approximated model of the equipment in the environment for a given period of time. We optimise the rule base size by using the Big Bang-Big Crunch (BB-BC) technique to avoid rule base explosion and its associated consequences on memory and CPU capacity. Erol and Eksin proposed the BB-BC optimisation, and [25] provides a comprehensive

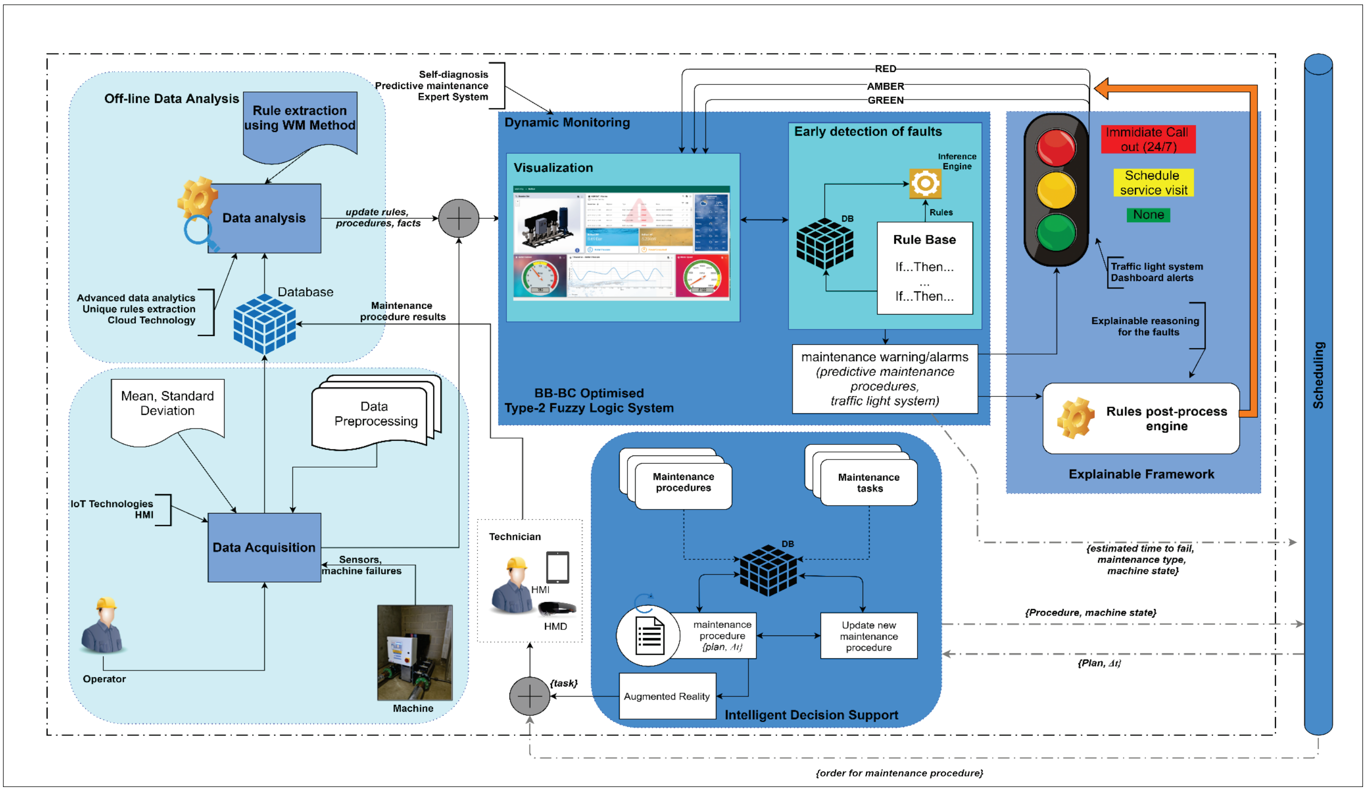


Fig. 10. Intelligent explainable system architecture

overview.

1) Optimizing the Rule Base of the IT2FLS with BB-BC

To apply the BB-BC, the rule base's parameters are encoded as a population as shown in Equation (3):

$$R_1^1 R_2^1, \dots, R_1^1, R_a^1, \dots, R_1^M R_2^M, \dots, R_a^M, R_{out}^M \quad (3)$$

where, R_k^m are the antecedents and R_{out}^m is the consequent of each rule respectively, m =1…, M, M is the number of rules and k = 1,…, a, a is the number of antecedents to be tuned. We use the following function as the cost function for BB-BC where RMSE is the Root Mean Square Error.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\|x_i - \hat{x}_i\|^2} \quad (4)$$

where m = size of the training set, x_i is the desired output and $\hat{x}_i$ is the actual output of the FLS.

2) Optimizing Type-2 membership functions with BB-BC

The type-2 Membership Function (MF) feature parameters must also be encoded into a population form to use BB-BC. We have used three fuzzy sets $(LOW, MEDIUM, HIGH)$ to model each of the inputs and parameters of the output MFs are also encoded for linguistic variables $(NORMAL, MARGINAL, SEVERE)$ [26] as shown in Equation (5):

$$\rho_1^1, \rho_2^1, \rho_3^1, \dots, \rho_1^q, \rho_2^q, \rho_3^q, \rho_1^{out}, \rho_2^{out}, \rho_3^{out} \quad (5)$$

where the superscript represents the number of antecedents, and the subscript represents the number of fuzzy sets. Given that the optimization problem is a minimization task and given the parameters of the MFs encoded as shown in Equation (5) and the constructed rule base, it is possible to minimize the error in our solutions space by utilizing Equation (4) as the cost function.

A smaller set of rules, as well as a smaller number of antecedents per rule leads to a search area that is not well covered. In some cases, this can result in no rules being fired because specific inputs from the present rule base are being ignored. [27]. As a result, we employed the similarity metric described in [28] in order to generate output from an incomplete rule base. To measure the similarity in the antecedent elements between the rule created by the input $\boldsymbol{x}_i^{(t)}$ and each rule $\boldsymbol{R}_m$, a function distance is constructed as $\mathcal{D}\left(\mathcal{A}_i^{k*}, \mathcal{A}_i^{(m)}\right)$, where $\mathcal{A}_i^{k*}(\boldsymbol{x}_i^{(t)})$ represents the fuzzy set matched by an input $\boldsymbol{x}_i^{(t)}$ and $\mathcal{A}_i^{(m)}$ represents the antecedent fuzzy set for rule m. For instance, if we have three fuzzy sets $(LOW, MEDIUM, HIGH)$ to model each of the inputs; each label can be encoded as an integer, whereby LOW is 1, $MEDIUM$ is 2 and $HIGH$ is 3. So $\mathcal{D}(LOW, LOW) = 0$, $\mathcal{D}(HIGH, LOW) = 2$, $\mathcal{D}(MEDIUM, HIGH) = 1$ and so on.

We establish a distance that determines the difference between the coded linguistic labels with this goal in mind. Using this distance, the similarity [28] between the rules created by input $\boldsymbol{x}_i^{(t)}$ with each rule $\boldsymbol{R}_m$ is calculated as in equation (6):

$$S\left(x^{(t)}, R_m\right) = \frac{\sum_{i=1}^{n}\left(1-\frac{\mathcal{D}\left(\mathcal{A}_i^{(t)}, \mathcal{A}_i^{(m)}\right)}{V-1}\right)}{n} \quad (6)$$

where $\boldsymbol{S}\left(\boldsymbol{x}^{(t)}, \boldsymbol{R}_m\right) \in [\boldsymbol{0}, \boldsymbol{1}]$, $\boldsymbol{V}$ is the number of fuzzy sets and $\boldsymbol{i} = 1, \dots, \boldsymbol{n}$, where $\boldsymbol{n}$ denotes the number of input values, which corresponds to the number of rule antecedents [28].

Assume there are just two rules in the rule base: $R_1\{LOW, LOW, LOW\}$ and $R_2\{HIGH, HIGH, HIGH\}$. In case the incoming crisp input vector $\boldsymbol{x}^{(t)}$ yields a fuzzy rule $\{LOW, LOW, HIGH\}$ which we cannot locate in the rule base, then similarity S for R_1 and R_2 are $S_{(x,R_1)} = \frac{\left(1-\frac{\mathcal{D}(LOW,LOW)}{3-1}\right)+\left(1-\frac{\mathcal{D}(LOW,LOW)}{3-1}\right)+\left(1-\frac{\mathcal{D}(LOW,HIGH)}{3-1}\right)}{3} = 0.33$ and $S_{(x,R_2)} = \frac{\left(1-\frac{\mathcal{D}(LOW,HIGH)}{3-1}\right)+\left(1-\frac{\mathcal{D}(LOW,HIGH)}{3-1}\right)+\left(1-\frac{\mathcal{D}(LOW,HIGH)}{3-1}\right)}{3} =$

0. So, in this case $S_{(x,R_1)}$ is bigger, then this crisp input $\boldsymbol{x}$ applies the rule $R_1\{LOW, LOW, LOW\}$ with its relevant similarity value as the matched rule.

C. *Dynamic Monitoring & Prediction phase*

The Dynamic Monitoring and prediction phase module comprises two components: The Early Detection of Faults component and the Visualisation component. When the system is operating in real-time, we collect telemetry data from the hardware to monitor its performance. Feature vectors are derived from the telemetry data and utilised as input values for the IT2FLS-based system. The associated rules specify each fault in our fuzzy system. Each output degree shows the probability of the projected fault being the same as the trained fault model in the knowledge base.

The fuzzy system has a_1, a_2, a_3 as antecedents and each of these is modelled by three fuzzy sets: $LOW, MEDIUM$ & $HIGH$. The output of the fuzzy system i.e., chance of fault occurrence is modelled by three fuzzy sets: $NORMAL, MARGINAL$ & $SEVERE$.

A type-2 singleton fuzzifier is used to fuzzify the crisp input and acquire the upper $\bar{\mu}_{\tilde{A}_s^q}\left(x_s^{(t)}\right)$ and lower $\underline{\mu}_{\tilde{A}_s^q}\left(x_s^{(t)}\right)$ membership values for the specified crisp input vector. After that, the firing strength $\underline{f}^i$ and $\bar{f}^i$ of each rule, where $i = 1, \dots, M$, and M is the number of rules and $\underline{f}^i(x') = \underline{\mu}_{\tilde{F}_1^i}(x_1') * \dots * \underline{\mu}_{\tilde{F}_1^i}(x_p')$ and $\bar{f}^i(x') = \bar{\mu}_{\tilde{F}_1^i}(x_1') * \dots * \bar{\mu}_{\tilde{F}_1^i}(x_p')$ is calculated. The Karnik-Mendel (KM) approach [28] is used to compute the type reduced set specified by the interval $[y_{lk}, y_{rk}]$, which is then used to perform the type reduction process. After the procedure, defuzzification is computed as $\frac{y_{lk}+y_{rk}}{2}$ in order to identify the output degree of the predicted fault.

The visualisation component receives inputs from the data acquisition module, which provides data for real-time monitoring of specified equipment variables on an intuitive user dashboard based on JavaScript. In contrast, the early detection of faults component activates the subsequent action, specifically a notification of the need for maintenance measures. When a need for maintenance interventions is detected, the scheduling tool sends the information to the Decision Support module.

D. *Explainable Framework*

The explainable framework is a novel framework that includes a rules post-processing engine and a traffic light system. In our IT2FLS-based system, explainability is a key value proposition. Table I shows some of the extracted rules to illustrate the IT2FLS's explainability benefits. Rule 1 states if the 5hrs amplitude of vibration is LOW and 8hrs score of fault occurrence is $MEDIUM$ and 4hrs pump pressure is LOW then the chance of pump failure is $NORMAL$.

Similarly, Rule 2 shows, if the 6hrs pump pressure is $MEDIUM$ and 9hrs amplitude of vibration is $HIGH$ and 7hrs score of fault occurrence is $HIGH$ then the chance of pump failure is $SEVERE$. The rules post-processing engine further analyses these rules to provide the service engineer with detailed explanations. For example, if the rule consequent for the pump failure fault is $SEVERE$, the rules post-processing engine examines the antecedents to determine the reasons behind that consequent (chance of pump failure). This is accomplished by introducing an additional layer of processing, as shown in Fig. 11, in which an antecedent containing a vibration level is selected as the dominant antecedent for the pump failure fault because vibration levels are the most important parameter (as they have the highest

	Dominant				
Consequent	Vibration Level	Pump pressure	Score of fault	Meaning for Vibration Level	Meaning for Pump pressure
Severe	High	Low	High	Vibration level has increased	Under limits
	High	Medium	High	Vibration level has increased	Slightly increased
	High	High	High	Vibration level has increased	Pressure range has increased

Fig. 11. Selection of dominant antecedent, High

average of lower and upper membership values) for accurately monitoring pump failures and their associated issues. The pump failure consequent (Normal, Marginal or Severe) is chosen as follows: For each defuzzified output $y_o^{(t)}$ we match it against its output type-2 MFs $\mu_{\tilde{A}_o^q}\left(y_o^{(t)}\right)$ which are defined by FOU $(\tilde{A}_o^q)$. The upper and lower MFs for FOU $(\tilde{A}_o^q)$ are $\bar{\mu}_{\tilde{A}_s^q}\left(y_o^{(t)}\right)$ and $\underline{\mu}_{\tilde{A}_o^q}\left(y_o^{(t)}\right)$ for each fuzzy set $q = 1, \dots, V_o$, and for each output variable $o = 1, \dots, 4$ respectively. We calculate the $q * \varepsilon\{1, \dots, V_i\}$ such that

$$\mu^{cg}_{\tilde{A}_o^{q*}}\left(y_o^{(t)}\right) \geq \mu^{cg}_{\tilde{A}_o^{q}}\left(y_o^{(t)}\right), \text{for all } q = 1, \dots, V_o \tag{7}$$

where, the centre of gravity of the interval membership of $\tilde{A}_o^q$ at $y_o^{(t)}$ is $\mu^{cg}_{\tilde{A}_o^{q*}}\left(y_o^{(t)}\right)$. This is obtained by averaging upper and lower MFs as:

$$\mu^{cg}_{\tilde{A}_o^{q*}}\left(y_o^{(t)}\right) = \frac{1}{2}\left[\bar{\mu}_{\tilde{A}_o^q}\left(y_o^{(t)}\right) + \underline{\mu}_{\tilde{A}_o^q}\left(y_o^{(t)}\right)\right]. \tag{8}$$

If the vibration levels are $HIGH$, it is assumed that the pump's vibration has risen, and this interpretation is provided to the service engineer. The dominating attribute is chosen based on our laboratory testing as well as our experience in the water technology business.

In the same way, if the FLS pump failure crisp output is classified under $NORMAL$ operation as illustrated in Fig.12, the rule post-processing engine examines the Vibration antecedent as LOW, indicating that the pump's vibrations are within acceptable limits.

	Dominant				
Consequent	Vibration Level	Pump pressure	Score of fault	Meaning for Vibration Level	Meaning for Pump pressure
Normal	Low	Low	Low	Vibration level is under limits	Under limits
	Low	Medium	Low	Vibration level is under limits	Slightly increased
	Low	High	Low	Vibration level is under limits	Pressure range has increased

Fig. 12. Selection of dominant antecedent, Low

The traffic light system uses a dictionary (shown in Fig. 13) stored in a database to further process the FLS model's outputs according to the fault levels of $NORMAL$, $MARGINAL$, and $SEVERE$. The dictionary's purpose is to evaluate interpretations based on the type of fault and provide guidance to the service engineer on what to do next. We used AGM Plc's 50 years of water technology experience to verify that the dictionary's interpretation was correct. The three traffic lights, *green*, *amber*, and *red* indicate the type of fault, as well as whether maintenance should be performed or not based on reasoning/explainability, and, ultimately, what maintenance procedure the service engineer should use.

TABLE I
SOME RULES GENERATED BY THE IT2FLS

Rule	Antecedents			Consequents
1	5hrs *amplitude of vibration* is Low	8hrs *score of fault occurrence* is Medium	4hrs *pump pressure* is Low	Then *chance of pump failure in next 4hrs* is Normal
2	6hrs *pump pressure* is Medium	9hrs *amplitude of vibration* is High	7hrs *score of fault occurrence* is High	Then *chance of pump failure in next 4hrs* is Severe

When the FLS system predicts a *Pump Failure* fault, the explainable framework, and the extracted rules in Table I indicate that *if the chance of pump failure is severe*, a red traffic light will illuminate, indicating the need for immediate call-out (24x7) maintenance. Additionally, informing the

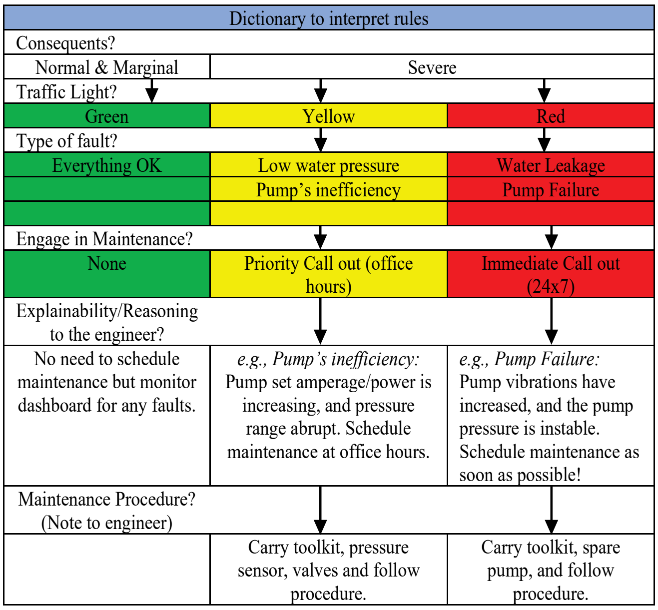


Fig. 13. Rules post-processing dictionary

service engineer that those vibrations have increased, and the pressure has become unstable indicates that pump failure is possible. He should be prepared with spare pump tools and follow the procedure provided by the decision support module.

Similarly, if the pump is running inefficiently, a yellow traffic light will illuminate, indicating that maintenance is required during business hours. Also, remind the service engineer to carry a toolkit, pressure sensor, and valves and follow the procedure. A green traffic light illuminates the dashboard when everything is in working order, and the model predicts no future problems.

In Fig. 14, we show our user dashboard developed using JavaScript to present predicted faults with explanations in an easy-to-understand manner, analyse telemetry data trends, and a visually appealing traffic light system. The JavaScript dashboard used in our research is an open source IoT platform called as ThingsBoard [41]. We have a licence to use most of its features and deploy them locally.

This gives a service engineer the ability to keep an eye on the trends in each pump and only intervene in the event of a major failure. Thus, unnecessary trips may be minimised, the total breakdown of a pump can be predicted, the engineer is always equipped with the necessary tools, and the end-user is never without a critical service. After the rule post-

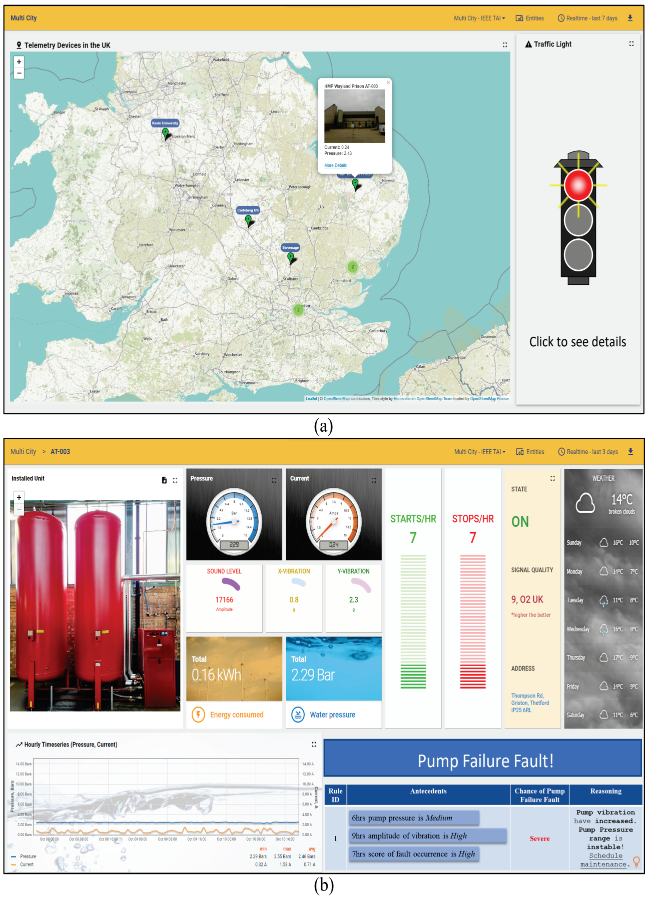
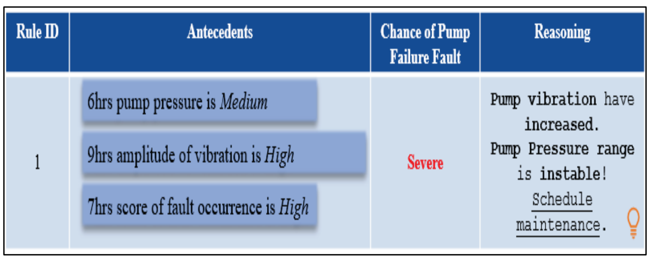


Fig. 14. User Dashboard: (a) Telemetry devices across UK; (b) Individual telemetry unit's data with fault's explanation

processing, Fig. 15 displays the final reasoning provided to the service engineer in an easy-to-understand manner. Finally, the explainable framework is also coupled to the Decision Support module, described in the next subsection, which provides the maintenance procedure to the engineer.

E. Decision Support

The decision support system is the final and essential component of our Intelligent explainable system architecture. It is composed of a database shared with the databases of



Rule ID	Antecedents	Chance of Pump Failure Fault	Reasoning
1	6hrs pump pressure is *Medium* 9hrs amplitude of vibration is *High* 7hrs score of fault occurrence is *High*	Severe	Pump vibration have increased. Pump Pressure range is instable! Schedule maintenance.

Fig. 15. Final reasoning for pump failure fault

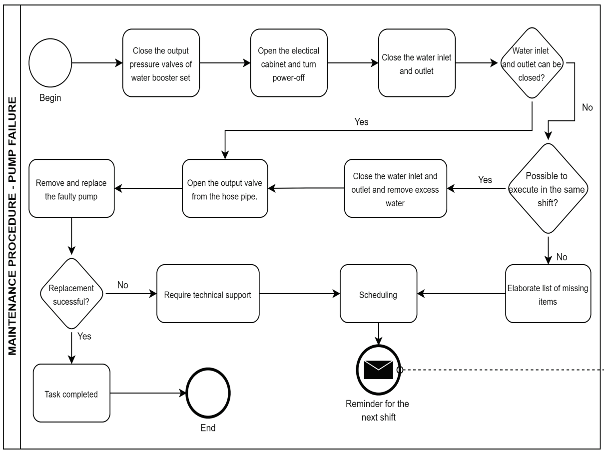


Fig. 16. Business Process Model and Notation

other modules and an engine that generates maintenance procedures. Fig. 16 illustrates the use of formal logical representations to describe maintenance operations, such as BPMN (Business Process Model and Notation), which defines the sequences of single maintenance tasks to be carried out when one of the faults is predicted by a type-2 fuzzy logic system and alerted by the traffic light system for maintenance intervention.

VI. Experiments And Results

The telemetry data we got from our hardware (detailed in Section IV) deployed at different places in the United Kingdom (as shown in Fig. 17) was used as the foundation for our experiments. The data were obtained over a long period using water pump sets, with all parameters for the equipment captured for one hour in each datum. A private cloud described above in our data acquisition hardware system (detailed in Section IV) is an API-driven predictive maintenance platform that facilitated the AI models' creation, configuration, and execution.

Fig. 17. Telemetry data collection from locations in the UK

For our explainable AI-based predictive maintenance system, we report the results of employing Interval Type-2 Fuzzy Logic System (IT2FLS) and comparing it to a Type-1 Fuzzy Logic System (T1FLS), a Multi-Layer Perceptron (MLP) Neural Network, an effective deep learning method known as stacked autoencoders (SAEs) and an interpretable model like decision trees (DT). The Multi-Layer Perceptron (MLP) Neural Network and stacked autoencoders (SAEs) are opaque models that have been shown to be quite accurate in many use cases and research articles. They are frequently used in use cases where the opacity of the model is not a concern. The decision trees are interpretable, but they have the potential to generate overly complicated trees that do not generalise the data well and lack performance. On the other hand, the FL model is a rule-based method with inherent explainability, coupled with evolutionary optimisation to maximise the model's interpretability and performance. Additionally, each model and data instance result may be fully explained in terms of global and local variables.

A. *Results discussion – Pump Failure*

In order to anticipate future pump failure at interval t, we used the 3 inputs (amplitude of vibration signal, pump pressure, score of fault occurrence) at previous time intervals i.e., $x_{t-1}, x_{t-2}, \dots, x_{t-r}$ where x_t contains the 3 inputs mentioned above. The score for fault occurrence is computed using a combination of vibration amplitude and pump pressure and it varies according to the type of fault. The score of fault occurrence for pump failure fault is generated based on the values of the amplitude of vibration, for water leakage fault it is generated based on the amplitude of ultrasonic data, for low water pressure it is generated based on pump pressure data, and finally, for pump inefficiency, it is generated based on current sensor data. The chance of pump failure at time interval t is the system output. We choose r in the range of 2 to 5 as from experience this is the effective range. The performance was determined by utilising the root mean square error (RMSE) in Equation (4) as the cost function. When r was 3, the best performance was noticed. The T1FLS & IT2FLS were optimised using the technique described in Section V. The type-2 membership functions for the pump failure fault (obtained and optimized by BB-BC) are shown in Fig. 19.

This experiment used the BB-BC algorithm with 15 candidates per iteration, a stopping condition of 500 iterations, and the RMSE in Equation (4) as the cost function. We used the same inputs as the IT2FLS FLS in our T1FLS, MLP Neural Network, SAE and DT models. The MLP neural network we used had one hidden layer with ten neurons and a single output neuron, while the SAE had three hidden layers with 90, 45 and 10 neurons in each layer. The MLP neural network is trained using backpropagation and gradient descent, with a learning rate of 0.2. The Rectified Linear Unit (ReLU) activation function is used in the hidden layer to transfer the input value to the desired output value. The SAE was trained with the same properties as the MLP. The DT uses entropy as attribute selection method, and it has the same set of features as the IT2FLS. We employed the cross-validation methodology, a standard tool for model evaluation, which was found to provide a less skewed or pessimistic estimate of the model competence than other approaches, such as a fundamental train/test split. The value of k was chosen to be 5 because it produced sufficiently good results. The models' performances result for 4 faults with k-fold cross validation is shown in Table II.

As shown in Table II, our IT2FLS based predictive system outperforms the T1FLSs based counterpart in predicting the chance of failure, with the IT2FLS system achieving 8.99% greater combined average accuracy over the test data of four faults than the T1FLS. It also outperforms the DT with a 529.21% improvement in accuracy. The output from DT is

TABLE II
OPTIMISED T1FLS, IT2FLS WITH MLP, SAE & DT OVER TESTING DATASET FOR 4 FAULTS

Faults	Model performances result for 4 faults with k-fold cross validation					
	k-folds (k=5)	RMSE T1FLS	RMSE IT2FLS	RMSE MLP	RMSE SAE	RMSE DT
Fault 1	k=1	0.099	0.088	0.056	0.052	0.60
	k=2	0.100	0.089	0.057	0.053	0.61
	k=3	0.098	0.090	0.055	0.051	0.59
	k=4	0.097	0.087	0.057	0.050	0.58
	k=5	0.096	0.088	0.056	0.054	0.60
Mean RMSE for 5 folds		**0.098**	**0.088**	**0.056**	**0.052**	**0.60**
Fault 2	k=1	0.098	0.089	0.057	0.053	0.57
	k=2	0.097	0.090	0.058	0.051	0.56
	k=3	0.099	0.088	0.056	0.054	0.56
	k=4	0.100	0.089	0.055	0.053	0.57
	k=5	0.097	0.090	0.058	0.052	0.55
Mean RMSE for 5 folds		**0.098**	**0.089**	**0.057**	**0.053**	**0.56**
Fault 3	k=1	0.096	0.087	0.056	0.051	0.55
	k=2	0.095	0.088	0.058	0.052	0.54
	k=3	0.098	0.090	0.057	0.050	0.53
	k=4	0.097	0.089	0.056	0.051	0.57
	k=5	0.100	0.087	0.058	0.053	0.58
Mean RMSE for 5 folds		**0.097**	**0.088**	**0.057**	**0.051**	**0.55**
Fault 4	k=1	0.099	0.090	0.058	0.053	0.57
	k=2	0.095	0.091	0.059	0.054	0.56
	k=3	0.098	0.090	0.057	0.052	0.54
	k=4	0.093	0.088	0.056	0.051	0.55
	k=5	0.096	0.089	0.060	0.050	0.56
Mean RMSE for 5 folds		**0.096**	**0.090**	**0.058**	**0.052**	**0.55**

shown in Fig. 20, where the tree is unpruned, provides an approximate explanation of faults and conveys a different result from what service engineers will get in a normal event offering no insights for the service engineer. Also, as the number of inputs increases, the number of nodes in the decision tree goes up quickly, making the tree hard to understand. Although the MLP and SAE models have done well in fault prediction achieving higher combined average accuracy, they are hard to understand when compared to our SAE inputs from the predicted outputs of a model, we find that the neurons' information is compressed after the activation functions are applied, making it very hard to decode. In the service-critical water pumping business, explainability should take precedence. Hence the proposed IT2FLS satisfied the trade-off between explainability and accuracy when compared to MLP and SAE models.

Corporations cannot afford disruptions in the supply of water. In the situation above, when a pump failure is predicted, the explainability framework enables the service engineer to understand why the fault happened, how essential it is to perform maintenance, and what spare parts he needs to bring with him. Thus, the IT2FLS has the benefit of having a model that can explain, ensuring that the service engineer understands how the system predicted a pump failure. In addition to a high level of explainability, the IT2FLS uses a rule base that can be understood and analysed, allowing the experts to enhance and complement the model with their experience by adopting or amending rules.

Explainable AI (XAI) models are used to describe rule-based FLS models, as previously stated in [13] because the proposed system includes most of the XAI components from [34]. The rule-based FLS built can communicate in the same languages as humans. [35], allowing any service engineer to analyse and interpret the model output [13] easily. Furthermore, engineers can improve the model by integrating their expertise in the rules [13].

B. Results discussion – Building trust

We conducted a survey with experienced service engineers from the water pumping industry to validate our deployed system and the results. The FLS and MLP Neural Network models and outputs were evaluated with service engineers to determine the level of trust provided by these two algorithmic approaches.

The models were tested against some of the common faults in pumping systems. The rules-post processing engine results were shared with service engineers so we could evaluate their level of trust in five different categories: *strongly disagree*,

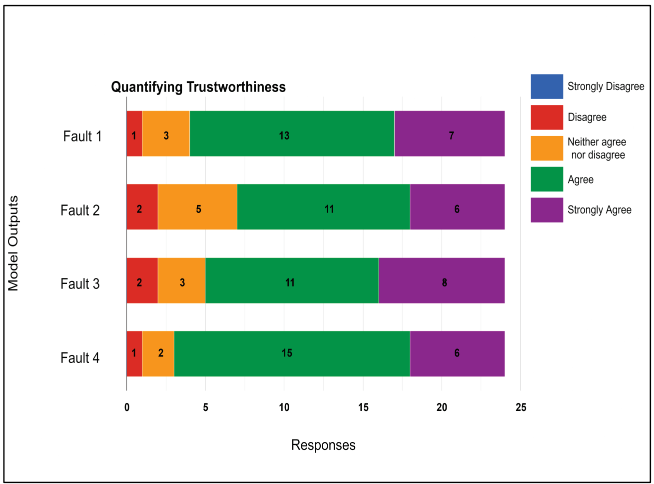


Fig. 18. Quantifying trustworthiness for XAI models

disagree, *neither agree nor disagree*, *agree* and *strongly disagree*. The quantified level of trustworthiness is shown in Fig. 18. A total of 24 responses were collected, with 80.3% strongly agreeing and agreeing, 13.5% neither agreeing nor disagreeing, and 6.2% disagreeing with the explanation presented to them. Because of this added benefit of explainability, the FLS model can enable establishing faith in the AI and decisions (as it produces comparable results to opaque box models like deep learning while producing transparent models and decisions allowing end users to analyse and understand them), which is critical in the water pumping business and other comparable utility-based industries that affect millions of people across the world more readily.

C. *XAI Results discussion – Gaining insights*

Explanations are typically sought in the context of AI-assisted decision-making to get further insights or evidence, as humans are not content with simply reading a recommendation or score provided by the AI. Explainability could aid decision-making confidence or offer causation hypotheses to follow-up actions. The fuzzy rules post-processing system utilizes a collection of fuzzy rules in the form of a dictionary, illustrated in Figure 13, to interpret the output of a machine learning model and offer a clear explanation of the model's decision to humans. This system employs fuzzy rules to evaluate different faults and instruct a service engineer on the necessary steps to take. This approach offers transparency and understandability to the complex decision-making process of the machine learning model, particularly in critical or high-stakes scenarios such as diagnostics, troubleshooting, and other technical fields. The system relies on the fuzzy rules dictionary, which is based on expert knowledge, to interpret the output of the machine learning model and provide a human-readable explanation of the decision. For instance, if the machine learning model outputs a high probability of a specific fault, the fuzzy rules post-processing system can use the dictionary to provide a detailed explanation of the fault, the actions the service engineer should take to fix it, and the necessary parts to bring for maintenance. The integration of XAI in predictive maintenance strategy can provide extra benefits that can have a positive effect on the modern environment, society, and economic sustainability. By utilizing XAI, organizations can optimize the usage of their equipment and decrease unnecessary downtime, resulting in decreased energy consumption and lower carbon emissions. Additionally, XAI-based predictive maintenance can lead to cost savings by reducing the need for unplanned maintenance and downtime. Our findings indicate that by utilising suitable tools and technology, only necessary, adequate, and correct maintenance may be performed using real-time predictions and diagnostics. Thus, reducing the service visits, time, and wastage of fuel. In conjunction with good environmental cognisance, correct maintenance contributes significantly to decreasing accidents by detecting and avoiding problems early and remotely. This fosters a safety culture, supports safe-conduct, and ensures a safe and healthy workplace. Hence, positive impact on society.

An intuitive user interface for remote monitoring and prediction-based fuzzy logic models helps in gaining insights about the health of equipment remotely. This may drastically lower service delivery costs and almost eliminates unexpected maintenance leading to lower operating expenses; lower travel expenditures, carbon footprint; and fewer shutdowns.

VII. Conclusions And Future Work

In this paper, we introduced a novel explanation framework for a fuzzy rule-based system. We added a rule post-processing engine in our explanation framework, which uses the dictionaries to provide straightforward explanations from the model consequents. Additionally, we demonstrated how faults could activate the traffic light system, which can be viewed on the intuitive dashboard. This enables a service professional to be well informed about any future faults and how he should respond and be well equipped.

We suggested that the type-2 Fuzzy Logic system's accuracy can be compared to the opaque box model MLP neural network and effective deep learning methods like stacked autoencoders (SAEs). We also compared our model with an interpretable model like decision trees which performed poorly on our dataset with an 529.21% decrease in accuracy compared to our proposed model. The decision tree only can provide an approximate explanation for the predicted failures. The number of nodes in the decision tree also grows exponentially with the number of inputs, which makes the tree confusing to humans. However, the IT2FLS provides highly accurate human-comprehensible interpretations of the current maintenance operations model. Being an explainable model assures that a service engineer can readily comprehend how the system surpassed the black-box models in predicting the future performance of a pumping system. The type-2 Fuzzy Logic system employs a rule base that can be comprehended and analysed in addition to its high degree of explainability, allowing human professionals to improve and augment the model with their knowledge by integrating or changing rules. We surveyed experts in the water industry and received 24 responses, with 80.3% strongly agreeing and agreeing with the explanation presented, demonstrating trust in the explainable AI decisions. Additionally, we discussed how the predictive maintenance method described in this work, when combined with XAI, may be utilised to develop, improve, and apply new tools and strategies for gaining additional insights from the model. The sustainability dimensions are explored and the benefits they provide to the environment, society, and businesses on an environmental, social, and economic level.

Our future research will focus on enhancing the system's prediction ability in more demanding indoor and outdoor environments and expanding our approach to larger industries. Additionally, we will look into self-learning models that can adapt to unfamiliar environments independently.

REFERENCES

[1] U. M. Al-Turki, T. Ayar, B. S. Yilbas, and A. Z. Sahin, “Maintenance in manufacturing environment: An overview,” in Integrated Maintenance Planning in Manufacturing Systems, Cham: Springer International Publishing, 2014, pp. 5–23.

[2] J. Chen, R. Zhang, and D. Wu, “Equipment maintenance business model innovation for sustainable competitive advantage in the digitalization context: Connotation, types, and measuring,” Sustainability, vol. 10, no. 11, p. 3970, 2018.

[3] L. Pintelon and A. Parodi-Herz, “Maintenance: An Evolutionary Perspective,” in Springer Series in Reliability Engineering, London: Springer London, 2008, pp. 21–48.

[4] P. Poor, J. Basl, and D. Zenisek, “Predictive Maintenance 4.0 as next evolution step in industrial maintenance development,” in 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE), 2019.

[5] I. Roda, M. Macchi, and L. Fumagalli, “The future of maintenance within industry 4.0: An empirical research in manufacturing,” in Advances in Production Management Systems. Smart Manufacturing for Industry 4.0, Cham: Springer International Publishing, 2018, pp. 39–46.

[6] Deloitte Analytics Institute, “Predictive Maintenance Taking pro-active measures based on advanced data analytics to predict and avoid machine failure,” Deloitte Analytics Institute. [Online]. Available: https://www2.deloitte.com/de/de/pages/deloitte-analytics/articles/predictive-maintenance.html.

[7] J. Nittke, “Proactive maintenance-downtime reduction,” Japan Tappi Journal, vol. 59, no. 9, pp. 1289–1292, 2005.

[8] J. Zhang, P. Wang, R. Yan, and R. X. Gao, “Long short-term memory for machine remaining life prediction,” Journal of Manufacturing Systems, vol. 48, pp. 78–86, 2018.

[9] X. Li, Q. Ding, and J. Q. Sun, “Remaining useful life estimation in prognostics using deep convolution neural networks,” Reliability Engineering and System Safety, vol. 172, pp. 1–11, 2018.

[10] B. Zhang, S. Zhang, and W. Li, “Bearing performance degradation assessment using long short-term memory recurrent network,” Computers in Industry, vol. 106, pp. 14–29, 2019.

[11] “Scalable Predictive Maintenance Solutions,” Senseye.io. [Online]. Available: https://www.senseye.io/.

[12] L. Liao, “A reconfigurable watchdog Agent® for machine health prognostics,” International Journal of COMADEM, Vol. 11, No. 3, pp. 2–15, July 2008.

[13] H. Hagras, “Towards Human Understandable Explainable AI”, IEEE Computers, Vol.51, No.9, pp. 28-26, September 2018.

[14] R. Chimatapu, H. Hagras, A. Starkey, and G. Owusu, “Interval Type-2 Fuzzy Logic Based Stacked Autoencoder Deep Neural Network for Generating Explainable AI Models in Workforce Optimization,” 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018.

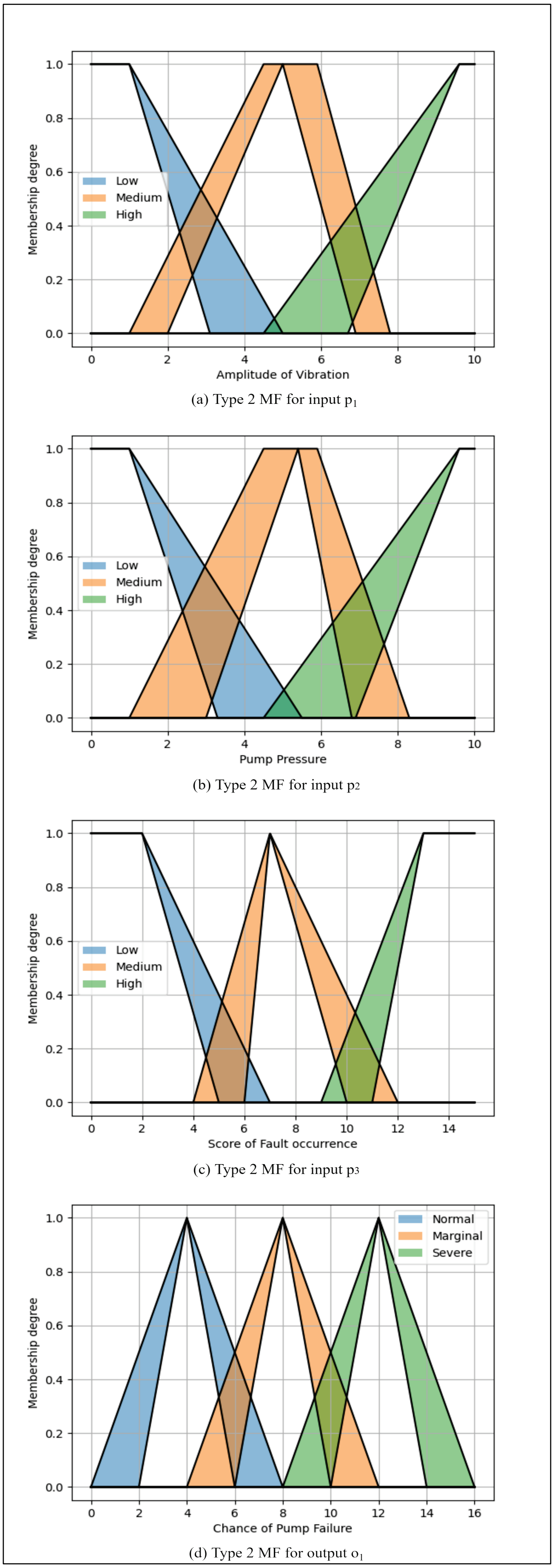
[15] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, ‘‘Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,’’ Inf. Fusion,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

[16] J. M. Mendel, Uncertain rule-based fuzzy logic systems: Introduction and new directions. Philadelphia, PA: Prentice Hall, 2000.

[17] J. M. Mendel, H. Hagras, W.-W. Tan, W. W. Melek, and H. Ying, Introduction to type-2 fuzzy logic control: Theory and applications. Nashville, TN: John Wiley & Sons, 2014.

[18] Patrick Jahnke, Machine learning approaches for failure type detection and predictive maintenance. Technische Universita ̈t Darmstadt, 2015.

[19] “Frequency domain,” Ifm.com. [Online]. Available: https://www.ifm.com/gb/en/gb/landingpages/real-time-maintenance/vibration/technology/frequency-domain.



(a) Type 2 MF for input p_1

(b) Type 2 MF for input p_2

(c) Type 2 MF for input p_3

(d) Type 2 MF for output o_1

Fig. 19. Type-2 membership functions optimized by using BB-BC

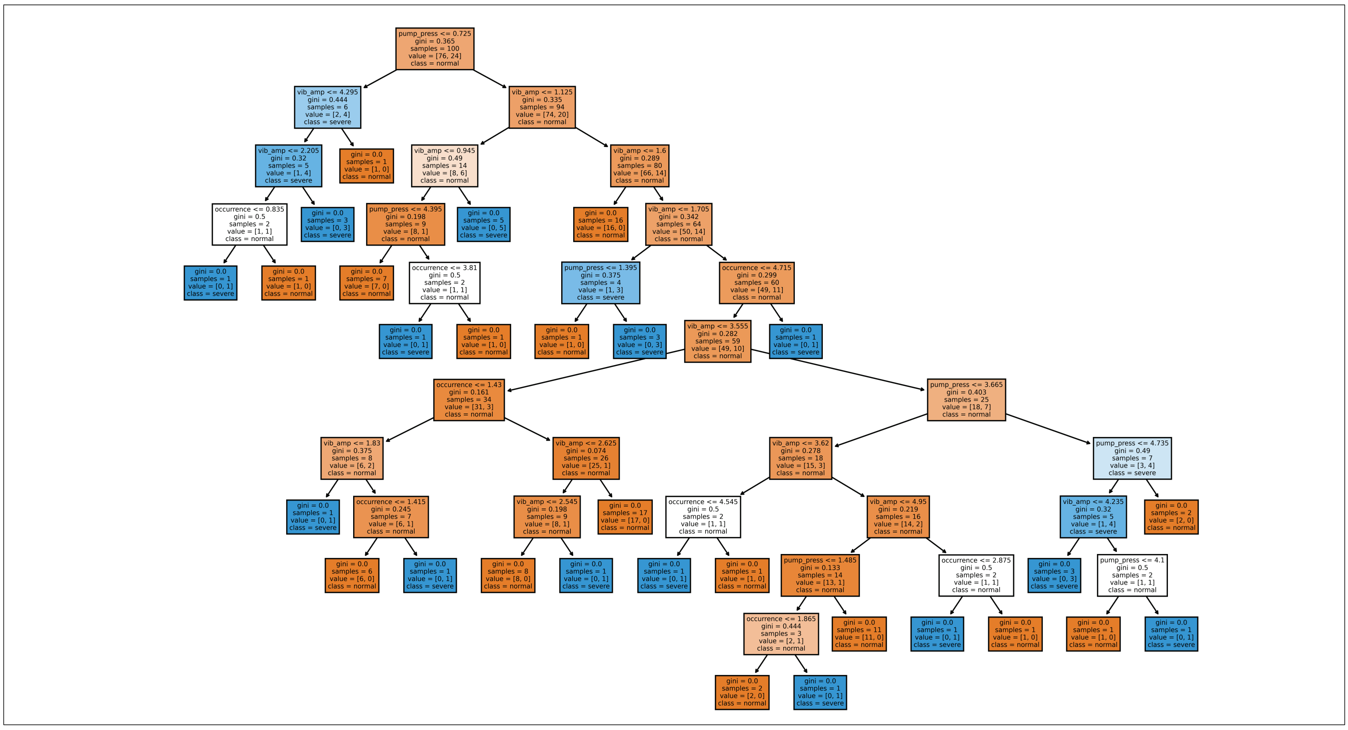


Fig. 20. Example of decision tree populated for fault 1

[20] "ThingsBoard - Open-source IoT Platform," Thingsboard.io. [Online]. Available: https://thingsboard.io.

[21] "MQTT," MQTT Version 3.1.1. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.

[22] 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 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2021.

[23] T. Kumbasar and H. Hagras, "A self-tuning zSlices-based general type-2 fuzzy PI controller," IEEE Transactions on Fuzzy Systems, Vol. 23, No. 4, pp. 991–1013, August 2015

[24] L. X. Wang, "The MW method completed: A flexible system approach to data mining," IEEE Transactions on Fuzzy Systems, Vol. 11, No. 6, pp. 768–782, December 2003.

[25] O. K. Erol and I. Eksin, "A new optimization method: Big Bang–Big Crunch," Advances in Engineering Software, Vol. 37, No. 2, pp. 106–111, 2006.

[26] B. Yao, H. Hagras, M. Alhaddad and D. Alghazawi "A fuzzy logic based system for the automation of human behaviour recognition using machine vision in intelligent environments," Soft Computing, Vol. 19, No.2, pp.499- 506, April 2014.

[27] D. Bernardo, H. Hagras and E. Tsang, "A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications," Soft Computing, Vol.17, No. 12, pp. 2185-2201, August 2013.

[28] T. Garcia-Valverde, A. Garcia-Sola, H. Hagras, J. A. Dooley, V. Callaghan and J. A. Botia, "A fuzzy logic-based system for indoor localization using WiFi in ambient intelligent environments," IEEE Transactions on Fuzzy Systems, Vol. 21, No. 4, pp. 702-718, August 2013.

[29] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, "Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification," IEEE Transactions on Fuzzy Systems, Vol. 25, No. 2, pp. 249–264, April 2017.

[30] A. Sakalli, T. Kumbasar, E. Yesil and H. Hagras, "Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system," Proceedings of the 2014 IEEE International Conference on Fuzzy Systems, Beijing, China, July 2014.

[31] "Case Studies, from various sources, published by Pump Systems Matter," Pumpsystemsmatter.org. [Online]. Available: http://www.pumpsystemsmatter.org.

[32] "Good Practice Guide 249: Energy Savings in Industrial Water Pumping Systems, prepared by ETSU for the Department of Environment," Transport and Regions, US, 1998.

[33] A.Bilgin, H. Hagras, A. Malibari, M. Alhaddad and D. Alghazawi, "Towards a linear general type-2 fuzzy logic based approach for computing with words," Soft Computing,pp.2203-2222, May 2013.

[34] C. Wierzynski, "The Challenges and Opportunities of Explainable AI," 2018. [Online]. Available: https://www.intel.ai/the-challenges-andopportunities-of-explainable-ai/.

[35] V. Callaghan, M. Colley, H. Hagras, J. Chin, F. Doctor and G. Clarke, "Programming iSpaces — A Tale of Two Paradigms," Computer Communications and Networks Intelligent Spaces, pp. 389–421, 2006.

[36] 'Deloitte Predictive-Maintenance Position Paper', Deloitte.com. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/deloitte-analytics/Deloitte_Predictive-Maintenance_PositionPaper.pdf.

[37] 'Operations & Maintenance Best Practices Guide: Release 3.0', Energy.gov. [Online]. Available: https://www.energy.gov/sites/prod/files/2013/10/f3/omguide_complete.pdf.

[38] S. Matzka, "Explainable artificial intelligence for predictive maintenance applications," in *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, 2020.

[39] I. T. Christou, N. Kefalakis, A. Zalonis, and J. Soldatos, "Predictive and Explainable Machine Learning for Industrial Internet of Things Applications," *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, May 2020.

[40] F. Abate, V. K. L. Huang, G. Monte, V. Paciello, and A. Pietrosanto, "A comparison between sensor signal preprocessing techniques," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2479–2487, 2015.

[41] "ThingsBoard – open-source IoT platform," *ThingsBoard*. [Online]. Available: https://thingsboard.io/. [Accessed: 25-Jul-2022].

[42] Y. Bouabdallaoui, Z. Lafhaj, P. Yim, L. Ducoulombier, and B. Bennadji, "Predictive maintenance in building facilities: A machine learning-based approach", *Sensors (Basel)*, vol. 21, no. 4, p. 1044, 2021.

[43] M. Y. L. Chew and K. Yan, "Enhancing interpretability of data-driven fault detection and diagnosis methodology with maintainability rules in smart building management", J. Sens., vol. 2022, pp. 1–48, 2022.

[44] D. Gonzalez-Jimenez, J. del-Olmo, J. Poza, F. Garramiola, and P. Madina, "Data-Driven Fault Diagnosis for Electric Drives: A Review," Sensors, vol. 21, no. 12, p. 4024, Jun. 2021.