Received: 5 January 2023 | Revised: 1 March 2023 | Accepted: 14 April 2023

DOI: 10.1111/exsy.13316

ORIGINAL ARTICLE

Expert Systems WILEY

Explainable fault prediction using learning fuzzy cognitive maps

Taha Mansouri | Sunil Vadera

School of Science, Engineering, and Environment, University of Salford, Manchester, UK

Correspondence
Taha Mansouri, School of Science, Engineering, and Environment, University of Salford, Manchester, UK.
Email: t.mansouri@salford.ac.uk

Funding information
Innovate UK

Abstract

IoT sensors capture different aspects of the environment and generate high throughput data streams. Besides capturing these data streams and reporting the monitoring information, there is significant potential for adopting deep learning to identify valuable insights for predictive preventive maintenance. One specific class of applications involves using Long Short-Term Memory Networks (LSTMs) to predict faults happening in the near future. However, despite their remarkable performance, LSTMs can be very opaque. This paper deals with this issue by applying Learning Fuzzy Cognitive Maps (LFCMs) for developing simplified auxiliary models that can provide greater transparency. An LSTM model for predicting faults of industrial bearings based on readings from vibration sensors is developed to evaluate the idea. An LFCM is then used to imitate the performance of the baseline LSTM model. Through static and dynamic analyses, we demonstrate that LFCM can highlight (i) which members in a sequence of readings contribute to the prediction result and (ii) which values could be controlled to prevent possible faults. Moreover, we compare LFCM with state-of-the-art methods reported in the literature, including decision trees and SHAP values. The experiments show that LFCM offers some advantages over these methods. Moreover, LFCM, by conducting a what-if analysis, could provide more information about the black-box model. To the best of our knowledge, this is the first time LFCMs have been used to simplify a deep learning model to offer greater explainability.

KEYWORDS
deep learning, explanation by simplification, learning fuzzy cognitive maps, predictive preventive maintenance

1 | INTRODUCTION

The Internet of Things has shown unprecedented breakthroughs in many domains, such as smart cities, smart agriculture, and health care, where condition monitoring is a critical application (Nwakanma et al., 2021; Zhang et al., 2018). In these applications, sensors are used to detect faults before happening (Guo et al., 2021; Plante et al., 2015). Unpredicted faults can cause many catastrophic consequences for people, types of machinery, and production lines (Kok et al., 2022; Long et al., 2022). To this end, a different aspect of the environment surrounding equipment can be sensed to estimate its normal or abnormal behaviour. Among those aspects, vibration is the main source of monitoring equipment conditions (Li et al., 2017; Plante et al., 2015; Rahnama et al., 2019).

IoT applications have remarkably applied data-driven models to cope with pervasive sensors and high throughput of generated data streams (Ghosh et al., 2018). These models have been extensively adopted in predictive preventive maintenance applications because building functional analytical models is overwhelming for complex mechanical systems, and IoT sensors facilitate gathering health condition data (Zhao et al., 2020; Zhao, Jia, Bin et al., 2021). Data-driven fault detection methods take advantage of machine learning, particularly deep learning, to build solutions based on the condition data concerning various states (Long et al., 2022). To this end, two phases are considered, (i) to recognize the fault patterns based on different extracted features (ii) to build a predictor using the extracted features, and machine learning/deep learning models (Hasan et al., 2021).

In conventional machine learning, a primary challenge is to select important features to train a model. Most feature selection techniques reported in the previous fault detection research are either the metaheuristics (Oreski & Oreski, 2014) or the filter-based approach (Ambusaidi et al., 2016). Moreover, data compression methods such as PCA (Xie et al., 2018), and manifold learning techniques are adopted for dimension reduction for fault detection (Refahi Oskouei et al., 2012). The majority of previous research also combines wavelet transformation with other techniques such as empirical mode decomposition and self-organizing maps (Hong et al., 2014), neural networks (Narendiranath et al., 2017; Zhou et al., 2019) and Multiclass SVM (Rahnama et al., 2019). Evolutionary strategies also have been deployed to tackle fault prediction (Long et al., 2019; Wang, Kang et al., 2019).

Deep learning models have shown more flexibility in task learning and have taken advantage of the embedded feature transformation to get to a more separable space. Convolutional Neural Networks (CNNs) are used in many fault prediction applications (Chen & Lee, 2020; Cheng et al., 2021; Mehdiyev & Fettke, 2021; Sun et al., 2020). Moreover, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), and Gated Recurrent Units (GRU) (Cho et al., 2014), have become popular time-series processing methods that can effectively encode temporal information (Guo et al., 2021; Okubo et al., 2017; Wang, Yan et al., 2019).

Despite the success of deep learning, it is often criticized for being a black-box model that lacks of transparency (Arrieta et al., 2019; Guidotti et al., 2019; Lin et al., 2020; Wang et al., 2021; Yoon et al., 2019; Mansouri & Vadera, 2022), and this notion does not allow users to digest and trust the output of deep learning models (Kok et al., 2022). In a black box model, internal processes are either unknown or known, but a human being cannot understand them. Explainable Artificial Intelligence techniques are any effort through which further explanation can be provided to shed light on such opaque machine learning models (Schlegel et al., 2019). Whilst some models are interpretable by nature, the explanation process can assist in more in-depth understanding of a black box model (Doshi-Velez & Kim, 2017; Guidotti et al., 2019; Kok et al., 2022).

Several ways of providing explanations include explanations by text, visualizations, local explanations, explanations by example, explanations by simplification, and feature relevance (Arrieta et al., 2019; Kok et al., 2022; Mansouri & Vadera, 2022). Explanations by text learn to generate texts that help interpret the results. Visualizations give visual understanding of a model's output. Local explanations split the solution space and describe simpler solution subspaces associated with the main model. Explanations, by example extract data instances that reflect the results of a given model. Explanations by simplification use a simplified yet interpretable auxiliary model whose output is loyal to the baseline black box model (Arrieta et al., 2019). Finally, feature relevance addresses the relationship between the model output and the most important inputs (Arrieta et al., 2019; Chen et al., 2018; Mansouri & Vadera, 2022; Yoon et al., 2019).

As LSTMs are widely adopted in fault analysis (Guo et al., 2021; Huang et al., 2021; Mansouri & Vadera, 2022; Nwakanma et al., 2021; Zhao et al., 2017; Zheng et al., 2017) we consider them as the baseline model for fault prediction in industrial bearings using vibration sensor readings. LSTMs are also black boxes in which the internal operation of the gates is very hard to understand. Although gaining an understanding of an LSTM's internal gates is quite interesting, it is unlikely to be suitable for users without in-depth knowledge about this model. In this paper, we explore the use of the Learning Fuzzy Cognitive Map (LFCM) (Salmeron et al., 2019) as an auxiliary model that is built to be loyal to the output of the trained LSTM for interpreting its results. Fuzzy Cognitive Maps (FCMs) are flexible and robust models for system state prediction and interpretable knowledge representation (Wang et al., 2021). FCMs are also capable of supporting what-if analysis which can help bring out the causal relationships between the input and output variables. To the best of our knowledge, this work is the first to take advantage of FCMs for explaining deep learning models through model simplification. The paper includes a comparison of the use of LFCM with very recent methods for interpreting deep learning models (Senoner et al., 2022); this model extracts the most important features of a black-box model by calculating its given SHAP values (Lundberg et al., 2017, 2020). SHAP values (SHapley Additive exPlanations) is an explaining technique based on game theory to improve the interpretability of machine learning models. To this end, this technique decomposes the output of the base model into the contribution of each feature named "SHAP values" (see Lundberg et al., 2017, 2020 for more details). In other words, the SHAP value method can be viewed as a cooperative game where the payoff should be assigned to each feature based on their contribution (Senoner et al., 2022).

The remaining part of this paper has been organized as follows. Section 2 reviews related research on model simplification techniques, FCM applications, and previous efforts for equipment fault diagnosis. Section 3 presents the framework for this research, and Section 4 describes the model, the dataset, and the experimental results of prediction and explanation tasks. Finally, Section 5 concludes the paper and provides new directions for research.

2 | PREVIOUS WORKS

Since the approach of this work is to adopt an LFCM to carry out an explanation by simplification of the baseline model, which is an LSTM, this section reviews related research on simplifying black-box models and the use of FCMs. Explanation by simplification means building an interpretable auxiliary model based on a trained black-box model to be explained (Arrieta et al., 2019). The supplemental model is usually much simpler than the baseline and tries to imitate its behaviour by reducing complexity.

Almost all simplification models extract interpreting rules (Arrieta et al., 2019). Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) is a typical model in this category. LIME builds locally linear models around the outputs of a black-box network to interpret it. G-REX (Konig et al., 2008) is another method that learns auxiliary rules (Konig et al., 2008). Bastani et al. (2018) implemented a model extraction process to approximate an interpretable model to the opaque one. Tan et al. (2018) took a different approach by combining two methods: a method for model distillation and comparison to investigate the baseline model's risk; and a statistical test to investigate whether the data is missing key training features.

There are several studies on simplifying deep learning models. Zilke et al. (2016) develop DeepRED, which uses an extended decomposition approach to rule extraction for deep neural networks. Che et al. (2017) present Interpretable Mimic Learning using gradient-boosting trees. Thiagarajan et al. (2016) proposed a hierarchical partitioning of the features that display the iterative elimination of improbable class labels until the association is predicted.

FCMs combine ANNs and Fuzzy Logic to introduce an interpretable representation for complex systems (Salmeron et al., 2019). FCMs can be built through experts' judgements or a data-driven learning process. An FCM is developed using a learning method called Learning FCM or LFCM (Salmeron et al., 2019). There are many applications of this technique in the literature, such as in medicine (Nápoles et al., 2014), customer behaviour analysis (Nasserzadeh et al., 2008), students' performance estimation (Mansouri et al., 2021), computer science (Osei-Bryson, 2004), project management (Kahvandi et al., 2018; Kordestani Ghaleenoei et al., 2021), and some other domains (Poomagal et al., 2021).

There are also some reported works in failure/fault modelling using FCMs. Enriquepelaez (1996) used FCMs to carry out Failure Mode Analysis in one of the first efforts. The model was applied to estimate the effects of component faults on the system operation. Ravasan and Mansouri, (2014, 2016) study expert-created FCMs that evaluate further failure associated with ERP implementation. In another work, Liang et al. (2019) deployed a LFCM for fault prediction in part of a railway signalling system. They used a real-coded genetic algorithm to train a LFCM and reported effective performance.

FCMs have been applied in modelling large-scale problems such as Gene Regulatory Networks (Hecker et al., 2009) containing a few thousand concepts (Salmeron et al., 2019). However, as with other interpretable methods, the performance of the model is affected when there are many features. In FCMs, this side effect would lead to a large adjacency matrix that is hard to interpret. To this end, the number of features should be managed through feature selection techniques.

2.1 | Fault diagnosis methods

This section describes previous studies on equipment fault diagnosis through data-driven approaches in more depth. As mentioned in the introduction, machine learning and deep learning are widely adopted in this field. Ben Ali et al. (2015) used the Empirical Mode for extracting features from vibration readings and a neural network for fault prediction. In another research, Zheng et al. (2019) developed a combination of the semi-supervised fisher discriminant analysis for fault prediction.

Deep learning models have been very successful in fault prediction and predictive preventive maintenance (Guo et al., 2021). Zhang et al. (2017) introduced a multi-objective deep belief networks ensemble algorithm by combining multiple training methods for predicting equipment's remaining useful life. Zhang et al. (2019) developed deep fuzzy echo state networks and deep hybrid state networks for equipment fault prediction. Wang, Li et al. (2019) also applied batch-normalized deep neural networks, which could quickly extract fault features. Zhao, Jia, and Liu (2021) applied a semi-supervised graph convolution deep belief network for fault detection of mechanical equipment. Long et al. (2022) proposed a self-training semi-supervised deep learning algorithm to train a fault detection model with a few annotated samples.

In the case of vibration analysis, LSTM networks can analyse the internal correlation of vibration signals among time series data (An et al., 2020; Guo et al., 2021; Huang et al., 2021; Nwakanma et al., 2021; Zhao et al., 2017; Zheng et al., 2017). Zhao et al. (2017) proposed using a Convolutional Bi-directional LSTM for fault prediction. Guo et al. (2021) offered a combination of several sparse auto-encoders and LSTMs for predicting mechanical faults. For fault prediction, Nwakanma et al. (2021) also applied a vibration sensor known as G-Link 200 and LSTM.

Some researchers have added (CNNs) to LSTM to improve feature selection. Niu et al. (2019) developed a system that incorporates programmable logic controller signals with sensor signals for online remaining useful life prediction of equipment; they deployed a 1D-CNN LSTM network architecture to this end. An et al. (2020) introduced a model including CNN with a stacked bi-directional and unidirectional LSTM network for predicting equipment's remaining life.

In terms of explanation, there are some recent efforts. Sun et al. (2020) developed a CNN for equipment fault detection. The authors added an extra layer, Class Activation Maps, into the model for a visual explanation. Another work for visualization has been conducted by Chen and Lee, (2020) in which a CNN for classification is proposed. They applied Gradient class activation mapping for generating heat maps by calculating the weights of each feature map according to the classification scores.

Decision Trees are widely applied to interpret a black box deep learning model through explanations by simplification approach (Christou et al., 2020; Mehdiyev & Fettke, 2021; Senoner et al., 2022). To this end, Senoner et al. (2022) proposed a gradient boosting with decision trees to improve process quality and used SHAP values to obtain the importance of the features. Mehdiyev and Fettke (2021) proposed a technique for predictive maintenance; they also proposed a model agnostic explanation approach called Surrogate Decision Trees. Christou et al. (2020) used a rule-based model to explain the results from a model used to estimate the remaining useful life of industrial equipment.

Brito et al. (2021) used a number of machine learning techniques along with the SHAP and Local Depth-based Feature Importance for the Isolation Forest. The model is applied to the bearing and mechanical fault datasets. Isolation Forests are similar to Random Forests and build on decision trees. In this technique, there is no label; therefore, it is unsupervised. Samples are processed in a tree based on randomly selected features. Samples with shorter branches in the tree are possibly isolated ones and anomalies (Liu et al., 2008).

Hasan et al. (2021) proposed an explainable fault diagnosis model for bearings, including five steps where they considered data preprocessing, feature selection, and feature importance. They also used an additive Shapley explanation followed by k-NN to diagnose and explain each decision of the k-NN. In another work, Li et al. (2022) proposed an adversarial domain generalization network based on class boundary feature detection to diagnose faults. Wang et al. (2020) applied a multi-headed attention mechanism for optimizing CNNs. In another work, Mansouri and Vadera (2022) proposed an instance-wise feature selection technique to highlight the most contributing features in a deep learning model aiming at fault prediction.

3 | FRAMEWORK

The above summarizes various studies in the field of explainable AI for fault detection. In this paper, we present an alternative approach to simplify and explain LSTM for fault detection by using LFCMs. The following section presents the framework for this LFCM-based approach to simplify and explain the LSTM for fault prediction. First, we build an LSTM deep neural network with a few fully connected layers to predict a fault happening in degraded bearings within the next few hours. To this end, dataset D contains n tuples:

$$D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$$

Where $x_i \in R^{T \times m}$ is the input sequence of length T, including m features and $y_i \in \{0,1\}$ is a binary label in which 0 represents normal, and 1 denotes a faulty condition. In this research, we have considered the classification task as a binary classification in which fault might happen or not, even though both the LSTM and LFCM as the auxiliary model can conduct categorical classification to classify different types of faults. The prediction model $f^{\gamma} : x \rightarrow [0,1]$ parametrized by γ, is the LSTM that undertakes the classification task.

After building the predictor, we aim to explain it by a Learned FCM, which is interpretable. FCM models a system containing several interactive concepts as a weighted d-graph, where the vertices indicate components of the system (C_i) and connecting weights (W_{ij}) show the interactions between those components (Figure 1) (Wang et al., 2021).

The sign of W_{ij} implies the relationship between concepts C_i and C_j, and its value shows the intensity of this relationship. A combination of concepts captures a snapshot of the system at any time as a state vector $A^t = \{a_1^t, \ldots, a_n^t\}$, this state vector shows the values associated with each concept at time t (Mansouri et al., 2021) and can be updated by Equation (1):

$$a_i^t = f\left(a_i^{t-1} + \sum_{j \neq i, j=1}^{n} a_j^{t-1} W_{ji}\right) \quad (1)$$

In Equation (1), a_i^t denotes the value of concept C_i at time t, W_{ji} is the weight between input concepts C_j and C_i, n is the number of concepts, and f is an activation function, for which mainly the unipolar sigmoid (Equation (2)) is used (Salmeron et al., 2019):

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (2)$$

Where x is the input, and α is the function slope estimated as a hyperparameter. Whether FCMs are created by experts' judgements or through a learning method they are interpretable (Wang et al., 2021). This interpretability is achieved by using the extracted weight matrix to represent the modelled system, facilitating static and dynamic analyses.

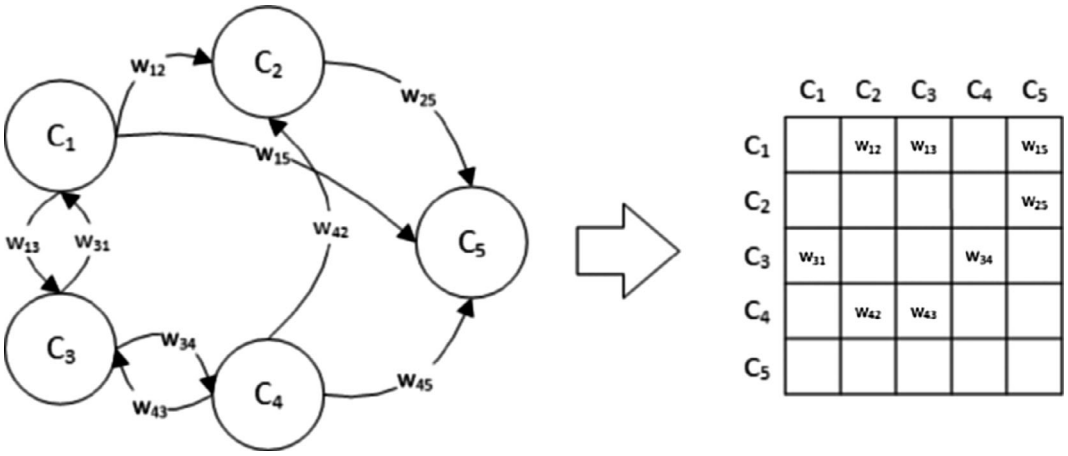

FIGURE 1 Fuzzy cognitive maps can be represented through two different approaches. For small to medium-size maps displaying a visualized graph where nodes are concepts and vertices are weights is informative (left), whereas in larger maps, an adjacency matrix is a more straightforward way to represent a FCM in which each cell shows the relationship between concepts located in the associated row and column (right).

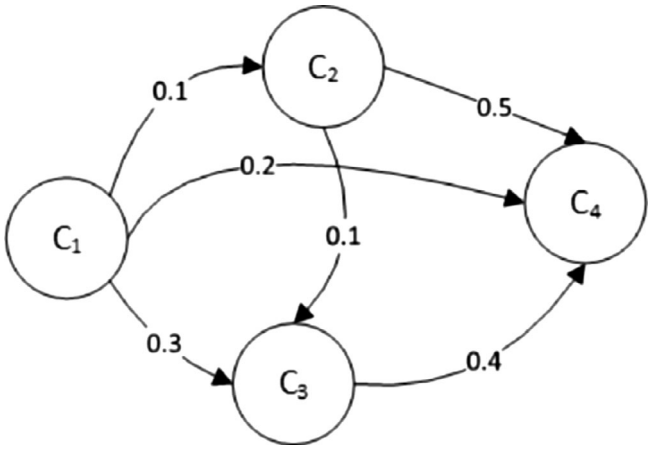

FIGURE 2 Static analysis in FCM, in this figure the direct effect of C_1 on C_4 is 0.2, whilst its indirect effect through C_1 to C_3 to C_4 is 0.3.

Causal effects among the concepts will be drawn in static analysis by finding the maximum value among several paths connecting an input concept to an output one (Ravasan & Mansouri, 2014). In this analysis, first, a partially ordered set P of causal values is taken into consideration. Let ζ be a causal concept space and $e:\zeta\times\zeta\to P$ a causal edge function. Then the simplest abstract operations are achieved by interpreting the indirect-effect operator I as some minimum operator and the total-effect operator T as some maximum operator; these operators can be a simple min and a simple max, respectively. Let there be m-many paths from C_i to $C_j:\left(i,k_1^l,\ldots,k_{nl}^l,j\right)$ for $1\le l\le m$. Let $I_l(C_i,C_j)$ depict the indirect effect of concept C_i on C_j on the lth path; and $T(C_i,C_j)$ be the total effect of C_i on C_j over all m casual paths (Equations (3), (4)).

$$I_l(C_i,C_j)=\min e(C_p,C_{p+l}):(p,p+l)\in\left(i,k_1^l,\ldots,k_{nl}^l,j\right) \tag{3}$$

$$T(C_i,C_j)=\max_{1\le l\le m} I_l(C_i,C_j) \tag{4}$$

Where p and $p+l$ are continuous left-to-right path indices (Kosko, 1986), this analysis highlights the fundamental importance of each concept for the target concept(s).

Figure 2 illustrates the static analysis in the FCM. C_1 is a given input concept, and C_4 is the output concept. There are four possible paths to connect them. The I operator finds the minimum among all weights in each path. For instance, the minimum weight in $C_1\to C_2\to C_4$ path is the weight connecting C_1 to C_2 which its value is 0.1. After calculating all possible paths by I operator, the T operator selects the path with the maximum impact, which in this example is $C_1\to C_3\to C_4$ and its value is 0.3.

The dynamic analysis commences with an initial state vector, such as $A^0 = \{a_1^0, ..., a_n^0\}$ Indicating the corresponding values of all concepts, and keep updating with Equation (1) and Equation (2) (Mansouri et al., 2021). Through this analysis, one can conduct a what-if analysis and check the effect of a change in a concept on the other system members in the future. Of course, before either static or dynamic analysis can be performed, a model and its weights are needed. This research aims to train an LFCM that demonstrates the result of the LSTM, and a Modified Asexual Reproduction Optimization method (FCM-MARO) proposed in Salmeron et al. (2019) is used to train the LFCM model. The objective function of FCM-MARO is Equation (5), and the algorithm aims to minimize this in-sample error through a novel evolutionary strategy.

$$error = \frac{1}{(K-1)N} \sum_{t=1}^{K} \sum_{n=1}^{N} | C_n(t) - \widehat{C}_n(t) | \tag{5}$$

Where $C_n(t)$ denotes the actual value of the concept n at time t, $\widehat{C}_n(t)$ is the estimated value through the LFCM, K is the number of samples, and N is the number of concepts. Algorithm 1 provides the pseudocode of the FCM-MARO algorithm (Salmeron et al., 2019).

This algorithm follows an evolutionary strategy, namely ARO (Farasat et al., 2010; Mansouri et al., 2011) A solution, in ARO is a vector of decision variables $X \in R^n$, where n is the length of the vector. The algorithm starts with a randomly initialized solution namely the parent and reproduces an offspring named bud through a specific reproduction mechanism. In the original ARO bud is replaced with the parent once it is better, but the algorithm is modified by adding another acceptance criterion for a new offspring, even if it is not better than the parent by considering the effect of falling in the local minima using Equation (6). Where *local* means the number of times being trapped in the local optimum and t is the number of iterations. FCM-MARO, which can train a data-driven FCM or LFCM, is fully described in Salmeron et al. (2019).

$$\delta_t = \frac{\log(local)}{\sqrt{t}} \tag{6}$$

The reproduction mechanism combines the mutation and the crossover operators in other evolutionary algorithms. A subset of the parent vector will be selected and mutated randomly to constitute an interim vector named larva. Then the crossover operator applies over the larva and the parent to build the ultimate offspring (bud). The length of this larva specifies the possibility of selecting each part of the bud from this vector. To this end, Equation (7) estimates the corresponding probability (see Farasat et al., 2010; Mansouri et al., 2011 for details).

$$P = \frac{1}{\log(g)} \tag{7}$$

Where g is the length of the mutated part of the parent vector.

In this paper, the initial state is an input sequence x^{t-1} with the window size T along with a slack variable $\widehat{y}_0 = 0$ to show the probability of being faulty at the beginning. The output is the following sequence, along with the result of the LSTM $\widehat{y}^t = f^{\prime}(x^{t-1})$. Therefore, the LFCM is as $LFCM : R^{T+1} \rightarrow R^{T+1}$ and learns to accept the input and generate the fault probability output.

Algorithm 1 **Pseudocode for FCM-MARO**

while true **do.**
  generate a random offspring from the parent.
  calculate the error of the offspring by Equation (5).
  measure the extra acceptance interval by Equation (6).
  if the error of the offspring is less than the error of the parent **then.**
    swap the parent with the offspring.
    reset the number of times falling in a local minima.
  else if the error of the offspring falls within the extra acceptance interval **then.**
    swap the parent with the offspring.
  else.
    discard the offspring.
  end if.
end while.

4 | EXPERIMENTATION AND DISCUSSION

After introducing the dataset and its related configurations in this section, an LSTM fault prediction model is developed, and all its hyper-parameters are set using a grid search. Then, we go through the explanation by simplification technique to interpret the trained LSTM. To this end, an LFCM model is learned and then used to carry out static and dynamic analysis to extract valuable insights. Moreover, a new model based on decision trees and SHAP values inspired by Senoner et al. (2022) is applied to validate the results of the proposed model to interpret the trained LSTM.

4.1 | Bearing dataset

We used the real-world dataset collected in Mansouri and Vadera (2022) which contains 4 months of vibration readings from four bearings of the same size, class, and category. One of the bearings is disintegrated and produced several faults, whilst others have better conditions. This dataset contains sequential data where each row consists of 20 vibration readings and one feature class in which 0 means normal situation and 1 means fault. All bearings operated in real situations and faults are captured through installed sensors. There are 29,646 sequences. Figure 3 shows the distribution of normal and fault samples in this dataset. We used 80% of this dataset for training and the rest for testing the models. Although the data is clearly imbalanced, the accuracy of the models was good and there was therefore no need to use sampling methods to balance the data.

4.2 | Baseline model

An LSTM network has been selected as the baseline model because it has shown good performance in time series analysis. As a deep learning model containing extra gates, LSTM is a deep learning model in which the relationship between an input and its associated output is quite complex and opaque. In order to select the topology of LSTM, different numbers of neurons for the LSTM layer, different numbers of fully connected layers and their given number of neurons, optimizers including *Adam*, *SGD*, and *RMSProp* as well as different learning rates are tested through the grid search (Liashchynskyi & Liashchynskyi, 2019). This led to a baseline model with a 60 node LSTM layer followed by a 32 node fully connected layer with a relu activation function and an output layer with one node with a sigmoidal activation function. The selected optimizer is *Adam*, and 0.01 was the best learning rate. Figure 4 shows the result of training LSTM on the bearing dataset.

As Figure 5 shows, the confusion matrix of the trained LSTM on the test dataset is promising. The number of false positives and false negatives is low even though the dataset is highly imbalanced. Therefore, LSTM has shown its performance to capture this dataset's complexity without being overfitted or underfitted.[1]

Although LSTM is a powerful model for fault prediction, as a deep learning model, it is a black box and uninterpretable. Therefore, explaining it as an acceptable obscure deep learning model is worthwhile. To provide some interpretability, an LFCM is developed and used to carry out some analysis. A combination of decision trees and SHAP values is then used to provide a contrasting approach.

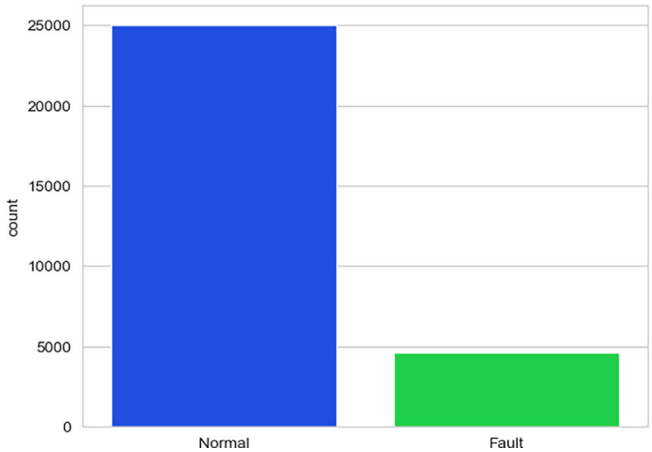

FIGURE 3 The distribution of normal and fault samples in the bearing dataset.

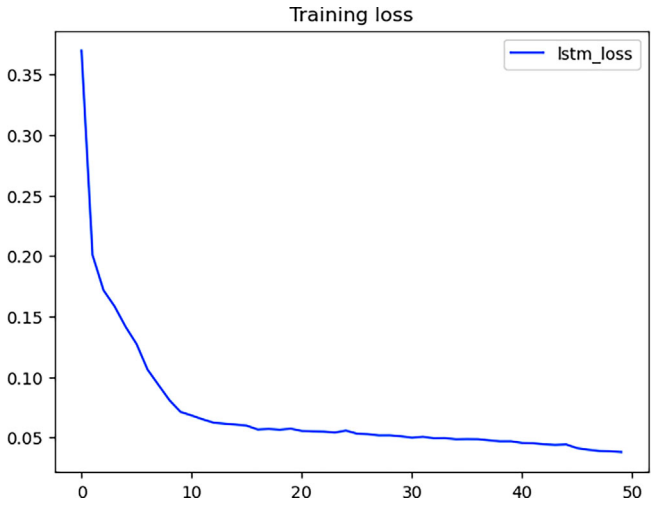

FIGURE 4 Plotting the loss function resulted from the training of the LSTM for fault detection on bearing dataset.

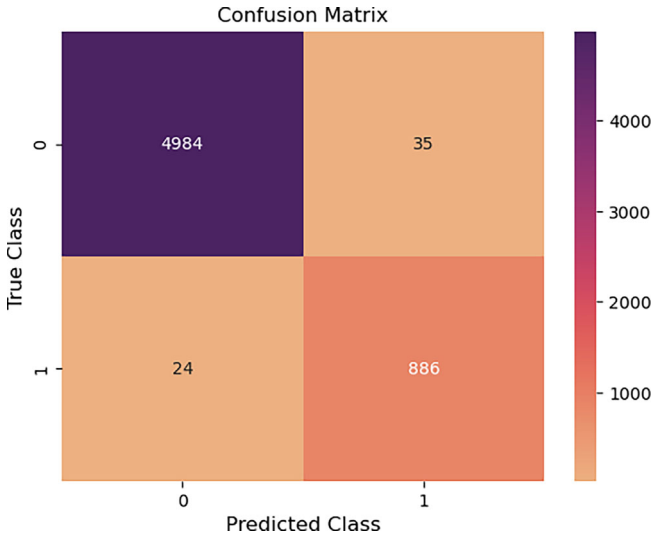

FIGURE 5 Confusion matrix resulted from running the trained LSTM on the test dataset. The error portion of this matrix is low.

4.3 | Explanation by simplification

To create the LFCM model, we used the bearing dataset by which we had trained the LSTM. LFCM aims to receive a sequence containing 20 readings and one null variable and create the next sequence along with the prediction result. Therefore, the input and output have 21 concepts each (sequence members and the slack variable denoting the probability of a fault) and the generated adjacency matrix is 21 by 21. As the current problem was binary classification, we used one output concept. In multi-class problems, one can add outputs to the model as many as the number of classes. So, the final adjacency matrix is $(T + m)$ by $(T + m)$, where T is the number of inputs and m is the number of classes. Figure 6 displays the LFCM and its related input and output structure. During the training, the LFCM learns to accept an input vector and predict the output vector with the same size containing the fault prediction close to the LSTM's output.

Since Algorithm 1 is an evolutionary algorithm with no hyperparameter, we just set the number of iterations as 100, and ran it to build a LFCM using the bearing dataset. As Figure 7 shows, the LFCM converges to a low error rate (as defined by Equation (5)).

The trained LFCM accepts input and estimates the fault probability during the simulation and test. The resulting LFCM has been tested on the same test dataset by which the LSTM was tested. Figure 8 displays the confusion matrix of LFCM on the test subset of the bearing dataset. Based on this result, the performance of LFCM is very close to the performance of the trained LSTM. However, because LFCM is much simpler than LSTM, its capacity to capture the nonlinearity among the data is less than LSTM (Wang et al., 2021).

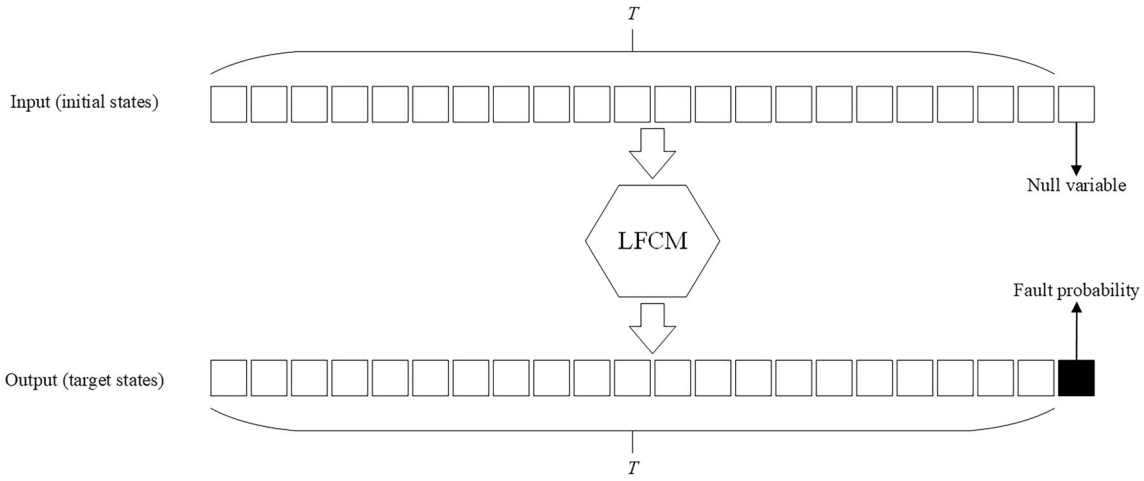

FIGURE 6 The structure of the LFCM and its input and output.

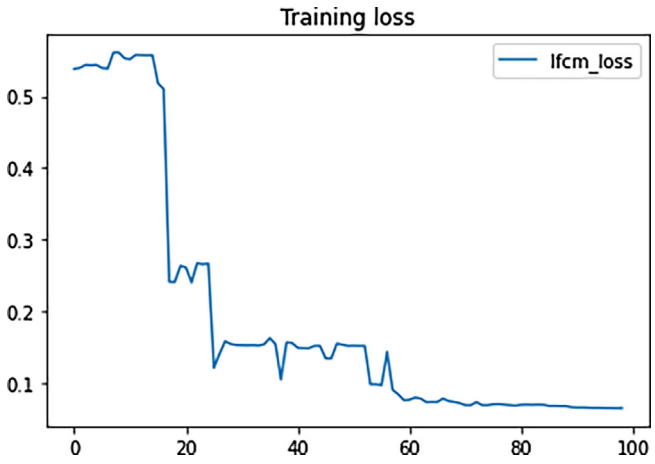

FIGURE 7 The structure of the LFCM and its input and output.

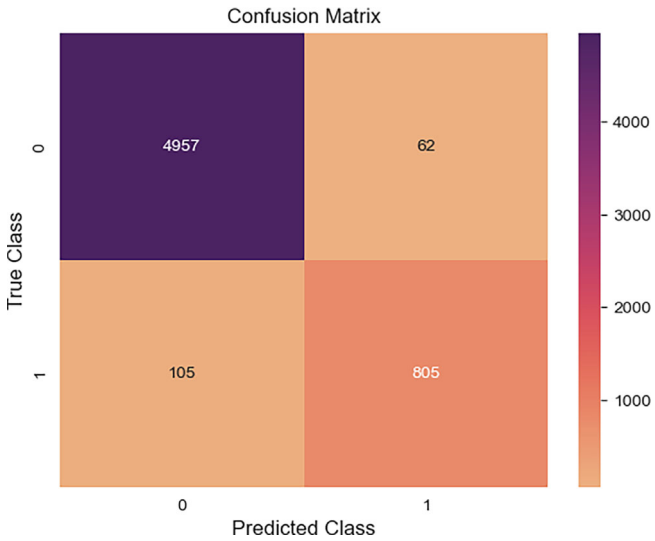

FIGURE 8 Confusion matrix resulted from running the trained LFCM on the test dataset. The error portion of this matrix is also quit low and close to LSTM performance.

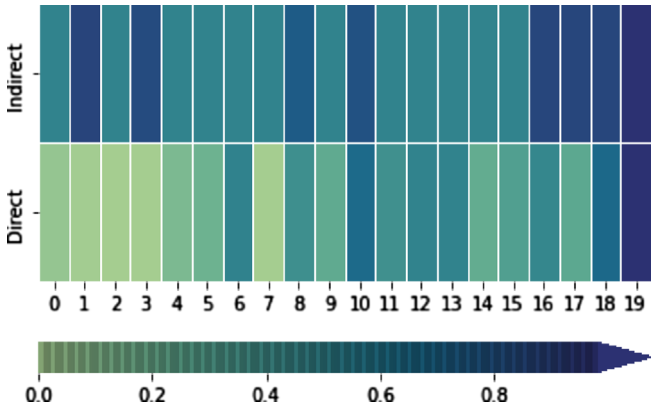

FIGURE 9 FCM static analysis, comparison between direct effects of each feature (member) on the class variable (the probability of being faulty) and their indirect effects by the static analysis.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Fault
0.03	0.03	0.03	0.04	0.03	0.04	0.04	0.03	0.03	0.03	0.04	0.03	0.03	0.04	0.03	0.04	0.04	0.03	0.03	0.03	0

FIGURE 10 A normal reading, which is predicted as normal by the LFCM.

Once a trained LFCM can predict the bearing condition using a sequence of vibration readings, it is time to conduct static and dynamic analysis to extract more insight into the baseline model.

4.3.1 | Static analysis

According to the weight matrix obtained from the trained LFCM, each input variable directly affects the target concept, which denotes the probability of being faulty in the future. On the other hand, since FCMs are dynamic systems, these variables have interconnected effects, so their total effects should be considered. Nevertheless, Equations (3) and (4) are applied to estimate the total effects of each variable on the target one. Figure 9 shows the direct and indirect effects achieved by the static analysis.

As Figure 9 implies, almost all concepts have more indirect effects than direct ones. Considering the direct effect, the starting members do not affect the target variables, but after going through the static analysis, their indirect effect becomes considerably higher. Based on the static analysis, it can be concluded that members indexed by 1, 3, 8, 10, 16, 17, 18, and 19 have more effects on the target variable. It means that more fluctuation in the target is expected by changing the related values. It should be noted that FCM static analysis draws the causal relation between input and output concepts throughout the dataset, and it is not an instance-wise feature selection technique. Nevertheless, it is expected that changing the values of the more essential concepts leads to more output change than the other concepts. Therefore, this analysis draws a dynamic system's relationship between the features and the target variable. In this research, features are the members of a sequence. However, this result should be validated through the what-if analysis carried out in the dynamic analysis part.

4.3.2 | Dynamic analysis

So far, considering the FCM's static analysis, the most critical variables for the prediction task have been identified. FCM can also carry out what-if analysis through dynamic analysis to gain more insight into behaviours of the baseline model in different situations. To this end, one normal and one fault sample are randomly selected; regarding the feature order, we grouped them into four different categories, each containing five features (e.g. feature number 1 to feature number 5 are grouped together). Therefore, four scenarios are conducted for each condition. The value of each selected part is changed and sent to the LFCM to estimate the effect of this change on the class variable or fault probability.

Figure 10 shows a normal sample correctly classified by the trained LFCM. Its variables are grouped into four categories including five members. It means for the first scenario, the values of the first five members have been changed to the threshold; for the second scenario, the values of the second five members have been changed, and so on to the end.

FIGURE 11 Four scenarios conducted on a normal sequence. In the first scenario, the first five members are manipulated randomly, and in the successive scenarios second, third, and fourth five members are manipulated.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Fault
0.46	0.07	0.06	0.06	0.07	0.07	0.06	0.06	0.06	0.44	0.46	0.06	0.47	0.47	0.07	0.55	0.57	0.05	0.06	0.05	1

FIGURE 12 A fault reading, which is predicted as fault by the LFCM.

Figure 11 shows the results of the mentioned scenarios on the chosen sequence. In the first scenario, the first five features are manipulated, and we have modified the others similarly. The fourth scenario highlights that just changing the fourth quarter's values has led to faulty conditions, while the others could not change the classification result.

The next set of experiments is related to the faulty condition. Figure 12 shows a random faulty sample. Here the question is, by controlling which members, one could prevent the fault from happening. As before, there are four experiments where each group member's values have decreased to 0.1. Through LFCM dynamic analysis, the effect of this manipulation is shown in Figure 13.

As noted in Figure 13, one can prevent the fault by controlling the values of the fourth quarter, while controlling the other categories has no direct influence on the condition. By knowing that, we could conduct a what-if analysis. Therefore, the dynamic analysis of a FCM can provide us with more valuable information about the behaviour of a black-box model.

4.3.3 | Validation

In order to check the result of LFCM, we took the idea proposed in Senoner et al. (2022) for fault detection by explainable models. They calculated the feature importance of all parameters of their base model with the tree implementation of the SHAP value method (Lundberg et al., 2020). However, the base model in their work is gradient boosting with decision trees (Ke et al., 2017), in our work it is LSTM. Therefore, to use the tree implementation of SHAP values, a decision tree is built on the same training set by which LSTM and LFCM had been trained. In order to tune the hyperparameter of this decision tree, the grid search on the most important parameters of the decision tree is conducted. Table 1 summarizes the results of the grid search. These values are selected among many combinations to buil a decision three such as different criterion including gini, entropy, and none, and also other hyperparameters.

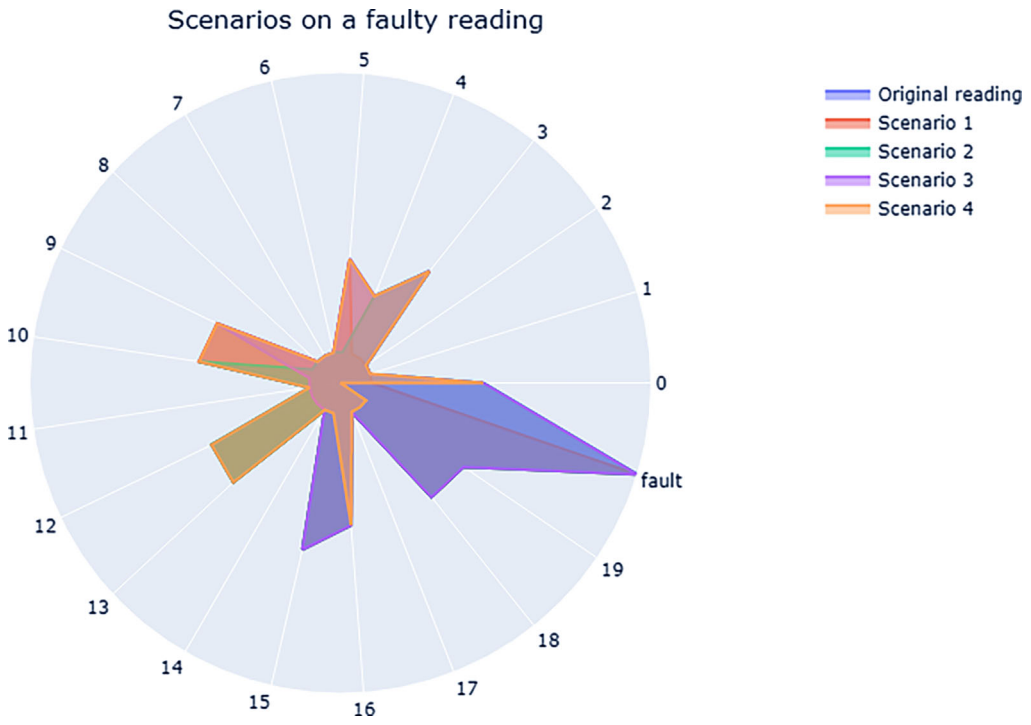

FIGURE 13 Four scenarios conducted on a faulty sequence. In the first scenario, the first five members are manipulated randomly, and in the successive scenarios second, third, and fourth five members are manipulated.

TABLE 1 The decision tree's hyperparameter set by the grid search

Parameter	Value
Criterion	Gini
Maximum number of features	Auto
Maximum number of leaf nodes	14
Splitter	Best

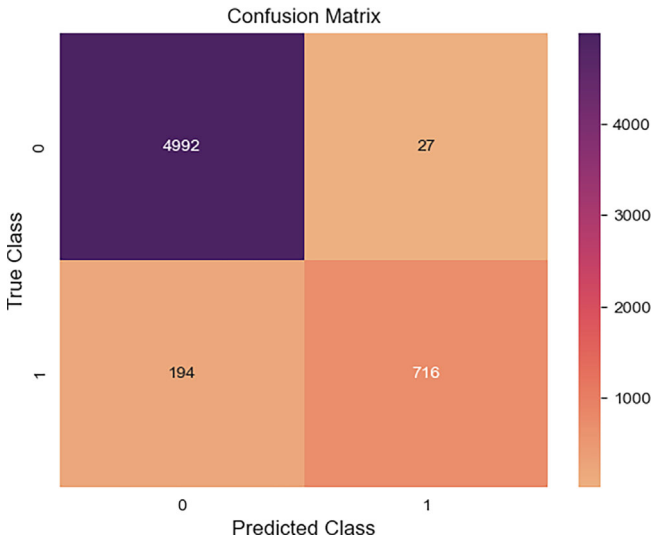

FIGURE 14 Confusion matrix resulted from running the trained decision tree on the test dataset. The error portion of this matrix is also low and close to LSTM performance; however, it is not as good as LFCM's performance.

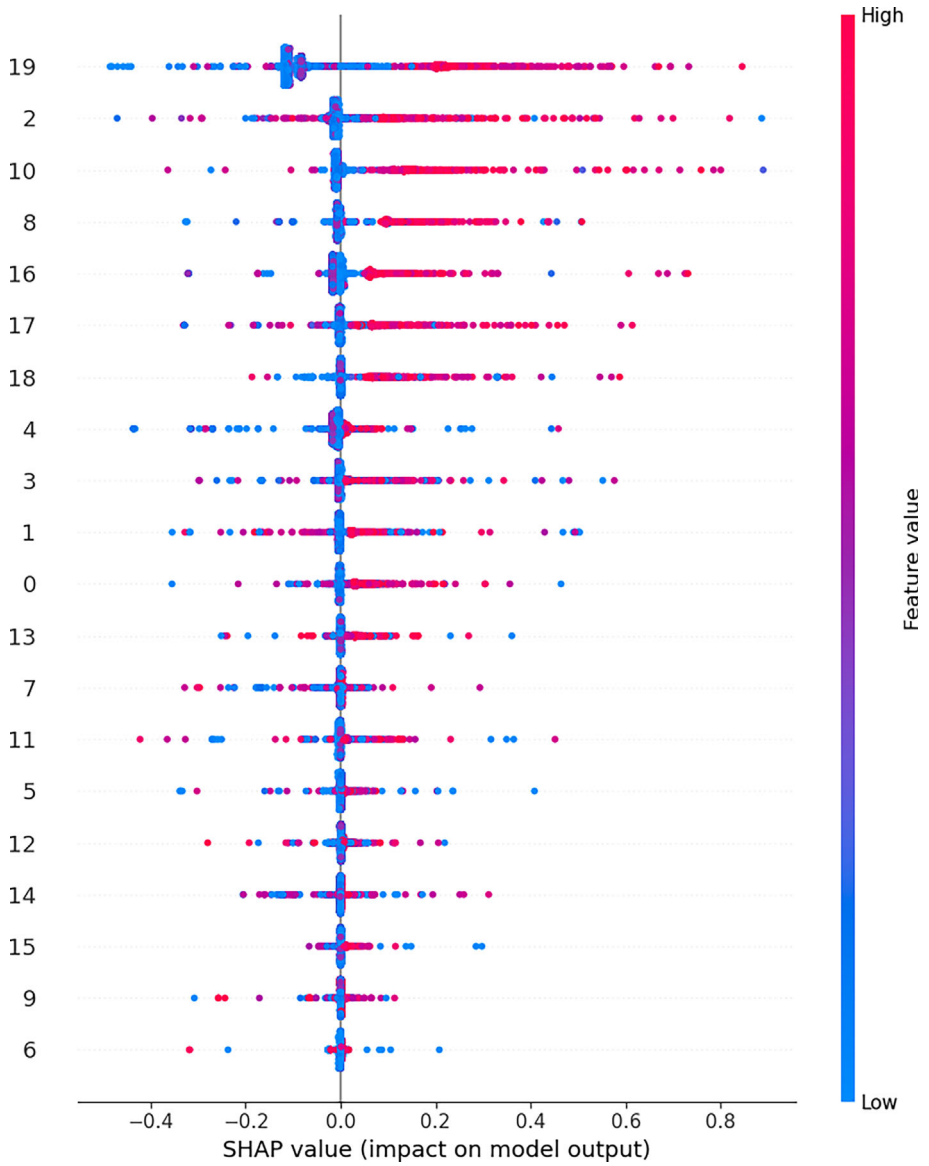

FIGURE 15 The average of SHAP values on the bearing dataset and the trained decision tree. This result shows that the most important feature was the last one; nevertheless, features belonging to the second part of the sequences were more contributing.

After building the decision tree with the designated parameters, it is tested on the test subset of the bearing dataset. Figure 14 shows the confusion matrix resulting from running the trained decision tree on the test dataset. Although, it is still close to LSTM outputs, the LFCM outperforms the decision tree.

In order to select the most contributing features, the average of SHAP values of the trained decision tree is calculated, and the result presented in Figure 15.

Based on Figure 15, the most important features are indexed by 19, 2, 10, 8, 16, 17, 18, 4, and 3; among them, 3, 8, 10, 16, 17, 18, and 19 are common with the LFCM which can validate its results. The average of SHAP values is close to LFCM, yet LFCM is more accurate in comparison to the decision tree and not only highlights the most important features but also carries out what-if analysis to estimate the change in each feature on the whole system.

5 | CONCLUSION

This paper tackles the problem of producing a transparent model for predicting faults from vibration sensor readings. The article first develops a LSTM model and then illustrates how a LFCM can carry out a static and dynamic analysis. In order to validate the performance of the proposed

method as a simplified model, a recent method including decision trees and SHAP values is used. A dataset consisting of sequences of vibration readings and its label is used, and after pre-processing, an LSTM neural network was trained on the training subset and examined on the test set. Different parameters and topologies were tried to tune the given hyperparameters using a grid search. The trained LSTM, an opaque deep learning model, is selected as the baseline model. Then a LFCM is trained to illustrate some parts of the baseline model. To this end, the LFCM is trained to generate outputs relative to the baseline's output using the same data and the baseline output. Experiments show the performance of the LFCM is like the baseline model.

Since an LFCM can carry out static and dynamic analysis, it can interpret some parts of the baseline model. The static analysis can highlight the effects of the input concepts on the probability of being faulty. In the example used in this paper, the concepts belonging to the fourth quarter had more effect on the prediction results. This analysis also showed that even those features that had no, or low direct impact on the target, affected the results through indirect paths. Moreover, the dynamic analysis showed that changing values of the fourth-quarter variables could lead to a fault being predicted for normal readings. A recent method proposed by Senoner et al. (2022) shows that the use of decision trees coupled with the use of SHAP values can also aid in making LSTM models more transparent. LFCM's performance was better than the decision tree, the result of SHAP values was close to LFCM, and finally, since LFCM can carry out dynamic and static analysis, it can conduct what-if analysis besides the feature importance.

In conclusion, this paper demonstrates that LFCMs can capture most of the LSTM's performance, provide added useful capabilities that aid interpretability, and outperform cutting-edge methods from different aspects. For future research, one can improve the performance of the LFCM by adding more nonlinearity without compromising interpretability (Wang et al., 2021), and one can explore if there are benefits in using them for other deep learning models and tasks. Although the use case investigated in this paper was a binary classification, applying the proposed model to multi-class problems is also fascinating. Moreover, combining the proposed method with local feature selection methods when there are many features has the potential to shed further insights about how to make black box models more transparent.

ACKNOWLEDGEMENTS

The work presented in this paper was carried out as part of a Knowledge Transfer Partnership project between the University of Salford and Invisible Systems Ltd (ISL). We are grateful to ISL for providing the data and assisting with the problem definition.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in stream at https://github.com/tahamsi/stream.

ORCID

Taha Mansouri https://orcid.org/0000-0003-1539-5546
Sunil Vadera https://orcid.org/0000-0001-6041-2646

ENDNOTE

[1] In this case, we did not need to adopt a validation set to work out the optimal number of epochs.

REFERENCES

Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Transactions on Computers*, *65*(10), 2986–2998. https://doi.org/10.1109/TC.2016.2519914

An, Q., Tao, Z., Xu, X., el Mansori, M., & Chen, M. (2020). A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. *Measurement*, *154*, 107461. https://doi.org/10.1016/j.measurement.2019.107461

Arrieta, A. B., Díaz-Rodríguez, N., del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2019). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. http://arxiv.org/abs/1910.10045

Bastani, O., Kim, C., & Bastani, H. (2018). Interpretability via model extraction. http://arxiv.org/abs/1706.09773

Ben Ali, J., Fnaiech, N., Saidi, L., Chebel-Morello, B., & Fnaiech, F. (2015). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, *89*, 16–27. https://doi.org/10.1016/j.apacoust.2014.08.016

Brito, L. C., Susto, G. A., Brito, J. N., & Duarte, M. A. V. (2021). An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery (arXiv:2102.11848). http://arxiv.org/abs/2102.11848

Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2017). Interpretable deep models for ICU outcome prediction. *American Medical Informatics Association Annual Symposium Proceedings*, 371–380.

Chen, H.-Y., & Lee, C.-H. (2020). Vibration signals analysis by explainable artificial intelligence (XAI) approach: Application on bearing faults diagnosis. *IEEE Access*, *8*, 134246–134256. https://doi.org/10.1109/ACCESS.2020.3006491

Chen, J., Song, L., Wainwright, M. J., & Jordan, M. I. (2018). Learning to explain: An information-theoretic perspective on model interpretation. http://arxiv.org/abs/1802.07814

Cheng, Y., Lin, M., Wu, J., Zhu, H., & Shao, X. (2021). Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network. *Knowledge-Based Systems*, *216*, 106796. https://doi.org/10.1016/j.knosys.2021.106796

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1724–1734. https://doi.org/10.3115/v1/D14-1179

Christou, I. T., Kefalakis, N., Zalonis, A., & Soldatos, J. (2020). Predictive and explainable machine learning for industrial internet of things applications. In *2020 16th international conference on distributed computing in sensor systems (DCOSS)* (pp. 213–218). IEEE. https://doi.org/10.1109/DCOSS49796.2020.00043

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. http://arxiv.org/abs/1702.08608

Enriquepelaez, C. (1996). Using fuzzy cognitive maps as a system model for failure modes and effects analysis. *Information Sciences*, *88*(1–4), 177–199. https://doi.org/10.1016/0020-0255(95)00161-1

Farasat, A., Menhaj, M. B., Mansouri, T., & Moghadam, M. R. S. (2010). ARO: A new model-free optimization algorithm inspired from asexual reproduction. *Applied Soft Computing*, *10*(4), 1284–1292. https://doi.org/10.1016/j.asoc.2010.05.011

Ghosh, A., Chakraborty, D., & Law, A. (2018). Artificial intelligence in internet of things. *CAAI Transactions on Intelligence Technology*, *3*(4), 208–218. https://doi.org/10.1049/trit.2018.1008

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM Computing Surveys*, *51*(5), 1–42. https://doi.org/10.1145/3236009

Guo, J., Lao, Z., Hou, M., Li, C., & Zhang, S. (2021). Mechanical fault time series prediction by using EFMSAE-LSTM neural network. *Measurement*, *173*, 108566. https://doi.org/10.1016/j.measurement.2020.108566

Hasan, M. J., Sohaib, M., & Kim, J.-M. (2021). An explainable AI-based fault diagnosis model for bearings. *Sensors*, *21*(12), 4070. https://doi.org/10.3390/s21124070

Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., & Guthke, R. (2009). Gene regulatory network inference: Data integration in dynamic models—A review. *Biosystems*, *96*, 86–103.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural. *Computing*, *9*(8), 1735–1780.

Hong, S., Zhou, Z., Zio, E., & Hong, K. (2014). Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method. *Digital Signal Processing*, *27*, 159–166. https://doi.org/10.1016/j.dsp.2013.12.010

Huang, D., Fu, Y., Qin, N., & Gao, S. (2021). Fault diagnosis of high-speed train bogie based on LSTM neural network. *Science China Information Sciences*, *64*(1), 119203. https://doi.org/10.1007/s11432-018-9543-8

Kahvandi, Z., Saghatforoush, E., Ravasan, A. Z., & Mansouri, T. (2018). An FCM-based dynamic modelling of integrated project delivery implementation challenges in construction projects, *26*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*, 3149–3157.

Kok, I., Okay, F. Y., Muyanli, O., & Ozdemir, S. (2022). Explainable artificial intelligence (XAI) for internet of things: A survey (arXiv:2206.04800). http://arxiv.org/abs/2206.04800

Konig, R., Johansson, U., & Niklasson, L. (2008). G-REX: A versatile framework for evolutionary data mining. *2008 IEEE International Conference on Data Mining Workshops*, Pisa, Italy, pp. 971–974. https://doi.org/10.1109/ICDMW.2008.117

Kordestani Ghaleenoei, N., Saghatforoush, E., Mansouri, T., & Ravasan, A. Z. (2021). An FCM-based dynamic modeling of operability and maintainability barriers in road projects. *International Journal of Pavement Research and Technology.*, *15*, 367–383. https://doi.org/10.1007/s42947-021-00027-z

Kosko, B. (1986). Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, *24*, 65–75.

Li, J., Shen, C., Kong, L., Wang, D., Xia, M., & Zhu, Z. (2022). A new adversarial domain generalization network based on class boundary feature detection for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–9. https://doi.org/10.1109/TIM.2022.3164163

Li, R., Li, C., Peng, X., & Wei, W. (2017). Electromagnetic vibration simulation of a 250-MW large hydropower generator with rotor eccentricity and rotor deformation. *Energies*, *10*(12), 2155. https://doi.org/10.3390/en10122155

Liang, Y., Dai, S., & Zheng, Z. (2019). Fault diagnosis of railway turnout based on fuzzy cognitive map. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, pp. 589–594. https://doi.org/10.1109/ITSC.2019.8917538

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for NAS (arXiv:1912.06059). http://arxiv.org/abs/1912.06059

Lin, Y.-S., Lee, W.-C., & Celik, Z. B. (2020). What do you see? Evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors. http://arxiv.org/abs/2009.10639

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *Eighth IEEE International Conference on Data Mining*, *2008*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Long, J., Chen, Y., Yang, Z., Huang, Y., & Li, C. (2022). A novel self-training semi-supervised deep learning approach for machinery fault diagnosis. *International Journal of Production Research*, *1–14*, 1–14. https://doi.org/10.1080/00207543.2022.2032860

Long, J., Sun, Z., Pardalos, P. M., Hong, Y., Zhang, S., & Li, C. (2019). A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. *Information Sciences*, *478*, 40–61. https://doi.org/10.1016/j.ins.2018.11.006

Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions (arXiv:1705.07874). http://arxiv.org/abs/1705.07874

Lundberg, S., Erion, G., Chen, H., DeGrave, A., Prutkin, J., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1), 56–67.

Mansouri, T., Farasat, A., Menhaj, M. B., & Reza Sadeghi Moghadam, M. (2011). ARO: A new model-free optimization algorithm for real-time applications inspired by the asexual reproduction. *Expert Systems with Applications*, *38*(5), 4866–4874. https://doi.org/10.1016/j.eswa.2010.09.084

Mansouri, T., & Vadera, S. (2022). A deep explainable model for fault prediction using IoT sensors. *IEEE Access*, *10*, 66933–66942. https://doi.org/10.1109/ACCESS.2022.3184693

Mansouri, T., Ravasan, A. Z., & Ashrafi, A. (2021). A learning fuzzy cognitive map (LFCM) approach to predict student performance. *Journal of Information Technology Education: Research*, *20*, 221–243. https://doi.org/10.28945/4760

Mehdiyev, N., & Fettke, P. (2021). Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring. In W. Pedrycz & S.-M. Chen (Eds.), *Interpretable artificial intelligence: A perspective of granular computing* (pp. 1–28). Springer International Publishing. https://doi.org/10.1007/978-3-030-64949-4

Nápoles, G., Grau, I., Bello, R., & Grau, R. (2014). Two-steps learning of fuzzy cognitive maps for prediction and knowledge discovery on the HIV-1 drug resistance. *Expert Systems with Applications*, *41*(3), 821–830. https://doi.org/10.1016/j.eswa.2013.08.012

Narendiranath, B. T., Himamshu, H. S., Prabin, K. N., Rama, P. D., & Nishant, C. (2017). Journal bearing fault detection based on Daubechies wavelet. *Archives of Acoustics*, *42*(3), 401–414. https://doi.org/10.1515/aoa-2017-0042

Nasserzadeh, S. M. R., Jafarzadeh, M. H., Mansouri, T., & Sohrabi, B. (2008). Customer satisfaction fuzzy cognitive map in banking industry. *Communications of the IBIMA*, *2*, 12.

Niu, J., Liu, C., Zhang, L., & Liao, Y. (2019). Remaining useful life prediction of machining tools by 1D-CNN LSTM network. *IEEE Symposium Series on Computational Intelligence (SSCI)*, *2019*, 1056–1063. https://doi.org/10.1109/SSCI44817.2019.9002993

Nwakanma, C. I., Islam, F. B., Maharani, M. P., Kim, D.-S., & Lee, J.-M. (2021). IoT-based vibration sensor data collection and emergency detection classification using Long short term memory (LSTM). *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, *2021*, 273–278. https://doi.org/10.1109/ICAIIC51459.2021.9415228

Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. (2017). A neural network approach for students' performance prediction. *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, *598–599*. https://doi.org/10.1145/3027385.3029479

Oreski, S., & Oreski, G. (2014). Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert Systems with Applications*, *41*(4), 2052–2064. https://doi.org/10.1016/j.eswa.2013.09.004

Osei-Bryson, K.-M. (2004). Generating consistent subjective estimates of the magnitudes of causal relationships in fuzzy cognitive maps. *Computers and Operations Research*, *31*(8), 1165–1175. https://doi.org/10.1016/S0305-0548(03)00070-4

Plante, T., Nejadpak, A., & Yang, C. X. (2015). Vibration analysis: Fault detection and failure prediction. *IEEE Autotestcon*, *2015*, 5.

Poomagal, S., Sujatha, R., Kumar, P. S., & Vo, D.-V. N. (2021). A fuzzy cognitive map approach to predict the hazardous effects of malathion to environment (air, water and soil). *Chemosphere*, *263*, 127926. https://doi.org/10.1016/j.chemosphere.2020.127926

Rahnama, M., Vahedi, A., Alikhani, A. M., & Montazeri, A. (2019). Machine-learning approach for fault detection in brushless synchronous generator using vibration signals. *IET Science, Measurement & Technology*, *13*(6), 852–861. https://doi.org/10.1049/iet-smt.2018.5523

Ravasan, A. Z., & Mansouri, T. (2014). A FCM-based dynamic modeling of ERP implementation critical failure factors. *International journal of Enterprise information systems*, *10*(1), 32–52. https://doi.org/10.4018/ijeis.2014010103

Ravasan, A. Z., & Mansouri, T. (2016). A dynamic ERP critical failure factors modelling with FCM throughout project lifecycle phases. *Production Planning and Control*, *27*(2), 65–82. https://doi.org/10.1080/09537287.2015.1064551

Refahi Oskouei, A., Heidary, H., Ahmadi, M., & Farajpur, M. (2012). Unsupervised acoustic emission data clustering for the analysis of damage mechanisms in glass/polyester composites. *Materials and Design*, *37*, 416–422. https://doi.org/10.1016/j.matdes.2012.01.018

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). 'Why should I trust you?': Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM. https://doi.org/10.1145/2939672.2939778

Salmeron, J. L., Mansouri, T., Moghadam, M. R. S., & Mardani, A. (2019). Learning fuzzy cognitive maps with modified asexual reproduction optimisation algorithm. *Knowledge-Based Systems*, *163*, 723–735. https://doi.org/10.1016/j.knosys.2018.09.034

Schlegel, U., Arnout, H., El-Assady, M., Oelke, D., & Keim, D. A. (2019). Towards a rigorous evaluation of XAI methods on time series. http://arxiv.org/abs/1909.07082

Senoner, J., Netland, T., & Feuerriegel, S. (2022). Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*, *68*(8), 5704–5723. https://doi.org/10.1287/mnsc.2021.4190

Sun, K. H., Huh, H., Tama, B. A., Lee, S. Y., Jung, J. H., & Lee, S. (2020). Vision-based fault diagnostics using explainable deep learning with class activation maps. *IEEE Access*, *8*, 129169–129179. https://doi.org/10.1109/ACCESS.2020.3009852

Tan, S., Caruana, R., Hooker, G., & Lou, Y. (2018). Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. *Proceedings of the 2018 AAAI/ACM conference on AI, Ethics, and Society*, 303–310. https://doi.org/10.1145/3278721.3278725

Thiagarajan, J. J., Kailkhura, B., Sattigeri, P., & Ramamurthy, K. N. (2016). TreeView: Peeking into deep neural networks via feature-space partitioning. http://arxiv.org/abs/1611.07429

Wang, H., Xu, J., Yan, R., Sun, C., & Chen, X. (2020). Intelligent bearing fault diagnosis using multi-head attention-based CNN. *Procedia Manufacturing*, *49*, 112–118. https://doi.org/10.1016/j.promfg.2020.07.005

Wang, J., Kang, J., & Hou, G. (2019). Real-time fault repair scheme based on improved genetic algorithm. *IEEE Access*, *7*, 35805–35815. https://doi.org/10.1109/ACCESS.2019.2905042

Wang, J., Li, S., An, Z., Jiang, X., Qian, W., & Ji, S. (2019). Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines. *Neurocomputing*, *329*, 53–65. https://doi.org/10.1016/j.neucom.2018.10.049

Wang, J., Peng, Z., Wang, X., Li, C., & Wu, J. (2021). Deep fuzzy cognitive maps for interpretable multivariate time series prediction. *IEEE Transactions on Fuzzy Systems*, *29*(9), 2647–2660. https://doi.org/10.1109/TFUZZ.2020.3005293

Wang, J., Yan, J., Li, C., Gao, R. X., & Zhao, R. (2019). Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Computers in Industry*, *111*, 1–14. https://doi.org/10.1016/j.compind.2019.06.001

Xie, H., Li, J., & Xue, H. (2018). A survey of dimensionality reduction techniques based on random projection (arXiv:1706.04371). http://arxiv.org/abs/1706.04371

Yoon, J., Jordon, J., & van der Schaar, M. (2019). INVASE: Instance-wise variable selection using neural networks. *International Conference on Learning Representations*. https://openreview.net/forum?id=BJg_roAcK7

Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2017). Multiobjective deep belief networks Ensemble for Remaining Useful Life Estimation in prognostics. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2306–2318. https://doi.org/10.1109/TNNLS.2016.2582798

Zhang, S., Sun, Z., Wang, M., Long, J., Bai, Y., & Li, C. (2019). Deep fuzzy Echo state networks for machinery fault diagnosis. *IEEE Transactions on Fuzzy Systems*, *1–1*, 1. https://doi.org/10.1109/TFUZZ.2019.2914617

Zhang, W., Guo, W., Liu, X., Liu, Y., Zhou, J., Li, B., Lu, Q., & Yang, S. (2018). LSTM-based analysis of industrial IoT equipment. *IEEE Access*, *6*, 23551–23560. https://doi.org/10.1109/ACCESS.2018.2825538

Zhao, M., Zhong, S., Fu, X., Tang, B., & Pecht, M. (2020). Deep residual shrinkage networks for fault diagnosis. *IEEE Transactions on Industrial Informatics*, *16*(7), 4681–4690. https://doi.org/10.1109/TII.2019.2943898

Zhao, R., Yan, R., Wang, J., & Mao, K. (2017). Learning to monitor machine health with convolutional Bi-directional LSTM networks. *Sensors*, *17*(2), 273. https://doi.org/10.3390/s17020273

Zhao, X., Jia, M., Bin, J., Wang, T., & Liu, Z. (2021). Multiple-order graphical deep extreme learning machine for unsupervised fault diagnosis of rolling bearing. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–12. https://doi.org/10.1109/TIM.2020.3041087

Zhao, X., Jia, M., & Liu, Z. (2021). Semisupervised graph convolution deep belief network for fault diagnosis of Electormechanical system with limited labeled data. *IEEE Transactions on Industrial Informatics*, *17*(8), 5450–5460. https://doi.org/10.1109/TII.2020.3034189

Zheng, J., Wang, H., Song, Z., & Ge, Z. (2019). Ensemble semi-supervised fisher discriminant analysis model for fault classification in industrial processes. *ISA Transactions*, *92*, 109–117. https://doi.org/10.1016/j.isatra.2019.02.021

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. *IEEE International Conference on Prognostics and Health Management (ICPHM), 2017*, 88–95. https://doi.org/10.1109/ICPHM.2017.7998311

Zhou, L., Duan, F., Corsar, M., Elasha, F., & Mba, D. (2019). A study on helicopter main gearbox planetary bearing fault diagnosis. *Applied Acoustics*, *147*, 4–14. https://doi.org/10.1016/j.apacoust.2017.12.004

Zilke, J. R., Loza Mencía, E., & Janssen, F. (2016). DeepRED – Rule extraction from deep neural networks. In T. Calders, M. Ceci, & D. Malerba (Eds.), *Discovery science* (pp. 457–473). Springer International Publishing. https://doi.org/10.1007/978-3-319-46307-0

AUTHOR BIOGRAPHIES

Taha Mansouri received his Ph.D. degree in information technology from Allameh Tabataba'i University, Iran, in 2016. He is currently pursuing his second Ph.D. degree in Artificial Intelligence at the University of Salford, U.K. Additionally, he works as a lecturer in Artificial Intelligence at the same university, where he used to be a Research Associate in the IoT. Taha's research has been published in highly respected journals such as Knowledge Based Systems, Expert Systems with Applications, Applied Soft Computing, Optic Letters, IEEE Access, International Journal of Production Economics, and Telecommunication Systems. His research interests focus on deep learning, computer vision, and natural language processing.

Sunil Vadera received the Ph.D. degree in computer science from The University of Manchester, in 1992. He is currently a Professor in computer science with the University of Salford, U.K., where he has served in many leadership roles, including as the Dean and the Head of the School of Computing, Science and Engineering, from 2011 to 2019. He is a fellow of the British Computer Society, a Chartered Engineer (C. Eng.), and a Chartered IT Professional (CITP). He was awarded the U.K. BDO Best Indian Scientist and Engineer in recognition of his contributions to computing, science, and engineering in U.K., in 2014. He was the Chair of the U.K. BCS Knowledge Discovery and Data Mining Symposium, Salford, in 2009. He has been the General Chair of numerous conferences, including the IFIP Conference on Intelligent Information Processing, in 2010, 2012, 2014, and 2016. He was the Organizing Chair of a workshop on Cost sensitive learning at the IEEE International Conference on Data Mining, in 2012. He is passionate about closing the gap between theory and application of AI and leads the University of Salford's contribution to the Greater Manchester AI Foundry Project that supports the development of innovative AI applications by SMEs.

How to cite this article: Mansouri, T., & Vadera, S. (2023). Explainable fault prediction using learning fuzzy cognitive maps. *Expert Systems*, e13316. https://doi.org/10.1111/exsy.13316