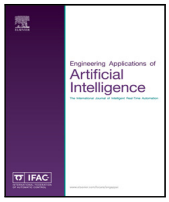
Engineering Applications of Artificial Intelligence 117 (2023) 105596

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Interpreting the antecedents of a predicted output by capturing the interdependencies among the system features and their evolution over time

Sonia Farhana Nimmy [a,*], Omar K. Hussain [a], Ripon K. Chakrabortty [b], Farookh Khadeer Hussain [c], Morteza Saberi [c]

[a] School of Business, University of New South Wales, Canberra, Australia
[b] School of Engineering and Information Technology, University of New South Wales, Canberra, Australia
[c] School of Computer Science, University of Technology Sydney, Sydney, Australia

ARTICLE INFO

Keywords:
Black-box
Explainable AI (XAI)
Glass-box
Interpretable models

ABSTRACT

Decision support systems (DSS) assist in a wide array of decision-making tasks in different domains. However, one of their common drawbacks is that their working is back box in nature. This means that while they recommend a decision, they cannot explain the 'why' behind reaching that decision. In prescriptive tasks such as risk management, this does not assist the risk manager in identifying the contributing features leading to the occurrence of a risk output against which corrective action/s needs to be taken. This limitation has sparked interest in explainability, where glass-box methods interpret the contributing features leading to the recommended decision. Approaches which do that however do not model how the contributing features' evolve over a period of time, till the predicted time period, to determine the output class before interpreting the reason for the decision output. To address these gaps, in this work we propose An Automated Interpretable Artificial Intelligence framework for Proactive Risk Management (AIAI-PRM). AIAI-PRM augments Local Interpretation-Driven Abstract Bayesian Network (LINDA-BN) with Knowledge Graph to determine the inter dependencies among the features, model how they evolve over a period of time and interpret the contributing features leading to the recommended output. In the domain of risk management, we show how this knowledge can be used by the risk manager to determine those key features against which risk management strategies need to be developed. Finally, we compare the AIAI-PRM's output with that of the most commonly used XAI approaches, namely LIME and SHAP, to prove its superiority.

1. Introduction

Decision Support Systems (DSSs) are used by decision makers in different domains to assist in their decision making. One such domain in which DSSs have been widely applied is proactive risk management in different areas such as predictive maintenance, fault detection, energy consumption monitoring, operational planning etc. (Luo et al., 2019). In contrast to reactive risk management, the objective of proactive risk management is to pre-determine the risk factors that have a chance of occurrence and implement plans to prevent or control either their occurrence or impact (Smith and Merritt, 2020). In recent years, Artificial Intelligence (AI) techniques such as Bayesian networks, Artificial Neural Networks, Fuzzy Logic, Markov models, and Support Vector Machines have been used to assist in proactive risk management. While these methods apply complex learning techniques that achieve high accuracy in their outputs, they have sophisticated internal computations and are thus termed "black box" in terms of how they generate an output (Gejke, 2018). This means that such systems show to the risk managers or decision makers the recommended outcome/s sans the reason why these outcome/s are recommended (Lakkaraju et al., 2019). Because of this drawback, these models do not assist decision makers to fully trust the outcome and the alternatives being recommended. To address these gaps, a new area of research, eXplainable AI (XAI), is being worked on in the literature. XAI emphasises the need for AI techniques to explain to the decision makers their black-box based outputs so that they can be interpreted and trusted (Dhanorkar et al., 2021). In other words, XAI explains the black-box model's output to allow (human) decision makers to confirm several aspects, such as fairness, reliability, robustness, causality, privacy and trust of a machine learning model (Ventura et al., 2018; Lambert et al., 2014). This is needed for the decision makers to trust the results recommended by AI methods, especially, in the enterprise domain where incorrect solutions can lead to serious errors (Liao et al., 2020).

* Corresponding author.
E-mail addresses: s.nimmy@adfa.edu.au (S.F. Nimmy), o.hussain@adfa.edu.au (O.K. Hussain), r.chakrabortty@adfa.edu.au (R.K. Chakrabortty), farookh.hussain@uts.edu.au (F.K. Hussain), morteza.saberi@uts.edu.au (M. Saberi).

https://doi.org/10.1016/j.engappai.2022.105596
Received 3 April 2022; Received in revised form 26 September 2022; Accepted 31 October 2022
Available online 15 November 2022
0952-1976/

Techniques such as SHAP (SHapley Additive exPlanations) (Bellucci et al., 2021; Angelov et al., 2021), LIME (Local Interpretable Model-agnostic Explanations) (Visani et al., 2020; Došilović et al., 2018), Anchors (Lin et al., 2021), CEM (Contrastive Explanation Method) (Burkart and Huber, 2021), LINDA-BN (Local Interpretation-Driven Abstract Bayesian Network) (Moreira et al., 2021) have been proposed in the XAI categories of approaches. However, these techniques are unable to:

1. Automatically model how the values of a feature evolve over the decision making time period: While existing XAI approaches model the influence of the key features towards an output class they require the values of these features to be manually inputted to them. In other words, for a time slot 't', the features' values need to be given for the XAI model to interpret the contribution of features towards the decision of that time slot. This works well when the post-hoc explainer need to explain the contributing features towards the decision in the same time slot for which they are given the inputs for. However, DSS are required to work over multiple, progressive time slots over time. So for them to explain the influencing features for a decision in time slot '$t + n$', the post-hoc explainer first needs to model how each input feature evolves till the time slot of interest, what the output decision class is before interpreting the decision output.
2. Visualise not just the key features influencing a decision but also the chain of contributing features behind them: In showing their outputs, existing XAI approaches represent the most influential features towards an output class. For taking proactive actions, a decision maker, apart from knowing the influential features should also know the contributing features that are the building blocks behind the influential features towards reaching that decision. In other words, the decision maker should be able to visualise why the output risk class has reached and what is the role of each feature towards that output. This will be of much use when preventive actions are being developed to address the problem not just by its symptom but from its root cause.

Addressing these two requirements are particularly important for XAI approaches to be applied in DSS that work over progressive time slots. In this paper we develop an *A*utomated *I*nterpretable *AI* framework for *P*roactive *R*isk *M*anagement (AIAI-PRM) that meets these requirements. The contribution of our proposed approach are:

1. Uses Knowledge Graphs (KGs) to determine interconnections between the different input features of a decision model. These interconnections are then used to ascertain the values of features in different future time slots based on the values of its related features.
2. Determines and visualises the contribution and impact of all the input features on the output decision class. AIAI-PRM augments LINDA-BN with KGs to model the interrelatedness among the features and their contribution towards the output decision class. This assists the decision maker not to focus only on the most obvious features impacting the decision class at a certain time period and thus make proactive strategies to manage those features that are the root cause of failures.

The remainder of the paper is structured as follows. Section 2 presents the literature review. Section 3 defines the case study that is used in this article. Section 4 details the proposed architecture of AIAI-PRM and explains its different modules. Section 5 presents the results of AIAI-PRM and compares it with LIME and SHAP. This section also discusses the limitations of AIAI-PRM. Finally, Section 6 summarises the research findings and concludes the paper with a discussion on future work.

2. Literature review

The need for automated decision-making models to provide their users with transparency and accountability is slowly becoming legal legislation (Malgieri, 2019). Initiatives such as the General Data Protection Regulation (GDPR) require organisations using AI-based decision models to explain to their customers how they have reached a conclusion on their affairs. As previously mentioned, this requirement arisen due to the lack of belief that the adopted process reaches a fair decision, which in turn exposes the companies to legal, social, and ethical implications. To satisfy this requirement, researchers in different domains have attempted to understand why their AI-based models provide them with a given conclusion. For example, Bussmann et al. (2020) propose an explainable AI model in Fintech to measure the risks involved in borrowing through peer-to-peer lending platforms. It does this by computing Shapley values and categorising borrowers as either high-risk or non-risk. This analysis is then used to ascertain their credit score and future behaviour. Chen et al. (2021) propose an artificial neural network (ANN)-based sales performance model based on customer demand and inventory cost control. The authors evaluated the performance of their approach with other models, namely linear regression and support vector machines. They then used SHAP to interpret the results. Li et al. (2019) proposed tree-LIME, a model-agnostic method developed by modifying the LIME algorithm based on a local adaptation by decision tree regression. The proposed method calculates the fidelity measure in the regression explanation using mean absolute error (MAE). Experiments that were conducted on a real-world application for service chain prediction show that the approach can improve the accuracy of the explainer, resulting in more accurate explanations and an enhanced representation for individual instances. Although the methods explain the results, shortcoming of SHAP is that the Shapley values of each variable cannot detect and evaluate the correlated variables and dependencies between integrated network models over a period of time. Similarly, a shortcoming of LIME is that it does not work well for complex and real-time data processing approaches. Moreover, it is challenging to implement LIME for time series data where variable values evolve over time. CEM is another approach for XAI that produces local interpretations for a black box model (Cali et al., 2021). It does this by using pertinent positives (PP) and pertinent negatives (PN) that work by presenting knowledge on the preferred and undesired characteristics (Rothman, 2020). Luss et al. (2021) generates such contrastive explanations by making use of latent features. To support prediction classification, the approach apart from highlighting the aspects that justify the classification also recommends those aspects, which, if added, will improve the result. The most important contribution is joining aspects to enrich data that will lead to generate intuitive explanations. However, a black-box classifier is required for effective application of CEM. Then CEM is applied to the classifier to improve accuracy and describe via explanations (Kenny and Keane, 2021). Giudici and Raffinetti (2021) developed a global explainable AI approach based on Lorenz decompositions, which enable quantifying the additional contribution of each attribute using a statistical test, rather than the value of the predictions directly. Slack et al. (2021) proposed a system that integrates a Bayesian network with LIME and KernelSHAP, namely BayesLIME and BayesSHAP that can also capture the interdependencies of the variables. The primary goal of the study is to narrow down the range of possible causes for the prediction gap. However, what these approaches lack is the ability to automatically determine the feature values at a future time period and graphical representation of the contribution of all the features towards an output class. Having such a representation is important for it to be understood by any user irrespective of the technical knowledge s/he has. In other words, to understand the existing approaches output, users' need technical understanding and knowledge to perceive the interdependencies and evaluate the overall impact of the input features on the system. As a result of these constraints, the system's degree of interpretability is restrained.

While the existing approaches interpret the antecedents to a predicted decision output, they need to consider the dependencies that exist among the different features and automatically model how they evolve till the point when the decision output needs to be interpreted. Unifying approaches such as Alibi (Klaise et al., 2021), captum (Kokhlikyan et al., 2020) and tf-explain Anon (2022) model the time series characteristics, however their focus is not to model how the interconnections of features impacting the predicted output class over time. Rather, they model the uncertainty present to determine the completeness of the data set over time. These approaches consider the various data types (text, tabular, image) in algorithms such as SHAP (Lundberg and Lee, 2017), GradCam (Selvaraju et al., 2017), Counterfactuals (Mothilal et al., 2020) to integrate and adapt existing methods to time series characteristics such as Temporal Saliency Rescaling for Saliency Methods (Ismail et al., 2020), Counterfactuals (Ates et al., 2021; Delaney et al., 2021), LEFTIST based on SHAP/Lime (Guillemé et al., 2019) and design new techniques precisely for time series interpretability. TSViz for interpreting CNN (Siddiqui et al., 2019), TSInsight based on autoencoders (Siddiqui et al., 2021) are examples of this new dimension. However, what they do not do is automatically determine the future values of features. Moreover, outputs from time series are not instinctive for human interpretation (Siddiqui et al., 2019) and need further visualisations. In other words, the existing time series approaches do not automatically model how changing a single feature in a time period cumulatively impacts on correctly interpreting the antecedents of a decision outcome at a future time period (Biecek and Burzykowski, 2021; Fidel et al., 2020). Without this capability, the existing interpretation models may assist in interpreting the key influencing features to arrive at a decision outcome, however they need to be manually given the input values in a time period. While this will assist the decision maker to interpret the features leading to a decision outcome, these may not be the root cause/s of why the risk is occurring. Such analysis of existing models may cause the DSSs to address the symptom/s of the occurrence of the risk rather than solving the root cause/s of its occurrence. To address these gaps, there is a need to design a model that explains to the decision makers why an output that is shown is being produced by determining the interrelatedness and evolution of the features values that impact it over a certain time period. In the next sections, we propose our approach to address this issue.

3. Case study

In this section, we define the case study that we use in this paper. We consider the domain of asset management and the decision maker as the maintenance manager of industrial machines (assets). We consider that each machine has a scheduled downtime for maintenance or repairs but during its uptime, it is expected to be operational in order to meet its objectives. From the perspective of the maintenance manager, proactive risk management during an asset's uptime refers to the process of s/he determining beforehand the chances it failing and devising remedial actions to avoid it from happening. Let us consider that the uptime period for a particular machine that we use in this case study is 12 months. Existing predictive asset management analytics assist the maintenance manager to predetermine the chances of that machine failing over this time period. However, they are black box in their working nature and thus the maintenance manager needs XAI techniques to interpret the decision output for him/her to develop remedial strategies. So the requirements of the maintenance manager are that s/he wants to:

1. Automatically determine and interpret the chances of a machine failing in different times of the uptime period.
2. Ascertain the role and influence of each input feature towards the output class of the machine's operating state in each time period.

The details of the asset's operating state are as follows:

1. Number of input features that are used to predetermine the chances of the machine failing: 23
2. Input features: Pressure, Friction, Crushing, Shearing, Entanglement, Trapping, Stabbing, Efficiency, Moving Parts, Heat, Noise, Vibration, Radiation, Fuel Oil, Fatigue, Jamming (Temp), Jamming (Dist), Workload, Dust, Fumes (PM2.5), Fumes (PM10), Stability and Ferromagnetic
3. Output classes over which the chances of a machine failing is determined: Safe, Low Risk, Medium Risk and High Risk. We consider these as the target risk classes.
4. The uptime period is of 52 weeks and the maintenance manager wants to determine the machine's operational state along with interpreting the reason behind that state each week.

In the next section, we explain how our proposed AIAI-PRM framework assist in achieving these requirements.

4. An automated interpretable artificial intelligent framework for proactive risk management (AIAI-PRM)

Fig. 1 shows the AIAI-PRM framework which augments LINDA-BN with KG to generate post-hoc model-agnostic trials to identify which features led to the prediction of the given risk class. AIAI-PRM comprises four modules, namely *Feature Selection Module (FSM)*, *Knowledge Propagation Module (KPM)*, *Mathematical Reasoning Module (MRM)*, and *Visualisation Module (VM)*. FSM is a semi-automated module that enables the maintenance manager to select those key features (KFs) from the given input set of features whose impact or role s/he wants to analyse on the output class. KPM is an automated module which determines the secondary features (SFs) related to a KF. An SF is a feature that either positively or negatively influences a KF. KPM also determines the interdependence among the various SFs and their effect on a KF, based on a change in their value. MRM uses the dependency between the features with respect to conditional probabilities and determines the output class in each time period and the impact of features on the output class. VM for each time period represents how the KFs and their SFs have contributed to the output risk class. In the next subsections, we provide a detailed explanation of the step-by-step working of each module.

4.1. Feature Selection Module (FSM)

FSM comprises of the following three steps:

Step 1: Defines the time space and categorises it into different time slots:

Time space, represented by T is the total time over which the activity is carried out. *Time slots*, represented by $(t_1 ... t_t)$ are the different non-overlapping sequential periods of time in T. For example, in the considered case study, as the maintenance manager wants to determine the risk of an asset failing weekly over a period of 52 weeks, then the total space is of 52 weeks and each time slot is one week, represented as t_1 to t_{52}.

Step 2: Allows the decision maker to select KFs of interest in each time slot:

In this step, the decision maker (maintenance manager in our case study) selects the KFs whose impact on the target risk classes s/he wants to analyse. For example, continuing the case study, if the maintenance manager in each time slot wants to ascertain the contributing role of 'dust', 'fuel oil temperature' and 'efficiency' on the output risk classes of 'Safe', 'Low Risk', 'Medium Risk' and 'High Risk', then these features are selected as the KFs and stored in the *Risk Table* as shown in Table 1. The *Risk Table* represents the features whose impact on the output risk classes need to be determined over the time space of the activity.

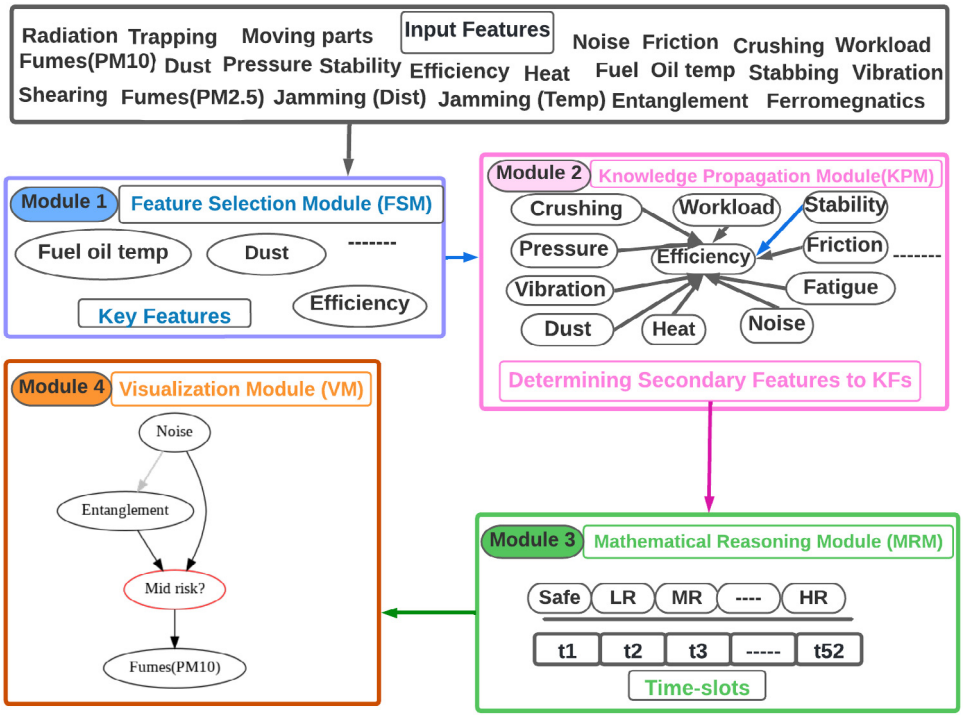


Fig. 1. AIAI-PRM's methodology to interpret the features contributing to a decision class.

Table 1
Risk Table storing the KFs with the output risk classes.

Key features/ output risk classes	Fuel oil temperature	Efficiency	Dust
Safe	$X_{start}...X_n$	$Y_{start}...Y_a$	$Z_{start}...Z_f$
Low risk	$X_{n+1}...X_m$	$Y_{a+1}...Y_b$	$Z_{f+1}...Z_g$
Medium risk	$X_{m+1}...X_o$	Y_{b+1} ...Y_c	Zg+1...Z_h
High risk	$X_{o+1}...X_{end}$	$Y_{c+1}...Y_{end}$	$Z_{h+1}...Z_{end}$

Step 3: Augment the KFs and output risk classes with Lifetime (L) and Degree of Influence (I) metrics:

In this step, FSM augments the KFs and output risk classes from Table 1 with the *Lifetime (L)* and *Degree of Influence (I)* metrics. *L* represents the numerical range across which both the KFs and the target risk classes span. This is determined using the available *Knowledge (K)*, that comes either from the expert knowledge or operational documents or manuals. For example, continuing with the case study, if the *K* from the operational documents mentions 'the flash point of the fuel oil is 93 °C or more after which it will catch fire and burn the motor', then we can determine the *L* of *fuel oil temperature* has a minimum value of 1 °C and a maximum value of 250 °C (from the expert knowledge). These values are represented as X_start and X_end respectively in Table 1. However, from *K*, it is known that a value of 93 °C and above leads to the occurrence of fire and thus that range for this feature is classified under the output risk class of 'High Risk'. So, the *I* of the output risk class 'High Risk' starts from 93 °C. Using either the expert knowledge or operational documents, we can further classify the remaining range of *L* of *Fuel Oil temperature* to the different output risk classes. For example, the temperature between 1 °C–25 °C can be linked to the output risk class of 'Safe', between 26 °C–50 °C to the output risk class of 'Low Risk', and between 51 °C–92 °C to the output risk class of 'Medium Risk'. Fig. 2 shows how the *L* and *I* metrics for each KF and output risk class are determined from its *K*.

4.2. Knowledge Propagation Module (KPM)

KPM is an automated module that determines the SFs to a KF, the interdependence among them and the effect on a KF based on a change in the SFs. These aims are achieved in the following two steps:

Step 1: Develop a knowledge base (KB) and derive inferences to determine the SFs to the KFs:

A knowledge base (KB) is a data set which contains facts about a system or its workings. The knowledge from this data set is used to derive automated deductive inferences either to reason about known facts or derive new facts (Tiddi et al., 2020; Holzinger et al., 2018). In other words, the KB stores structured and unstructured information which can be used by an expert system to derive further inferences from it (Pathak et al., 2018). KBs also have built in artificial intelligence capabilities that can interact with the underlying data and respond to a user's query. Fig. 3 shows how KPM uses automated deductive reasoning to not only derive the SFs influencing the KFs but also the intensity of the impact. For example, for the KF, 'Efficiency', the fact ***Pressure negatively effects Efficiency*** infers that *Pressure* is an SF to it and there is an inversely dependent relationship between them. Once the dependence of a KF on an SF is determined, KPM then infers the other features that influence the SF to ascertain the chain reaction they can have on the KF. Each SF identified to influence the KF is stored in the *Risk Table* (Table 1) and FSM is used to determine their *K* and augment them with the *L* and *I* metric values corresponding to each output risk class.

Step 2: Represent the linkages between the features using a knowledge graph (KG):

In the domain of knowledge representation and reasoning, a KG is a graphically structured data model or topology that integrates data from a KB (Jia et al., 2018; Ko et al., 2021). In other words, KGs represent interrelated descriptions of entities – objects, events, situations or abstract concepts – with freely formulated semantics (Mohamed et al., 2021). KPM uses the modelled relationship between the features from the KB and develops a KG to represent a linked relationship between them.

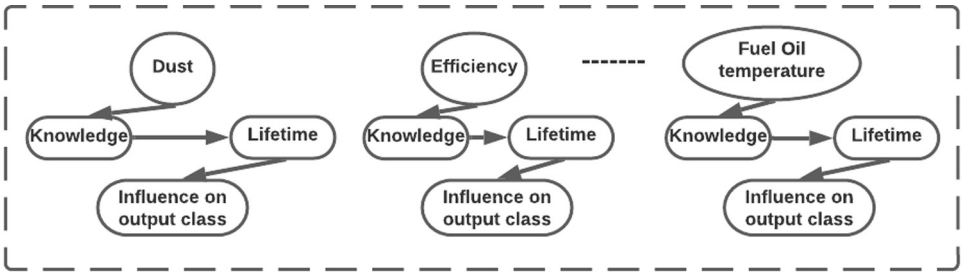


Fig. 2. Process of extracting the L and I metric values of a feature from its K.

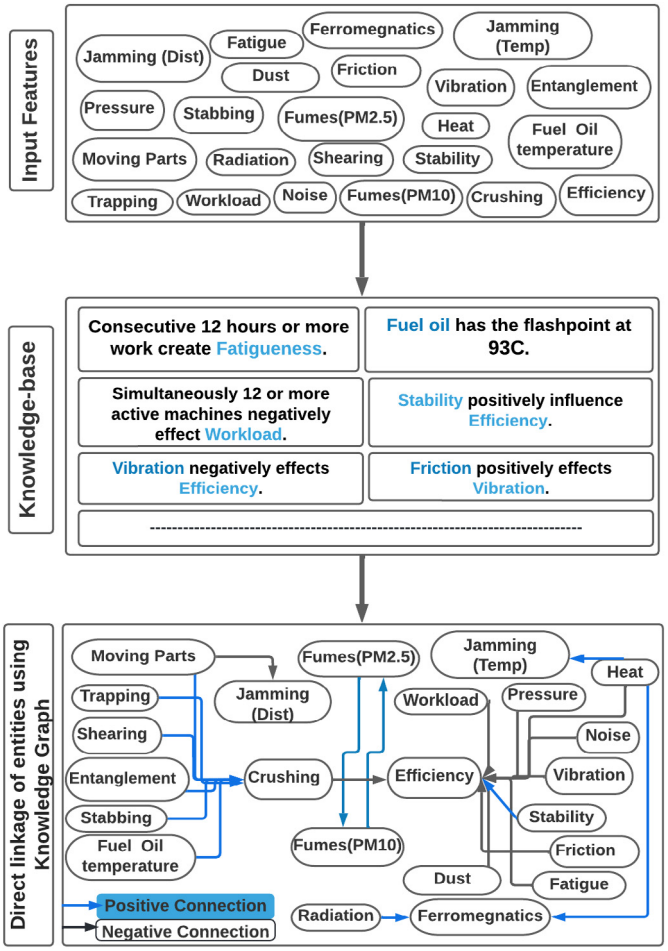


Fig. 3. Determining the SFs to the KFs and the impact they will have.

Furthermore, the KG also indicates if the interdependence between the features is of a positive or negative nature. A positive nature means that an increase in the value of SF will increase the KF's value and a negative nature means otherwise. Using the metric I and KG, it is possible to define the problem boundaries, and identify how the changes in features will affect the output according to the level of influence between them (Chen et al., 2020). Fig. 3 shows how KPM uses the KB to develop a linked relationship between the KFs and SFs along with the polarity of the influence between them.

The pseudo code by which the SFs to a KF are determined is shown in Algorithm 1. After updating the feature values using the above Pseudocode, we the applied the LINDA-BN application. The application of LINDA-BN is available on https://github.com/catarina-moreira/LINDA_DSS (Moreira et al., 2021).

Algorithm 1: Determining the SFs to a KF

Input: K is knowledge, L is a feature from K, I is the related feature with L, and time slot, T$\epsilon \{T_1, T_2, T_3, \cdot, T_n\}$.
Output: Related feature, I
Process:

```
while i ≥ length[K] do
    for each time slot, T_1, T_2, ⋯, T_n
    if K is true then
        while j ≥ length[L] do
            if L[j] related to knowledge, K then
                Update related feature, I←L[j]
            else
                I← I                    ▷ No related features to update
            end
        end
    else
        if K is true then
            Continue;
        end
    end
end
```

4.3. Mathematical Reasoning Module (MRM)

Once KPM determines the links between the features, MRM in each time slot first determines the conditional probability of each feature being in each output risk class. This analysis is used to determine the probability of the occurrence of each output risk class in each time slot. MRM then uses LINDA-BN (Moreira et al., 2021) to determine the main contributing features that lead to the occurrence of the most probable risk class in each time slot. MRM achieves these aims in the following three steps:

Step 1: Permutation of KFs to represent their consequent and antecedent SFs:

In this step, MRM permutes the KFs using uniform distribution with permutation variance, ϵ, where $\epsilon \, \epsilon \, [0, 1]$. Such permutation for each KF is performed in each time slot over the interval $[F_i - \epsilon, F_i + \epsilon]$. It means that, if in time-slot T_n, a feature value is F_i, then in time-slot T_{n-1} the feature value was $F_{i-\epsilon}$ and in time-slot T_{n+1}, feature value will be $F_{i-\epsilon}$. Having such a representation will assist MRM to determine the impact of each KF and/or SF on the output risk classes in each time slot using Bayesian network modelling.

Step 2: Bayesian network (BN) modelling to ascertain the impact of each feature on the output risk classes:

In this step, a BN is used to determine the probability of each SF or KF contributing to each output risk class by considering their numerical value and dependence on other features. This analysis determines the probability of the occurrence of each output risk class in each time slot. BN models the dependencies between two or more features and

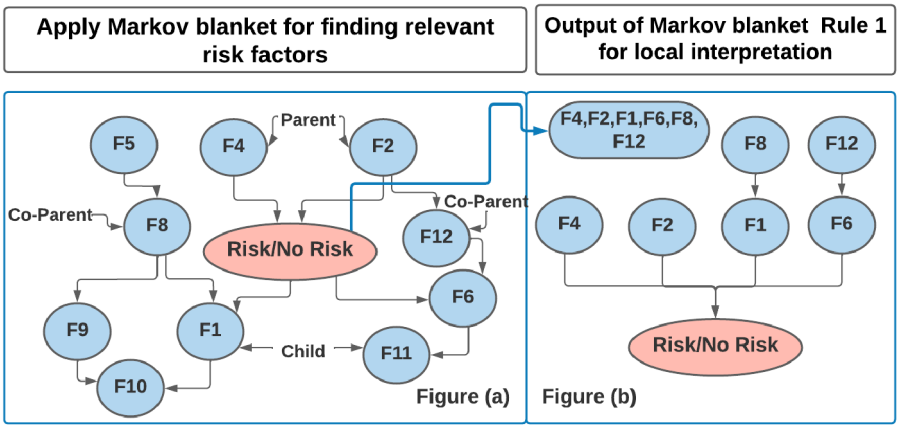


Fig. 4. Selection process using Markov blanket of a target variable.

according to the conditional probabilities between them, ascertains their impact on their consequent feature (Marcot, 2017). A BN is a directed acyclic graph where each node represents a feature, and each edge represents the connectivity between any two features. The dependency and nature of the relationship between one or more SFs and the impact that they will have on the KF is captured by graphical reasoning, which is a structural representation of a BN. Using such dependencies, the impact of a feature on an output risk class can be determined by a Bayesian classifier. The output of this step is shown in Fig. 4(a), which for an output risk class in a time slot is a union of conditionally dependent SFs from its parents, child and co-parent (parent of a child) that directly or indirectly impacts it. The process of achieving such representation is explained next.

Let the BN graph for the features $F_1, F_2, \ldots., F_n$ be represented by B. The probability of P over the sample for graph B can be represented by Koller and Friedman (2009).

$$P(F_1, F_2, \ldots, F_n) = \prod_{i=1}^{n} P(F_i | Pa_{F_i}) \tag{1}$$

where Pa_{F_i} represents the all parents variables for features F_i.

Bayesian networks work with all the variables using the full joint probability theory for inference. For some events E and some observed variable v, the inference of the Bayesian network can be calculated using the following equation (Koller and Friedman, 2009).

$$P(E|V = v) = \alpha P(E, v) = \alpha \sum_{w \epsilon W} P(E, v, w), \quad with\ \alpha = \frac{1}{\sum_{e \epsilon E} P(e, v)} \tag{2}$$

where W refers to the set of random variables that are neither in events nor evidence.

A BN has two principal parameters: a directed acyclic graph B and a set of conditional probability parameters ϕ that represents the conditional dependency. Given a dataset d with n observations, $P(B, \phi|d)$ is divided into two phases: structure learning and parameter learning which are represented as follows (Scutari et al., 2019):

$$P(B, \phi|d) = p(B|d).P(\phi|B, d) \tag{3}$$

where $p(B|d)$ defines structure learning and $P(\phi|B, d)$ defines parameter learning.

Structure learning is used to design the directed acyclic graph B by maximising $P(B|d)$. Parameter learning focuses on probability parameter ϕ which is obtained from structure learning. Considering parameter ϕ with an independent distribution, the learning process can be described as follows (Heckerman et al., 1995; Heckerman, 2008):

$$P(\phi|B, d) = \prod_{i} P(\phi_{F_i} | \prod F_i, d) \tag{4}$$

However, structure learning is an NP-hard and NP-complete problem due to following equation (Karci, 2020):

$$P(B|d) \propto P(B)P(d|B) \tag{5}$$

P(d|B) can be decomposed into:

$$P(d|B) = \int P(d|B, \phi)P(\phi|B)d\phi \tag{6}$$

$$= \prod_i \int P(F_i | \prod F_i, \phi_{F_i}) P(\phi_{F_i} | \prod F_i) d\phi_{F_i} \tag{7}$$

In structure learning, a Bayesian information criterion (BIC) is used to find the maximum score that can be expressed by the following equation:

$$\text{Score(B, d)} = \text{BIC(B, } \phi|d) = \sum_i log P(F_i | \coprod F_i, \phi_{F_i}) - \frac{log(n)}{2} |F_i| \tag{8}$$

LINDA-BN applies the Hill climbing approach to learn structure B for simplicity and for an effective result (Taheri et al., 2011). Once the conditional probabilities of each SF are determined, the abductive reasoning process (Gabbay and Woods, 2006), which is a human inference to form a conclusion from the known information, is used ascertain the state of the target KF or output risk class. An abduction inference for a feature provides a trustable explanation with respect to the graphical structure (Gabbay and Woods, 2006).

Step 3: Developing a Markov blanket for each KF:

As shown in Fig. 4(b), a Markov blanket is a tree-based structure of the most influential features that lead to the occurrence of an output risk class. Having this Markov blanket representation for a KF or an output risk class enables the decision maker to see the impact of each KF or SF in each time slot. Markov blanket generates four different rules based on four different situations (Moreira et al., 2021). Among them, only Rule 1 is used (Fig. 4(b)) that provides a high level of confidence, this means that the recommended output guaranteed accuracy and trustworthiness.

4.4. Visualisation Module (VM)

The role of VM is to represent the Markov blanket for each output class or KF in each time slot to the decision maker. This assists them to interpret the reason behind the occurrence of an output risk class in each time slot in a chain-based format. In the next sections, we detail the working of AIAI-PRM and compare its ability to provide an interpretative chain of reasoning behind the occurrence of a KF or output risk class over the activity time period.

5. Demonstration and comparison of AIAI-PRM against SHAP and LIME in providing an interpretable output to the maintenance manager

In this section, we compare the output of AIAI-PRM with SHAP and LIME to show its advantages and benefit in giving a logical and interpretable output.

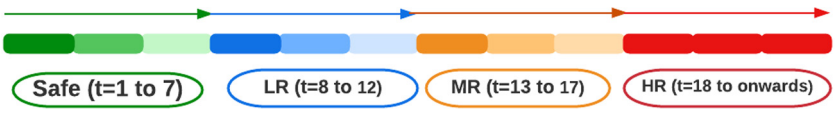


Fig. 5. Most probable output risk classes across the different time slots of the time space.

Table 2
Risk Table by considering the KFs.

Key features/ output risk classes	Workload	Fumes (PM2.5)	Fumes (PM10)	Noise	Entanglement
Safe	1 to 12	00 to 35.40	00 to 75	00 to 70	30 to 20
Low risk	13 to 17	35.41 to 35.90	76 to 85	71 to 80	19 to 15
Medium risk	18 to 23	35.91 to 36.40	86 to 95	81 to 85	14 to 07
High risk	24 to 28	36.41 to 40.00	96 to 103	86 to 100	06 to 01

Table 3
Extended Risk Table by considering all the KFs and SFs.

Key features/ Values	Safe	Low risk	Medium risk	High risk
Pressure	29.6 to 30.2	30.3 to 30.6	30.7 to 31.2	31.3 to 32
Friction	1 to 70	71 to 75	76 to 84	85 to 99
Crushing	6 to 5	4.9 to 4	3.9 to 2	1.9 to 1
Shearing	8 to 10	11 to 15	16 to 24	25 to 30
Entanglement	30 to 20	19 to 15	14 to 10	09 to 01
Trapping	30 to 20	19 to 15	14 to 10	09 to 01
Stabbing	30 to 20	19 to 15	14 to 10	09 to 01
Efficiency	100 to 75	74 to 61	60 to 51	50 to 10
Moving Parts	50 to 150	151 to 174	175 to 199	200 to 240
Heat	1 to 50	51 to 70	71 to 85	86 to 100
Noise	00 to 70	71 to 80	81 to 85	86 to 100
Vibration	0.1 to 1.6	1.7 to 1.9	2 to 2.3	2.4 to 3
Radiation	0 to 20	21 to 22	23 to 26	26.1 to 27.4
Fuel Oil	0 to 60	61 to 75	76 to 85	86 to 93
Fatigue	0 to 24	25 to 34	35 to 47	48 to 56
Jamming (Temp)	1 to 60	61 to 68	69 to 84	85 to 100
Jamming (Dist)	6 to 5	5.1 to 3.5	3.4 to 2	1.9 to 1
Workload	1 to 12	13 to 17	18 to 23	24 to 28
Dust	1 to 50	51 to 174	175 to 250	251 to 320
Fumes (PM2.5)	00 to 35.40	35.41 to 35.90	35.91 to 36.40	36.41 to 40.00
Fumes (PM10)	00 to 75	76 to 85	86 to 95	96 to 103
Stability	100 to 75	74 to 69	68 to 61	60 to 50
Ferromagnetic	0.1 to 27.6	27.7 to 28	28.1 to 29.4	29.5 to 30

5.1. AIAI-PRM's output

Let us consider that the maintenance manager selects *Fumes (PM10), Workload, Fumes (PM2.5), Noise, Entanglement* as the five KFs of interest in each time slot. As discussed in Section 4.1, the FSM of AIAI-PRM augments these KFs with the L and I metrics using its K (Fig. 2) and creates the Risk Table as shown in Table 2. For each KF, the KPM of AIAI-PRM augments the SFs to it using the KB. This process results in the addition of 18 SFs, increasing the total number of features being modelled in the system to 23. Fig. 3 shows the linkages between the 23 features (KFs and SFs) using KG. For each of the identified SFs, FSM associates them with L and I metrics using their K. A snapshot of the extended Risk Table is shown in Table 3. AIAI-PRM then uses the input values for the five KFs in t_1 as its starting point and models the values of the dependent SFs to ascertain the probability of the occurrence of each output class in each time slot of the time space. Fig. 5 shows the most probable output risk class occurring in each time slot of the time space. From the figure, it can be seen that the most probable output class in $t_1 - t_7$ is *Safe* after which it moves to *Low Risk* till t_{12}. From t_{13}–t_{17}, the most probable output risk class is *Medium Risk* followed by *High Risk* from t_{18} onwards.

While the analysis of Fig. 5 shows the maintenance manager the most probable output risk class in different time slots, which existing prediction approaches also do, it does not represent the reasons or features that will lead to the occurrence of that output risk class. As shown in Fig. 6, AIAI-PRM does this using LINDA-BN which demonstrates how each feature has contributed to the output risk class *Safe*. As the outputs shows by AIAI-PRM correspond to Rule 1 of LINDA-BN, there is high confidence in the accuracy of the predictions. However, as previously mentioned, with an increase in the number of features, the representation becomes complex for a human to understand. So, the Markov blanket of the key features corresponding to an output class instead of the entire Bayesian network assist the decision maker to interpret locally the logic behind the occurrence of an output risk class. For the output class *Safe*, Fig. 7 shows the Markov blanket which represents only those features that directly contribute to the corresponding output class. Similar sort of analysis can be represented for the other output risk classes but due to space limitations we do not show it for all of them except for the output class of High Risk in Fig. 8.

By finding the key features related to an output risk class, AIAI-PRM can determine how the values of these features change or evolve from the first time slot. For example, as shown in Fig. 8, four key features contribute to the output risk class *High Risk* from time slot t_{18} onwards. Table 4 shows how the values of each of these features change from the beginning of the time space to time slot t_{18} from which the model gives an output of *High Risk*. This allows the maintenance manager to understand how the values of the most influential features to an output risk class are changing and how addressing these features will manage the occurrence of the output risk class. Using Table 4, the maintenance manager can attempt to modify, update or change the feature values either to make the whole system *Safe* or reduce the risk of failure. A similar sort of analysis can be determined for the other output classes in other time slots.

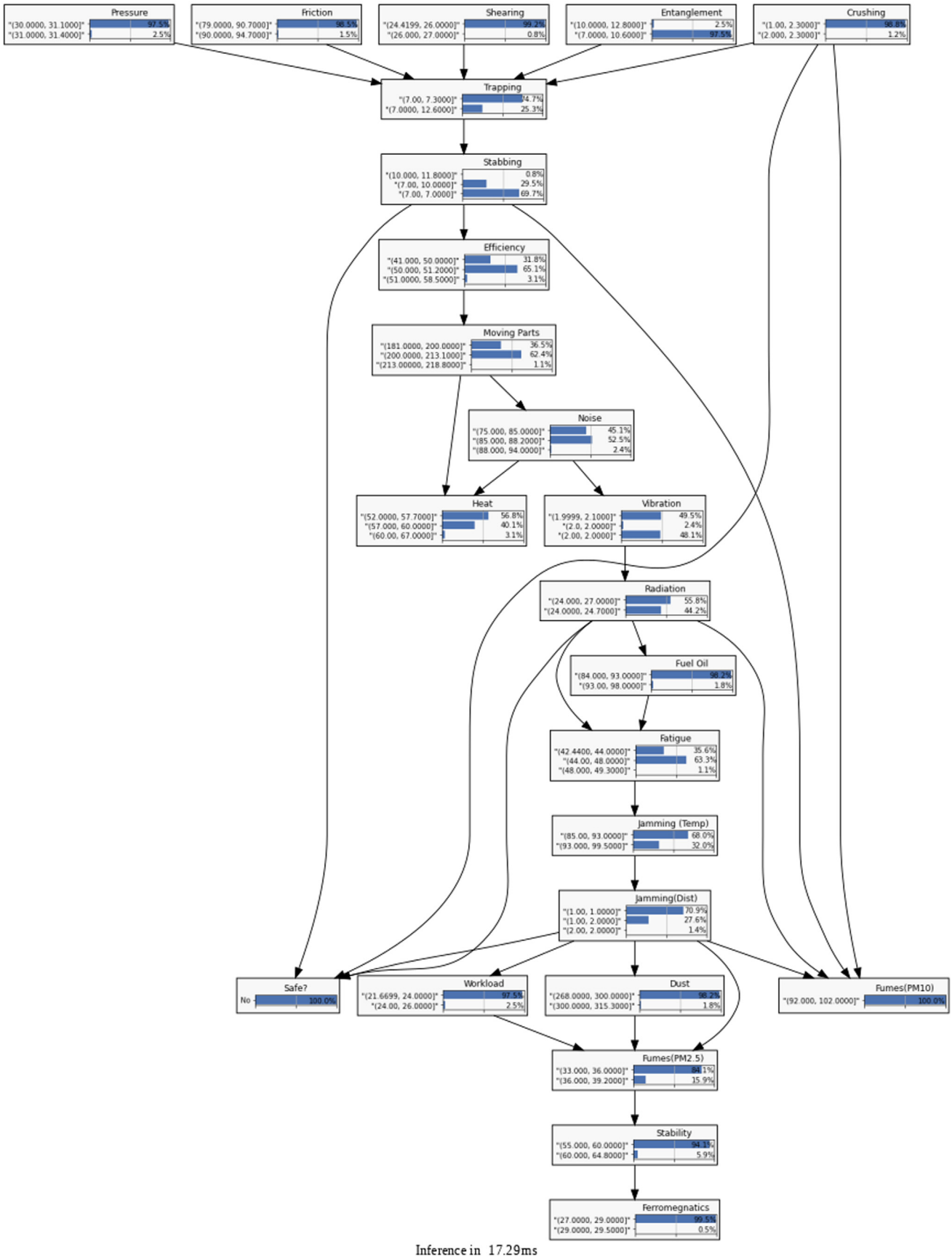


Fig. 6. Relationship between features that contribute to the output class *Safe*.

Fig. 7. Features that directly contribute to the output risk class *Safe*.

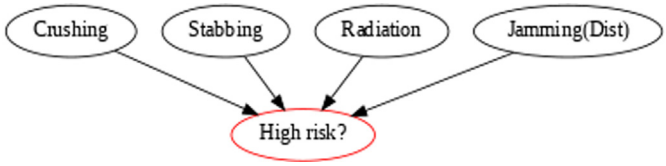


Fig. 8. Features that directly contribute to the output risk class *High Risk*.

Table 4
Change in the values of features related to the occurrence of output class — High Risk over the time slots.

Time/ Features	Crushing	Radiation	Stabbing	Jamming (Dist)	Risk
Time1	6.0	0.00	30	6.0	Safe
Time2	5.9	5.00	29	5.9	Safe
Time3	5.8	7.00	29	5.7	Safe
Time4	5.6	10.0	28	5.5	Safe
Time5	5.4	18.0	26	5.3	Safe
Time6	5.2	18.0	24	5.1	Safe
Time7	5.0	20.0	22	5.0	Safe
Time8	4.8	21.0	21	4.0	LR
Time9	4.6	21.4	19	3.8	LR
Time10	4.4	21.6	17	3.6	LR
Time11	4.2	21.8	16	3.4	LR
Time12	4.0	22.0	15	3.2	LR
Time13	3.9	22.0	14	3.0	MR
Time14	3.5	22.0	13	2.8	MR
Time15	2.8	23.0	12	2.5	MR
Time16	2.4	24.0	11	2.2	MR
Time17	2.2	26.0	10	2.0	MR

Table 5 shows the influence of the KFs on the output risk class in each time slot. Such an analysis assists the maintenance manager in forming a chain between the KF and its related SFs to understand which features need to be focused on to reduce the risk of failure. For example, while the output risk class in time slot t_3 is *Safe*, from Table 5, it can be seen that the KF *Noise* has an influence of only 45.1% on it. Fig. 9 shows a snapshot of the relationship of KF *Noise* with the other SFs that influence it. Using this analysis, it can be interpreted that *Noise's* low level of influence on the output class in that time slot is because of the SFs, *Moving Parts, Efficiency, Stabbing* etc. Such analysis can be used by the maintenance manager in attempting to address the risk of failure not just by its symptoms but from its root cause/s.

5.2. LIME's output

LIME is an interpretable machine learning framework that can be used to explain independent instance predictions arithmetically. As LIME does not capture the interdependence among the features and model how they evolve across different time slots, we gave the values of all the 23 features (which we derived using KG) at the beginning of a time slot at which the output risk class had to be determined. LIME gave us the most influential features contributing to that output class by decomposing the features to their values and determining how increasing or decreasing these values can impact the output class risk. The decision maker can also specify a numerical probability of turning a feature off or on in each variation sample so that they can observe only the features with the highest contribution. Fig. 10 shows LIME's

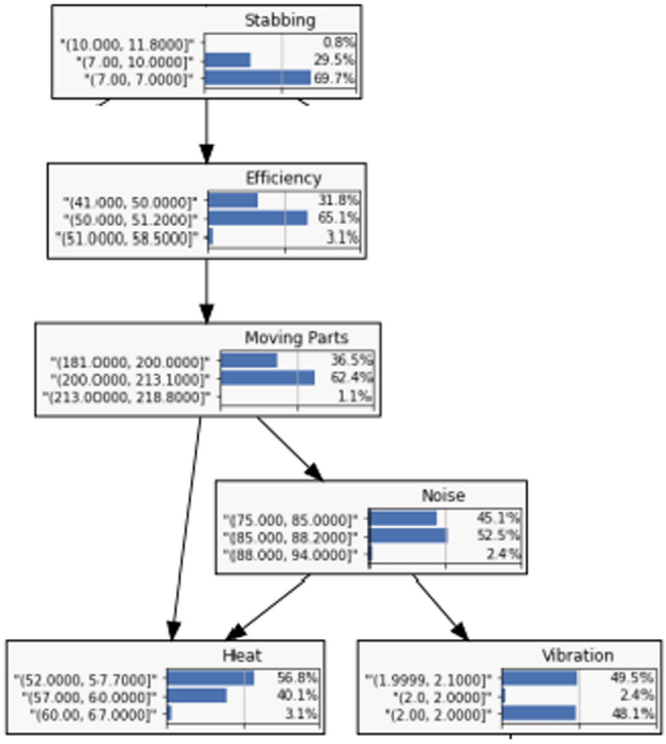


Fig. 9. Influences of different connected features on the KF *Noise*.

output for time slot t_{18}. Fig. 10(a) shows that LIME predicts *High Risk* as the output risk class with a probability of 59% in that time slot. This is consistent with that predicted by AIAI-PRM in Table 5. The most common features associated with LIME's output are *Workload, Radiation, Fumes (PM2.5)* and *Fuel oil*. Of these, the contribution of the feature *Radiation* is the highest at 99% while *Workload* has the lowest contribution of 85%. The other features such as *Jamming (Temp), Moving Parts, Fatigue* and *Pressure* lead to an output that is different to *High Risk*. Fig. 10(b) shows the features along with their contribution which is either to or against the recommended output risk class in that time slot. A similar sort of analysis can be plotted for the output risk classes, *Safe (S), Low Risk (LR)* and *Medium Risk (MR)*.

5.3. SHAP's output

SHAP provides local interpretability of the recommended output by identifying the impact of each feature on the prediction (Matin and Pradhan, 2021). But similar to the shortcoming of LIME, SHAP also does not work with time series problems (Molnar, 2020). While SHAP can explain the importance of the input features to a black box model's output, it does this with a lack of understanding of how the model works. Thus, when modelling the cumulative impact of features over a time period, Deep Explainer does not consider the temporal impact of the features and requires a defined aggregation from the user. To address this, we gave as input the values of all the 23 features at a time slot which we derived using KG. Using these values as inputs, SHAP gives a general understanding of the influence of a feature, based on the additive property. For time slot t_{18}, Fig. 11 shows the output class being recommended, *High Risk (HR)* along with the four different features, namely *Workload, Trapping, Radiation* and *Fuel Oil* which positively influence it. The probability of the occurrence of output risk class *High risk* is 59% in which the contribution of the feature *Radiation* is highest at 99% while the contribution of the feature *Trapping* is the lowest at 31%. A similar analysis can be determined for the other output classes.

Table 5
Influence of KFs on the most probable output risk class in each time-slot.

Time-slots	Output risk class	KFs and the probabilities of their contributions to the risk classes				
		Workload	Fumes (PM2.5)	Fumes (PM10)	Noise	Entanglement
t1	Safe	99.2%	96.2%	100%	84.1%	15.9%
t2	Safe	98.4%	92.2%	100%	53.8%	7.8%
t3	Safe	97.5%	84.1%	100%	45.1%	2.5%
–	–	–	–	–	–	–
t8	Low risk	97.5%	100%	0.5%	44.7%	97.5%
t9	Low risk	92.2%	97.5%	0.2%	15.9%	92.2%
–	–	–	–	–	–	–
t13	Medium risk	97.5%	100%	100%	53.8%	97.5%
t14	Medium risk	92.2%	97.5%	97.5%	44.7%	84.1%
–	–	–	–	–	–	–
t18	High risk	97.5%	84.1%	100%	45.1%	2.5%
–	–	–	–	–	–	–

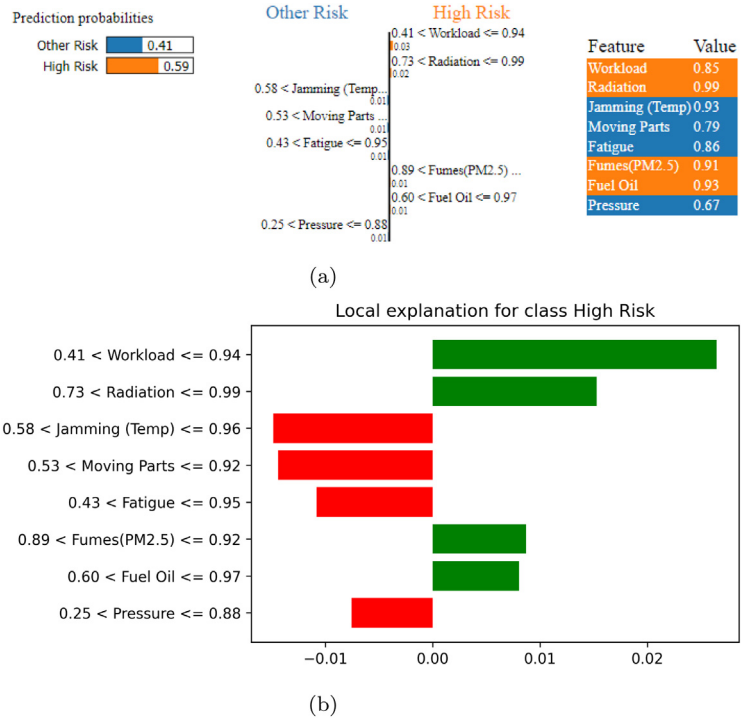


Fig. 10. (a–b) LIME outputs (a) details with probabilities (b) with bar.

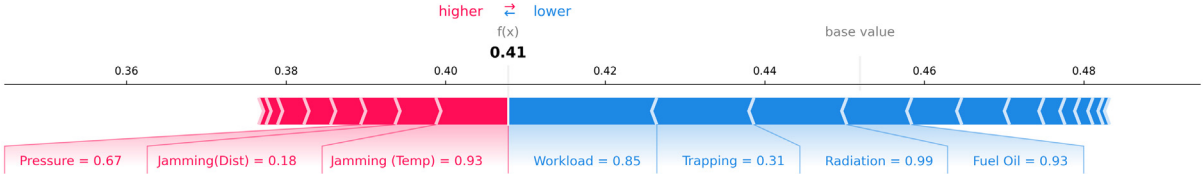


Fig. 11. SHAP output associated with class High risk.

5.4. Comparison of AIAI-PRM's output with that of LIME and SHAP

This section discusses the main differences between AIAI-PRM's output and that of LIME and SHAP. While for a time slot, AIAI-PRM predicts the same output risk class as given by SHAP and LIME, the contributing features for them are not the same. This is because, as previously mentioned, both LIME and SHAP do not model how a feature's value changes over a period of time. While we address this by inputting the numeric values of all the 23 features (which we derived using KG) at the beginning of a time slot, AIAI-PRM then determines the impact of each feature on an output risk class by considering the values of features that are either dependent on it or on which it is impacting. SHAP and LIME do not do this automatically, therefore they determine the effect of a feature on an output class without considering its evolving interdependence on other features. This means that in a complex, dynamic and interdependent system, while the features recommended by SHAP and LIME in a time slot may be the most obvious, they may not be the most significant ones. AIAI-PRM addresses this and thus the main advantages of AIAI-PRM's output over these algorithms in providing an interpretable output can be summarised as follows:

1. **Considers the cumulative interdependence among features across the time slots:** Risk evolves over time. So, addition to considering the importance of individual features in a time slot, AIAI-PRM models the conditional interdependency among the features and how they evolve over the time slots to cumulatively affect the output risk classes. This characteristic of AIAI-PRM is significant because independent features after the first time slot of the time space are no longer independent but interconnected. So, AIAI-PRM considers that the local explanations should not be computed in terms of individual feature weights but rather as how features interact with each other over a period of time. LIME computes the weight of an individual feature that contributes positively or negatively to a given prediction (Visani et al., 2020). In contrast, SHAP computes the Shapley values of each feature that shows its contribution to the prediction (Yeung et al., 2020). However, both LIME and SHAP do not model the cumulative contribution of features that lead to the occurrence of an output risk class, which AIAI-PRM does.
2. **Shows the confidence in features being recommended as the ones contributing to an output risk class:** AIAI-PRM uses a specific rule of LINDA-BN in recommending the contributing features to an output risk class to the decision-maker. This rule compares the features being recommended with those of another classifier and shows them only if they match with each other to avoid misclassification. This, in turn, represents AIAI-PRM's confidence in the correct classification of features being recommended as the ones contributing to an output risk class. These rules are not easily represented in LIME and SHAP (Offert and Bell, 2020; Riis et al., 2021) and thus they do not represent the confidence in the features which they show are contributing to an output risk class.
3. **Represents causal relationship between the KFs and SFs to the output risk classes:** AIAI-PRM uses KG to represent the direct and indirect links between the different features being modelled. Using the level of impact which each feature will have, it generates a local probabilistic graphical model in each time slot. This shows the causal analysis, i.e., an experimental design and statistics to establish the cause and effect relationship among the considered features to represent their impact on the output risk classes (Hall et al., 2019). The approaches provide the decision-maker with a causal understanding of why a particular prediction was calculated, which existing approaches do not do.

5.5. Limitations of AIAI-PRM

While AIAI-PRM assists the decision maker by providing an interpretable output, it has the following limitations:

1. While one of the advantages of AIAI-PRM is that it considers the interdependence among features across the time slots and models their cumulative inference on the output risk class, it also leads to the disadvantage of increasing the complexity of modelling after some time slots.
2. As is the case with LIME and SHAP (ElShawi et al., 2020), the explainability nature of AIAI-PRM diminishes with an increase in the number of features. Furthermore, AIAI-PRM learns a Bayesian network structure from a complete data set. This is an NP-hard problem which is complex and therefore time-consuming.
3. AIAI-PRM uses an expert's prior knowledge and the company's goal to augment the relationship between the features in the KG and their impact on the output class. As we do not range test the impact of these features on the output classes, in a complex environment, this understanding of the inputs may depend on different evaluation criteria and vary according to the expert viewpoints, leading to a change in the output.

6. Conclusion

In this paper, we introduced AIAI-PRM which is a framework for interpreting the antecedent features that lead to a risk-based prediction. Our proposed framework augments LINDA-BN with knowledge graph and system dynamics to capture the interdependent nature of features and utilise these to model their level of contribution to an output risk class. The interpretations provided by AIAI-PRM only correspond to Rule 1 of LINDA-BN which shows a high confidence in it with other classifiers, thereby increasing its trustworthiness. Compared to LIME and SHAP, AIAI-PRM improves the interpretability aspect by using the chain of links between different features across different time slots, thereby considering a system rather than a silo-based approach among the features. This will assist the risk manager to identify the root cause of risk occurring and taking proactive steps to address this rather than addressing the symptom. Our future work will focus to explain how the contributing features leading to the occurrence of an output risk class evolve over a period of time to explain the logic behind their occurrence. Another action item for future work is to extend the evaluation of AIAI-PRM to real-world, more complex data sets. Sindhgatta et al. (2020) propose doing this on an organisation's data such as event logs etc.

CRediT authorship contribution statement

Sonia Farhana Nimmy: Co-conceptualization of the idea, Co-development of the idea, Data collection, Coding, Experiments, Results analysis, Initial draft version of the manuscript. **Omar K. Hussain:** Co-conceptualization of the idea, Co-development of the idea, Supervision of the project, Finalising the manuscript. **Ripon K. Chakrabortty:** Co-conceptualization of the idea, Co-development of the idea, Co-supervision of the project, Finalising the manuscript. **Farookh Khadeer Hussain:** Co-conceptualization of the idea, Review & editing, Feedback towards finalization. **Morteza Saberi:** Co-conceptualization of the idea, Review & editing, Feedback towards finalization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

Sonia Farhana Nimmy acknowledges the financial support from The University of New South Wales, Canberra, Australia to support her time on his project.

References

Angelov, P.P., Soares, E.A., Jiang, R., Arnold, N.I., Atkinson, P.M., 2021. Explainable artificial intelligence: an analytical review. Wiley Interdisc. Rev. Data Min. Knowl. Discov. 11 (5), e1424.

Anon, 2022. Interpretable time-series. https://github.com/sicara/tf-explain (Accessed: 21 Sept 2022).

Ates, E., Aksar, B., Leung, V.J., Coskun, A.K., 2021. Counterfactual explanations for multivariate time series. In: 2021 International Conference on Applied Artificial Intelligence. ICAPAI, IEEE, pp. 1–8.

Bellucci, M., Delestre, N., Malandain, N., Zanni-Merk, C., 2021. Towards a terminology for a fully contextualized XAI. Procedia Comput. Sci. 192, 241–250.

Biecek, P., Burzykowski, T., 2021. Explanatory Model Analysis: Explore, Explain, and Examine Predictive Models. CRC Press.

Burkart, N., Huber, M.F., 2021. A survey on the explainability of supervised machine learning. J. Artificial Intelligence Res. 70, 245–317.

Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J., 2020. Explainable AI in fintech risk management. Front. Artif. Intell. 3, 26.
Cali, U., Kuzlu, M., Pipattanasomporn, M., Kempf, J., Bai, L., 2021. Foundations of big data, machine learning, and artificial intelligence and explainable artificial intelligence. In: Digitalization of Power Markets and Systems using Energy Informatics. Springer, pp. 115–137.
Chen, X., Jia, S., Xiang, Y., 2020. A review: Knowledge reasoning over knowledge graph. Expert Syst. Appl. 141, 112948.
Chen, J., Koju, W., Xu, S., Liu, Z., 2021. Sales forecasting using deep neural network and SHAP techniques. In: 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering. ICBAIE, IEEE, pp. 135–138.
Delaney, E., Greene, D., Keane, M.T., 2021. Instance-based counterfactual explanations for time series classification. In: International Conference on Case-Based Reasoning. Springer, pp. 32–47.
Dhanorkar, S., Wolf, C.T., Qian, K., Xu, A., Popa, L., Li, Y., 2021. Who needs to know what, when?: Broadening the explainable AI (XAI) design space by looking at explanations across the AI lifecycle. In: Designing Interactive Systems Conference 2021. pp. 1591–1602.
Došilović, F.K., Brčić, M., Hlupić, N., 2018. Explainable artificial intelligence: A survey. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics. MIPRO, IEEE, pp. 0210–0215.
ElShawi, R., Sherif, Y., Al-Mallah, M., Sakr, S., 2020. Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. Comput. Intell..
Fidel, G., Bitton, R., Shabtai, A., 2020. When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures. In: 2020 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–8.
Gabbay, D., Woods, J., 2006. Advice on abductive logic. Logic J. IGPL 14 (2), 189–219.
Gejke, C., 2018. A new season in the risk landscape: Connecting the advancement in technology with changes in customer behaviour to enhance the way risk is measured and managed. J. Risk Manag. Financial Inst. 11 (2), 148–155.
Giudici, P., Raffinetti, E., 2021. Shapley-Lorenz explainable artificial intelligence. Expert Syst. Appl. 167, 114104.
Guillemé, M., Masson, V., Rozé, L., Termier, A., 2019. Agnostic local explanation for time series classification. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence. ICTAI, IEEE, pp. 432–439.
Hall, M., Harborne, D., Tomsett, R., Galetic, V., Quintana-Amate, S., Nottle, A., Preece, A., 2019. A systematic method to understand requirements for explainable AI (XAI) systems. In: Proceedings of the IJCAI Workshop on EXplainable Artificial Intelligence, Vol. 11. XAI 2019, Macau, China.
Heckerman, D., 2008. A tutorial on learning with Bayesian networks. Innov. Bayesian Netw. 33–82.
Heckerman, D., Geiger, D., Chickering, D.M., 1995. Learning Bayesian networks: The combination of knowledge and statistical data. Mach. Learn. 20 (3), 197–243.
Holzinger, A., Kieseberg, P., Weippl, E., Tjoa, A.M., 2018. Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable AI. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer, pp. 1–8.
Ismail, A.A., Gunady, M., Corrada Bravo, H., Feizi, S., 2020. Benchmarking deep learning interpretability in time series predictions. Adv. Neural Inf. Process. Syst. 33, 6441–6452.
Jia, Y., Qi, Y., Shang, H., Jiang, R., Li, A., 2018. A practical approach to constructing a knowledge graph for cybersecurity. Engineering 4 (1), 53–60.
Karci, A., 2020. Finding innovative and efficient solutions to NP-hard and NP-complete problems in graph theory. Bilgisayar Bilimleri 5 (2), 137–143.
Kenny, E.M., Keane, M.T., 2021. On generating plausible counterfactual and semi-factual explanations for deep learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. (13), pp. 11575–11585.
Klaise, J., Van Looveren, A., Vacanti, G., Coca, A., 2021. Alibi explain: Algorithms for explaining machine learning models. J. Mach. Learn. Res. 22, 1–181.
Ko, H., Witherell, P., Lu, Y., Kim, S., Rosen, D.W., 2021. Machine learning and knowledge graph based design rule construction for additive manufacturing. Addit. Manuf. 37, 101620.
Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., et al., 2020. Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896.
Koller, D., Friedman, N., 2009. Probabilistic Graphical Models: Principles and Techniques. MIT Press.
Lakkaraju, H., Kamar, E., Caruana, R., Leskovec, J., 2019. Faithful and customizable explanations of black box models. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. pp. 131–138.
Lambert, G.F., Lasserre, A.A.A., Ackerman, M.M., Sánchez, C.G.M., Rivera, B.O.I., Azzaro-Pantel, C., 2014. An expert system for predicting orchard yield and fruit quality and its impact on the Persian lime supply chain. Eng. Appl. Artif. Intell. 33, 21–30.
Li, H., Fan, W., Shi, S., Chou, Q., 2019. A modified LIME and its application to explain service supply chain forecasting. In: CCF International Conference on Natural Language Processing and Chinese Computing. Springer, pp. 637–644.
Liao, Q.V., Gruen, D., Miller, S., 2020. Questioning the AI: informing design practices for explainable AI user experiences. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–15.
Lin, C.-H., Azabou, M., Dyer, E.L., 2021. Making transport more robust and interpretable by moving data through a small number of anchor points. Proc. Mach. Learn. Res. 139, 6631.
Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. 30.
Luo, W., Zhang, G., Tran, H.D., Setunge, S., Hou, L., 2019. Implementing proactive building asset management through deterioration prediction: A case study in Australia. In: International Symposium on Advancement of Construction Management and Real Estate. Springer, pp. 951–965.
Luss, R., Chen, P.-Y., Dhurandhar, A., Sattigeri, P., Zhang, Y., Shanmugam, K., Tu, C.-C., 2021. Leveraging latent features for local explanations. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1139–1149.
Malgieri, G., 2019. Automated decision-making in the EU member states: The right to explanation and other "suitable safeguards" in the national legislations. Comput. Law Secur. Rev. 35 (5), 105327.
Marcot, B.G., 2017. Common quandaries and their practical solutions in Bayesian network modeling. Ecol. Model. 358, 1–9.
Matin, S.S., Pradhan, B., 2021. Earthquake-induced building-damage mapping using explainable AI (XAI). Sensors 21 (13), 4489.
Mohamed, A., Abuoda, G., Ghanem, A., Kaoudi, Z., Aboulnaga, A., 2021. Rdfframes: Knowledge graph access for machine learning tools. VLDB J. 1–26.
Molnar, C., 2020. Interpretable Machine Learning. Lulu. com.
Moreira, C., Chou, Y.-L., Velmurugan, M., Ouyang, C., Sindhgatta, R., Bruza, P., 2021. LINDA-BN: An interpretable probabilistic approach for demystifying black-box predictive models. Decis. Support Syst. 113561.
Mothilal, R.K., Sharma, A., Tan, C., 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 607–617.
Offert, F., Bell, P., 2020. Perceptual bias and technical metapictures: critical machine vision as a humanities challenge. AI & Soc. 1–12.
Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B.R., Girvan, M., Ott, E., 2018. Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. Chaos 28 (4), 041101.
Riis, C., Kowalczyk, D.K., Hansen, L.K., 2021. On the limits to multi-modal popularity prediction on instagram-a new robust, efficient and explainable baseline. In: 13th International Conference on Agents and Artificial Intelligence. SCITEPRESS Digital Library, pp. 1200–1209.
Rothman, D., 2020. Hands-on Explainable AI (XAI) with Python: Interpret, Visualize, Explain, and Integrate Reliable AI for Fair, Secure, and Trustworthy AI Apps. Packt Publishing Ltd.
Scutari, M., Vitolo, C., Tucker, A., 2019. Learning Bayesian networks from big data with greedy search: computational complexity and efficient implementation. Stat. Comput. 29 (5), 1095–1108.
Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626.
Siddiqui, S.A., Mercier, D., Dengel, A., Ahmed, S., 2021. TSInsight: A local-global attribution framework for interpretability in time series data. Sensors 21, (21), 7373.
Siddiqui, S.A., Mercier, D., Munir, M., Dengel, A., Ahmed, S., 2019. Tsviz: Demystification of deep learning models for time-series analysis. IEEE Access 7, 67027–67040.
Sindhgatta, R., Ouyang, C., Moreira, C., 2020. Exploring interpretability for predictive process analytics. In: International Conference on Service-Oriented Computing. Springer, pp. 439–447.
Slack, D., Hilgard, A., Singh, S., Lakkaraju, H., 2021. Reliable post hoc explanations: Modeling uncertainty in explainability. Adv. Neural Inf. Process. Syst. 34, 9391–9404.
Smith, P.G., Merritt, G.M., 2020. Proactive Risk Management: Controlling Uncertainty in Product Development. productivity Press.
Taheri, S., Mammadov, M., Bagirov, A.M., 2011. Improving naive Bayes classifier using conditional probabilities. In: AusDM. pp. 63–68.
Tiddi, I., Lécué, F., Hitzler, P., 2020. Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges, Vol. 47. IOS Press.
Ventura, F., Cerquitelli, T., Giacalone, F., 2018. Black-box model explained through an assessment of its interpretable features. In: European Conference on Advances in Databases and Information Systems. Springer, pp. 138–149.
Visani, G., Bagli, E., Chesani, F., Poluzzi, A., Capuzzo, D., 2020. Statistical stability indices for LIME: obtaining reliable explanations for machine learning models. J. Oper. Res. Soc. 1–11.
Yeung, C., Tsai, J.-M., King, B., Kawagoe, Y., Ho, D., Knight, M.W., Raman, A.P., 2020. Elucidating the behavior of nanophotonic structures through explainable machine learning algorithms. ACS Photonics 7 (8), 2309–2318.