IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 29, 2019, accepted September 12, 2019, date of publication September 16, 2019, date of current version October 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941566

Safety + AI: A Novel Approach to Update Safety Models Using Artificial Intelligence

YOUCEF GHERAIBIA[1,2], SOHAG KABIR[1], KOOROSH ASLANSEFAT[1], (Member, IEEE), IOANNIS SOROKOS[1], AND YIANNIS PAPADOPOULOS[1]

[1]Department of Computer Science and Technology, University of Hull, Hull HU6 7RX, U.K.
[2]Assuring Autonomy International Programme, University of York, York YO10 5GH, U.K.

Corresponding author: Koorosh Aslansefat (k.aslansefat-2018@hull.ac.uk)

This work was supported by the DEIS H2020 Project under Grant 732242.

ABSTRACT Safety-critical systems are becoming larger and more complex to obtain a higher level of functionality. Hence, modeling and evaluation of these systems can be a difficult and error-prone task. Among existing safety models, Fault Tree Analysis (FTA) is one of the well-known methods in terms of easily understandable graphical structure. This study proposes a novel approach by using Machine Learning (ML) and real-time operational data to learn about the normal behavior of the system. Afterwards, if any abnormal situation arises with reference to the normal behavior model, the approach tries to find the explanation of the abnormality on the fault tree and then share the knowledge with the operator. If the fault tree fails to explain the situation, a number of different recommendations, including the potential repair of the fault tree, are provided based on the nature of the situation. A decision tree is utilized for this purpose. The effectiveness of the proposed approach is shown through a hypothetical example of an Aircraft Fuel Distribution System (AFDS).

INDEX TERMS Fault tree, reliability, safety modeling, model repair, machine learning, artificial intelligence.

I. INTRODUCTION

Safety critical systems are systems for which human life, environmental health, and financial assurance need to be guaranteed. Medical and surgery equipment, aviation and air traffic control, hazardous and toxic chemical processes and nuclear power plant are among safety-critical systems [1]. Different key performance indices such as reliability, safety, availability, and security have been introduced as a measure for the evaluation of safety-critical systems [2]. Reliability of a system can be defined as the probability of its functioning as expected without any malfunctioning or failure during a certain and pre-defined period of time [3], [4]. For the safety attribute, different definitions exist. For instance, the probability of either a system functioning correctly without any fault during the mission time or terminating its service(s) through a safe procedure can be called safety [5]. Another example would be aircraft emergency landing safety. Considering the possibility of an aircraft engine failure, the probability of either reaching the destination without crashing or having a successful emergency landing can be considered as a safety measure for the aircraft.

Because of the criticality of the system functioning, a rigorous reliability and safety evaluation requires comprehensive and certified model(s) that are usually provided by a team of high-level experts. A variety of approaches are developed for modelling and evaluation of dependability attributes, notably reliability and safety. The existing approaches can be classified into four categories; I) state-space modelling such as Continuous-Time Markov Chains (CTMCs), Semi-Markov Processes (SMPs) and Markov Regenerative Process (MRGP) [6], II) Non-State-Space (Combinatorial) Models like Reliability Block Diagram (RBD) and Fault Tree Analysis (FTA) [7], III) Numerical methods like Monte Carlo Simulations, and IV) multi-level models that can be created through a combination of mentioned methods [8], [9]. It should be noted that safety models can also be categorized in terms of qualitative and quantitative analysis.

As fault trees feature easy to understand structure and widespread use, in this paper, we will consider fault trees

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

as an example of safety artefacts. Fault Trees are one of the well-known deductive techniques in which the systems' failures and their combinations can be modeled in a logical and hierarchical manner. A Fault Tree (FT) consists of different levels; top level and top event: usually, in the top level of a Fault Tree, there is a top event representing the failure of the whole system or mission. Intermediate level(s): this level includes the failure of sub-systems. As an example, the failure of an aircraft is a top event and the failure of its sub-systems such as the propulsion system, navigation system, etc. are the intermediate events located at an intermediate level. Basic events: in the FTA, a system can decompose to sub-systems and each sub-system can decompose to sub-sub-systems. This procedure will continue to the level that no more decomposition is affordable or possible. The events in the final decomposition level are called basic events. A failure of a GPS in a navigation system or a short circuit in an electronic board can be considered as examples of basic events. Gates: as mentioned before, the combination of failures in Fault Tree illustrates through logic gates [10].

FTA is widely used in many industries as a mean of providing evidence while assuring safety through certification. The process typically begins with an argument about the safety of a system. For instance, in the automotive industry, such an argument for a braking system could be ''it is guaranteed that the braking system will provide service at ASIL level D''. To support this argument, as a basis of the above guarantee, the analyst may show the result of an FTA. Therefore, the correctness of FTA plays a vital role in the integrity of the of safety certificate. An error in the model (e.g. FT) used for providing evidence can make the safety guarantee void, thus make safety certificate invalid.

A. MOTIVATION AND CONTRIBUTIONS

The correctness of the safety artefacts is very important for providing the right level of safety assurance. An error in these models may lead to a false safety assurance provision. It is important to note that every step of safety artifact construction process heavily relies on the expertise of the analysts. The IET has developed a brand-new set of standards [11] by defining three levels of competency of an analyst such as supervised practitioner, practitioner, and an expert. There also exists a possibility of no established confidence. Under this condition, when the analyst has no or very limited evidence in hand, the developed FT could be inferior, and any safety guarantee provided based on this FT is highly likely to be of very low quality.

In the literature, some researchers have pointed out the flaws in safety artefacts. For instance, Manion [12] critically scrutinized the FTA and correctness of the FT-based safety analysis. He mentioned FTA as a flawed approach based on a series of false assumptions and pointed out what can go wrong in five different steps of FT construction. Moreover, a number of flaws of safety analysis methods and processes have been reported in [13], [14]. One of the limitations is related to the completeness of the hazard identification process of safety analysis. It is not possible to guarantee that all possible hazards are identified during the safety analysis process. Suokas and Pyy [15] and Carter and Smith [16] investigated several incidences in the process industry and construction industry respectively, to find the relation between hazards that were identified during the analysis and hazards that were identified during operation. Based on their investigation, they found significant gaps between these two sets of hazards list.

Fig. 1 shows a typical safety analysis where a group of safety analysts use the system design and safety requirements

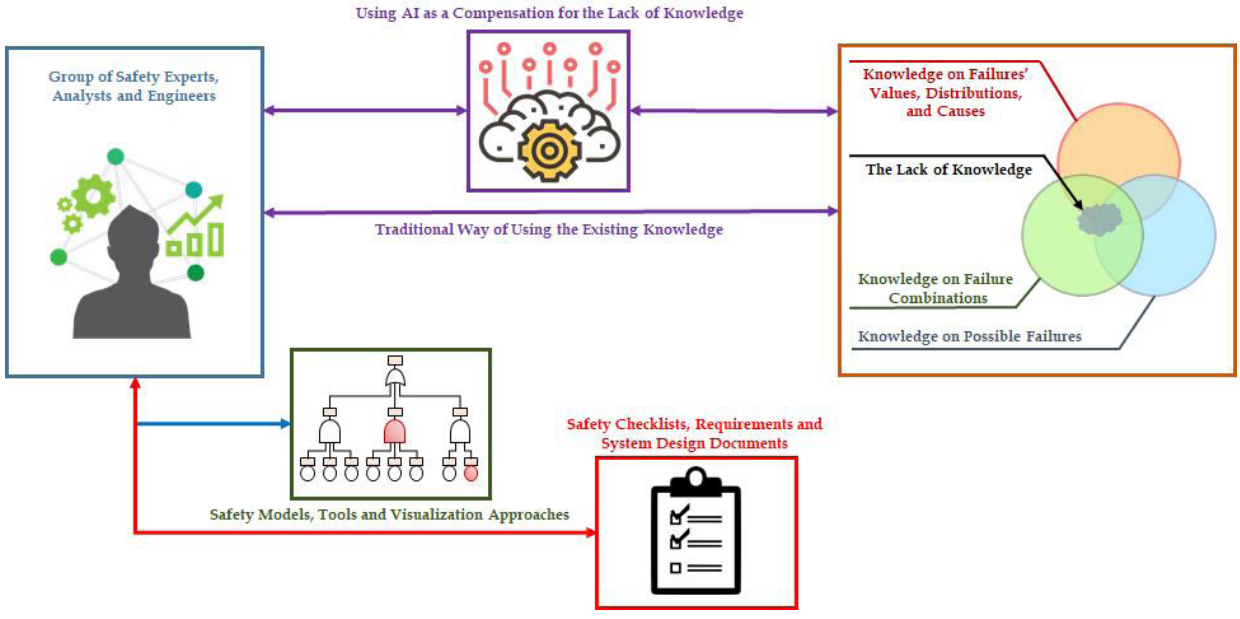

FIGURE 1. A typical safety analysis process.

to identify the possible causes of system failure. However, because of the limited knowledge of the experts, unpredicted causes of failure can exist that are unforeseen, thus not considered in the safety model(s). Generally, the mentioned issue can occur and cause catastrophes when an unpredicted failure event with low probability and high impact happens. The Fukushima Daiichi nuclear disaster [17] initiated by the tsunami following an earthquake on 11 March 2011 is an example of a case where the designer of the nuclear facility failed to foresee the environmental circumstances that may cause the system failure. The statement "We can only work on precedent, and there was no precedent. When I headed the plant, the thought of a tsunami never crossed my mind" [18] given by Tsuneo Futami, a former director of Fukushima Daiichi plant, makes it clear that sometimes it is not possible to foresee all possible failure modes, especially if the failure modes represent infrequent events. However, such events are often discovered during the operation of the system.

As argued in [19], a system can have behaviours which are not non-conformant to the specifications but are still unsafe. These kinds of situations are usually not captured by safety artefacts because of the nature of the behavior of the system. For instance, according to Leveson [20], the Mars Polar Lander accident and Ariane 5 launch failure are two industrial incidences where the system components operated exactly as they were planned to work, however, the systems still failed because of the wrong perception of the effects of their behavior. Therefore, it is obvious that the safety model of the systems where the behavior of the components is wrongly perceived is bound to be incorrect. If we consider this issue from the point of view of creating a FT, then this may lead to inserting wrong logic in the FT, i.e., using a wrong gate to model the relationship between events. Moreover, analysts may make such mistakes by not following systematic ways of creating a FT. For instance, consider that the failure behavior of a system is depicted by the FT of Fig. 2. The outcomes of qualitative and quantitative analysis of this FT depend on the logical structure of the tree. Changing the type of a gate, i.e., the logical relationship between events may significantly change the safety and reliability of the system. If we replace one of the AND gates by an OR gate, then we will achieve a completely new failure behavior of the system. Consider the AND gate in red color, which says that the system will fail if all BE4, BE5, and BE6 occur. However, if we replace this gate with an OR gate, then this will mean that the system will fail if any of the above-mentioned BE occurs.

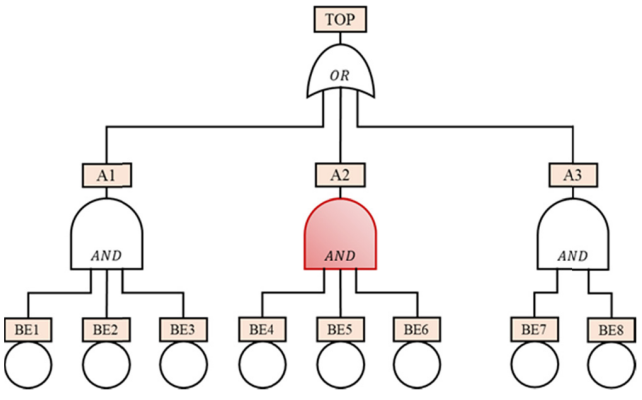

FIGURE 2. An example FT.

In addition to the above-mentioned issues, we can have invalid assumptions, such as statistical independence of basic events. This can also produce misleading results. It is important to note that, not all the accidents are caused due to a problem with safety analysis, but in many cases, improper actions taken by system operators cause the accident. A survey on aircraft accidents conducted by Lloyd and Tye [21] suggested that almost 50% of those accidents were caused due to the improper actions of crews. For instance, in the Kegworth air disaster on January 1989, there was a delay in alerting the pilot about the occurrence of the fault and its causes. As a result, the pilots made a misjudgement and took the wrong action, causing the accident. In another fatal accident caused partially due to a misleading alarm annunciation, the Airbus A330 of the Air France flight AF447 crashed in the Atlantic, killing all 228 people on board. As the information about the blocked pitot tube and the information about the angle of attack reading were not properly conveyed to the crews, they were not able to take the appropriate actions. Moreover, there were no clear guidelines available of the crews regarding those particular emergency situations [22]. From these two cases, we can see that the lack of knowledge about some particular events and delay in the alarm communication contributed significantly to the occurrence of the accidents. Therefore, timely communication of alarms and more knowledge (information) might help the users of a system to take more informed and appropriate actions during an unforeseen scenario.

To address the above-mentioned issues related to unforeseen events and misunderstanding about system behavior, in this paper, we propose a data-driven approach using machine learning to provide assistance when an unknown or unconsidered scenario encountered during system operation. The aim of our approach is to crosscheck the real time operational behavior of the system with the safety artefacts (in this case fault tree) created for the system during design time to see if the current operational scenario is explained by the FT of the system. To achieve this, we used a machine learning based method for forming the normal behavior model of a system based on operational data. Afterward, we provide the process of identifying anomalies in the behavior of the system at any time instance during system operation by cross checking with the data-driven nominal behavior model of the system formed earlier. Finally, a decision-making system is provided for verifying whether an abnormal behavior detected based on the operational data is explained by the fault tree of the system. If the explanation is not found in the existing FT, recommendation about potential update of the safety artifact of the system is provided.

II. BACKGROUND AND RELATED WORKS

In this section, the background has been divided into two subsection; I) A brief literature survey on model repair and II) Machine learning approaches associated with safety models.

A. MODEL REPAIR AND PROCESS MINING

Process mining enables analysts to extract insights from log data and create a performance model of an industrial plant. It is also possible to repair or update existing models through data mining [23]. Regarding this area, substantial research has been done and a brief literature review will be provided in what follows.

An approach based on the alpha algorithm has been proposed in [24] to generate a Petri net model of the process from its workflow log. However, the inability to extract a model from processes with arbitrary workflow was a limitation of the proposed method. A hierarchical and iterative process-mining technique has been introduced in [25] to refine the process model. The paper constructed the original system model and verified this model based on the incoming data. At each iteration, the steady-state behavior of the model is checked regarding any changes in data. van der Aalst *et al.* [26] emphasized the importance of aligning the system model and the workflow log file, which represents data from the system deployment. These relationships have been used to check the conformity of the system and to evaluate the performance of the system as well. From a computational point of view, these alignments are the challenging problem, which is an open challenge to find the optimal alignment.

Authors in [27] identified all possible variants of the system model, by applying a clustering algorithm from data of the logs file (execution traces). The paper proposed a greedy method to make an exhaustive search of possible behaviours, and at each step try to cluster traces sharing similar behaviours. Augusto *et al.* [28] presented a literature review of automated process discovery methods, and the outcome of the paper revealed that some methods suffer from the lack of scalability and inconsistent performance results.

B. MACHINE LEARNING ASSOCIATED WITH SAFETY MODELS

Safety models are one of the most important pillars in reliability and safety assessment that can represent the failure combinations of a system, and they need to be created by certified designers and experts. In the following, research works that combined Fault Tree Analysis (FTA) with Machine learning approaches will be discussed.

Hurdle *et al.* [29] used a non-coherent Fault Tree for the fault diagnosis of a water tank system. The limitations in this method was a need for consistency checks from observation points. Two years later, the approach has been updated by combining the FTA and Bayesian Belief networks in [30]. Cai *et al.* [31] proposed a new method for real-time reliability analysis through a combination of traditional Bayesian networks derived from root cause diagnosis and dynamic Bayesian networks. In fact, this study updates prior reliability knowledge of the system (failure distributions) via dynamic Bayesian networks. A subsea pipe ram BOP system has been addressed as a case study in this paper.

Askarian *et al.* [32] proposed a new method for fault diagnosis through a fusion of micro-macro data. In this paper, the FTA and Bayesian networks have been combined to gain the advantages of both prior probability distribution in FTA and real-time data in Bayesian networks. Remaining Useful Life (RUL) is a parameter usually estimated through Machine learning approaches [33]. A method for combining failure rate and RUL as the basic event in Dynamic Fault Tree has been proposed in [34], [35].

A hierarchical Bayesian network-based model has been provided for process monitoring and decision making in [36]. This article used a data-driven algorithm to update the sub-Bayesian networks in the model. Getir *et al.* [37] focused on semi-automated and co-evaluated process as a case study and defined a number of intra- and inter-model rules of transformation to cover the evaluation scenarios. The outcome of this study has shown that realizing the co-evolution of the proposed approach required fewer user interactions. The potential challenges and opportunities of using machine learning in a safety-critical application have been reviewed in [38]. The paper illustrated how missing casualties in the model can be reduced through the incorporation of safety models and data-driven knowledge.

A conceptual idea regarding the combination of artificial intelligence methods with safety models has been presented in [39]. In this report, examples of golf-shot on the moon and Falcon launch from SpaceX have been demonstrated. Cheng *et al.* [40] proposed an Imitation Medical Diagnosis Method (IMDM) in which three types of Bayesian networks have been used; Machine Learning BN, Expert empirical BN, and maintenance decision BN. The method also applied the fuzzy theorem to achieve uncertainties and conditional probabilities.

From the above discussion, all the works related to model repair and/or machine learning for safety analysis consider the system models and based on the conformity analysis or model checking different potential actions such as to repair a model, update a model, upgrade a model, etc. are recommended. However, to the best of the authors' knowledge, no work has considered the same for the safety artefacts that are used to safety certification of the system models. Therefore, in this paper, we aim to combine machine learning with safety analysis for runtime evaluation of correctness of safety models developed during the design time.

III. THE PROPOSED APPROACH

The approach proposed in this paper considers that at design time, safety analysts have knowledge about the system model and behavior, and they have already created a fault tree of the system based on their knowledge about the foreseeable failure events. It also considers that at runtime the system is

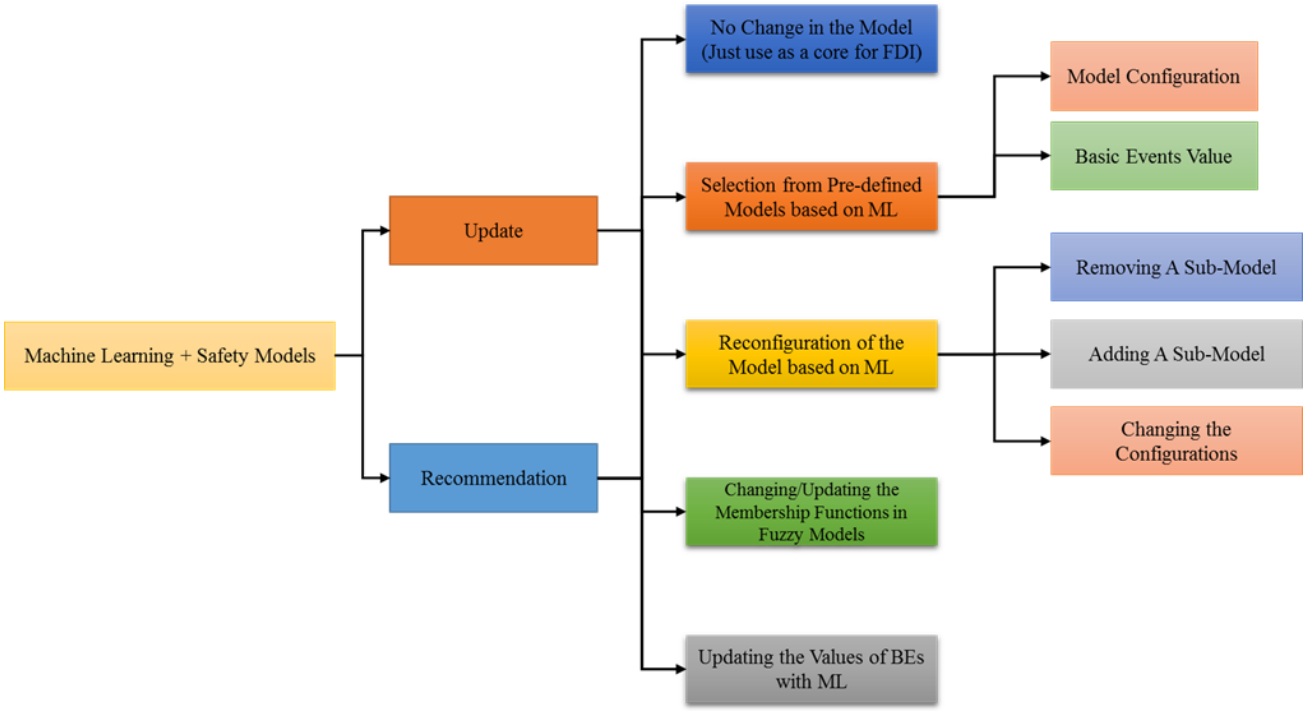

FIGURE 3. A classification for safety models associated with machine learning.

continuously monitored for some parameters, i.e., operational data is available. The basic idea of the approach is to use the real time operational data of the system to learn the normal behavior of the system. Afterwards, when a new set of operational data is available, the knowledge about the normal behavior of the system is used to see if there exists any anomaly in the new record. If any anomaly is detected in the behavior, then the existing system fault tree is consulted to see if it can explain the reason for abnormal behavior, i.e., if the FT contains a node that is associated with this current event. If no explanation is found in the fault tree, different recommendations are provided based on the perceived severity of a scenario. The framework of the proposed approach is shown in Fig. 4. It can be seen that the approach is divided into two parts: the anomaly detection (AD) part and the decision making (DM) part. The AD part is responsible for formulating the normal behavior model of the system and for checking for anomaly in the newly arrived record. We used One Class Support Vector Machine (OC-SVM) to accomplish this task. If an abnormal behavior is detected, the DM part processes the information made available by the machine learning part to suggest appropriate actions. A detailed description of the AD and DM parts of the approach is provided in the next two sections.

A. ANOMALY DETECTION

Anomaly detection is the process of abnormal behavior identification. The abnormal behavior can be an item, object, observation or any unusual pattern from the expected behavior [41]. For example, in the banking sector, to check if a bank card has been stolen or an account is hacked, we can identify any abnormal transaction made from the bank account by generating the normal behavior of the account holder from his/her previous transactions and comparing it with the new transaction. Fig. 5 graphically represents the anomaly detection process, where the blue region (dots) represents datasets forming the expected behavior of a system and the red circle represents an anomalous dataset. In this illustrative example, we showed the behavior formed based on three arbitrary parameters such as FTV, Tmp, and PFT. Each point in the graph represents an observation on these three parameters at different point in time.

Based on the testing data, the anomaly detection problem can be formulated as a supervised, unsupervised or semi-supervised classification problem. In supervised anomaly detection, the data set is labeled as *'normal'* or *'abnormal'*, and the algorithm will learn the model of each class and provide a separator for the classification engine. The data will help the algorithm to identify the importance of each feature to the problem in hand. In contrast, the unsupervised anomaly detection techniques use unlabeled test data. Therefore, there is no straightforward technique to evaluate the quality of the results. Semi-Supervised learning anomaly detection uses a mixture of labeled and unlabeled data for the learning phase. In most anomaly detection problems, the data is unlabeled with the assumption that the system is by default most likely to perform its normal expected behavior.

In order to detect anomalies in the system behavior, the first step is to generate the normal behavior of the system after a dataset is received from the real-time operation of the system.

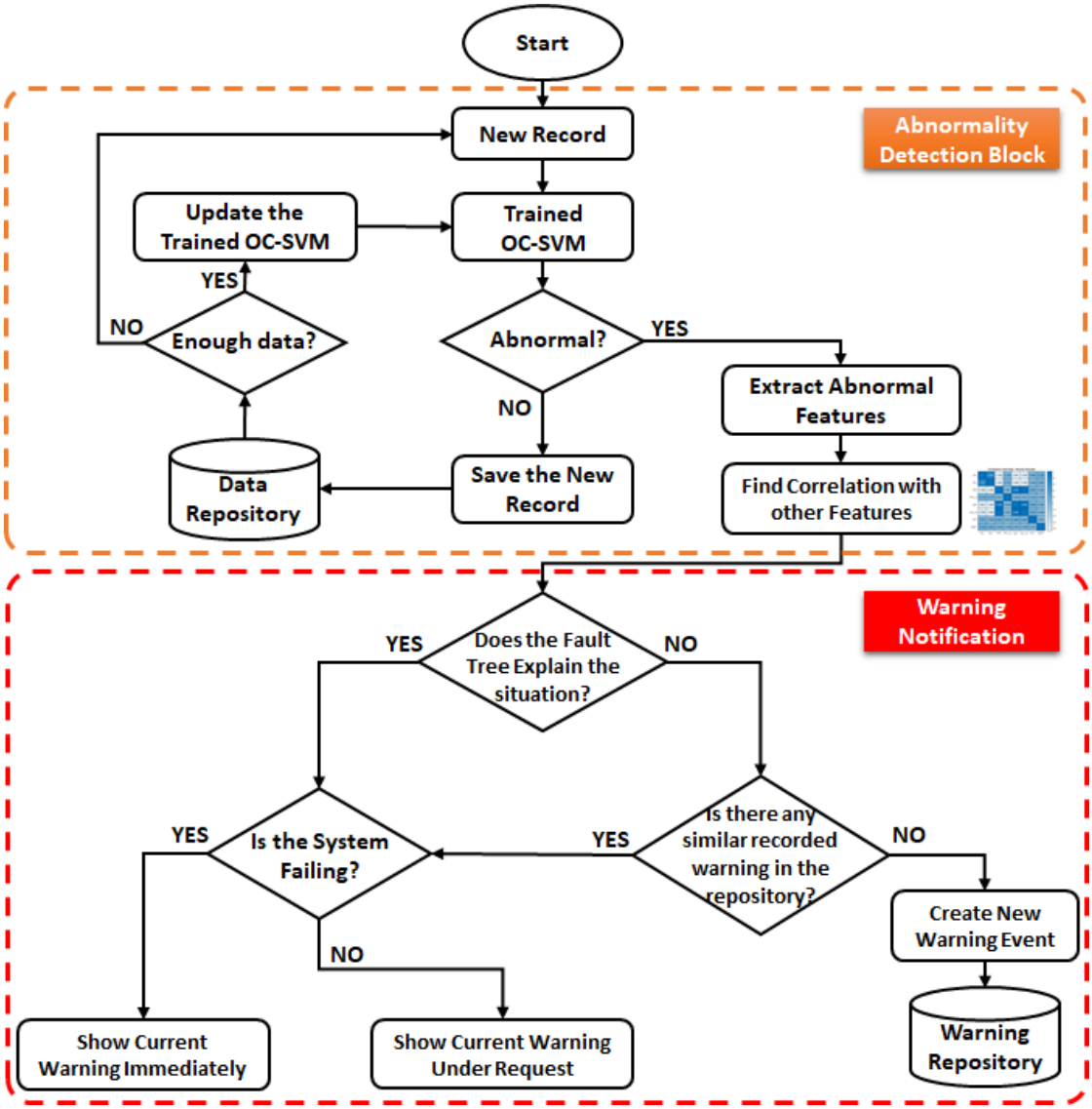

FIGURE 4. Framework of the proposed approach for abnormality detection and warning notification.

This means this step should wait for a sufficient number of records to be available. The incoming data from the system is considered to be normal, i.e., all data are labeled with one class. Regarding that, the normal behavior generation problem is formulated as a one-class classification problem using only data from the assigned class. The One Class Support Vector Machine (OC-SVM) classifier [40] has been used to generate normal behavior. OC-SVM uses a pattern analysis algorithm to study the general type of relations among the instances of data. This type of algorithm is known as a 'Kernel', which represents a similarity measure between any two inputs, also known as a weighting factor. There are different types of kernel functions such as linear, Gaussian, polynomial, and hyperbolic tangent. A kernel function is used inside the decision function. Usually, the selection of a particular type of kernel is problem-specific and contributes considerably to the success of the learning algorithm. SVM uses the kernel to map the data into the feature space H and tries to converge the data points into a hypersphere in feature space.

The OC-SVM can be formulated as follows: Let $X = \{x_1, x_2, x_3, \ldots, x_n\}^m$ be a set of instances with one label *'Normal'* representing the streaming data coming continuously in real time from the system. n is the number of all parameters of the system (data captured from the different system's sensors) and m is the number of instances at a time instant t.

Let $K : R^n \rightarrow H$ be the kernel function that transforms the input data to the features space H. The OC-SVM is in general an optimization problem, which tries to minimize the distance between points on the same class and maximize the distance between points inside the class and

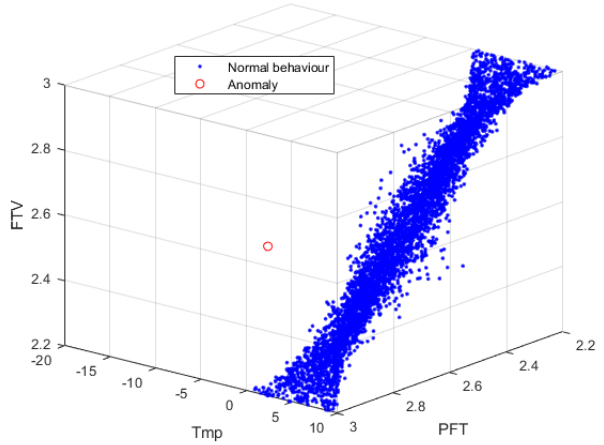

FIGURE 5. Graphical representation of anomaly detection process.

the origin [42].

$$Min\ F(\omega, b, \delta, \rho)^n = \frac{1}{2}\|\omega\|^2 + \frac{1}{v_n}\sum_{i=1}^{n}\delta_i - \rho$$
$$Subject\ to: \left(\omega^T K(x_i)\right) \geq \rho - \delta_i \quad \text{for } \delta_i \geq 0,$$
$$\text{for all i} = 1, \ldots, \text{n}, \ \text{v} \in (0, 1] \qquad (1)$$

where δ_i is the relaxation parameter, which is used to balance the experienced risk minimisation. ω, b parameters are used for deciding the separating line (hyperplane), which defines the decision distance that separates points assigned to the normal behavior to other points.

v_n sets upper bound of the out-of-class training examples and lower bound on the number of training used as support vector. n is the number of points in the training dataset.

The problem of finding the optimal hyperplane, which makes separation between classes of data, is a quadratic problem. The main objective of the quadratic problem is to find the optimal separating hyperplane between classes. A general quadratic programming problem can be described as:

$$\text{Min } Q(\alpha) = \frac{1}{2}\sum_{i,j}^{n}\alpha_i\alpha_j K\left(x_i, x_j\right)$$
$$\text{Subject to: } 0 \leq \alpha_i \leq \frac{1}{v_n}, \quad \sum_{i}^{n}\alpha_i = 1 \qquad (2)$$

where α_i is the influence of example $\boldsymbol{i}$.

$$f(x) = sign\left((\omega, K(x)) - \rho\right) \qquad (3)$$

where *sign* function is the derivative of the absolute value function $(-1, +1)$.

$$\rho = \sum_{j=0}^{n}\alpha_i K\left(x_i, x_j\right) \qquad (4)$$

The kernel is a positive symmetric function where it projects input vectors into a feature space allowing for non-linear decision boundaries. Let φ denote the feature mapping, which maps from the attributes to the features. The kernel uses a feature mapping φ, which maps the data to a new space. The construction of the mapping function is very expensive in very high dimensional spaces.

$$\emptyset : \text{X} \rightarrow \text{R}^{\text{N}}, \quad K\left(x_i, x_j\right) = \emptyset\left(x_i\right)^T \cdot \emptyset\left(x_j\right)$$
$$\emptyset\left(x_i\right) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} \qquad (5)$$

After the initial normal behavior model is formed, whenever a new monitoring dataset is received from the sensors of system, this new record is checked against the normal behavior model of the system to detect anomalies. If no anomaly is detected, the record is saved in a central repository. Note that, to keep the normal behavior model of the system updated, it is regenerated after a certain number of new normal records have arrived. This number can be defined by the user. In this paper, we regenerated the model after receiving 10 normal records.

If an anomaly is detected in the new record, correlations are generated among the different parameters within the record. Correlation is a measure of change between two variables. Correlation between variables does not necessarily mean causality, but this measurement is used to provide extra knowledge for the decision-making process. Two variables X and Y are highly correlated if any variation (positive or negative) in X corresponds (or does not correspond) to a similar variation in Y, and vice versa. The correlation between two variables can be interpreted as one of the variables influenced by the other one, or both of them being influenced by a third variable. The interpretation of the correlation can be used as a parameter to the overall decision-making process.

One of the well-known correlation measures is the correlation coefficient (Pearson r). The advantage of this measure is that it is sensitive to outliers. Therefore, a high correlation means the two variables match each other with high probability over an observation period. The formula for computing the Pearson r is as follows:

$$r_{XY} = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{Y}\right)^2}} \qquad (6)$$

The correlation among different variables are utilised to identify the system parameters that have pushed the system outside the boundary of its normal behavior. This information is used in the decision-making process as follows for recommending actions.

B. DECISION-MAKING PROCESS

The result of the anomaly detection part is a set of parameters; these parameters are divided into categories. Some parameters are pushing the behavior of the system to the abnormal region, and other parameters that are highly correlated to the first category's parameters. The decision-making process will take this relevant information as input. Based on this information, several alternative decision paths are identified (see Fig. 6). The decision process will use some external

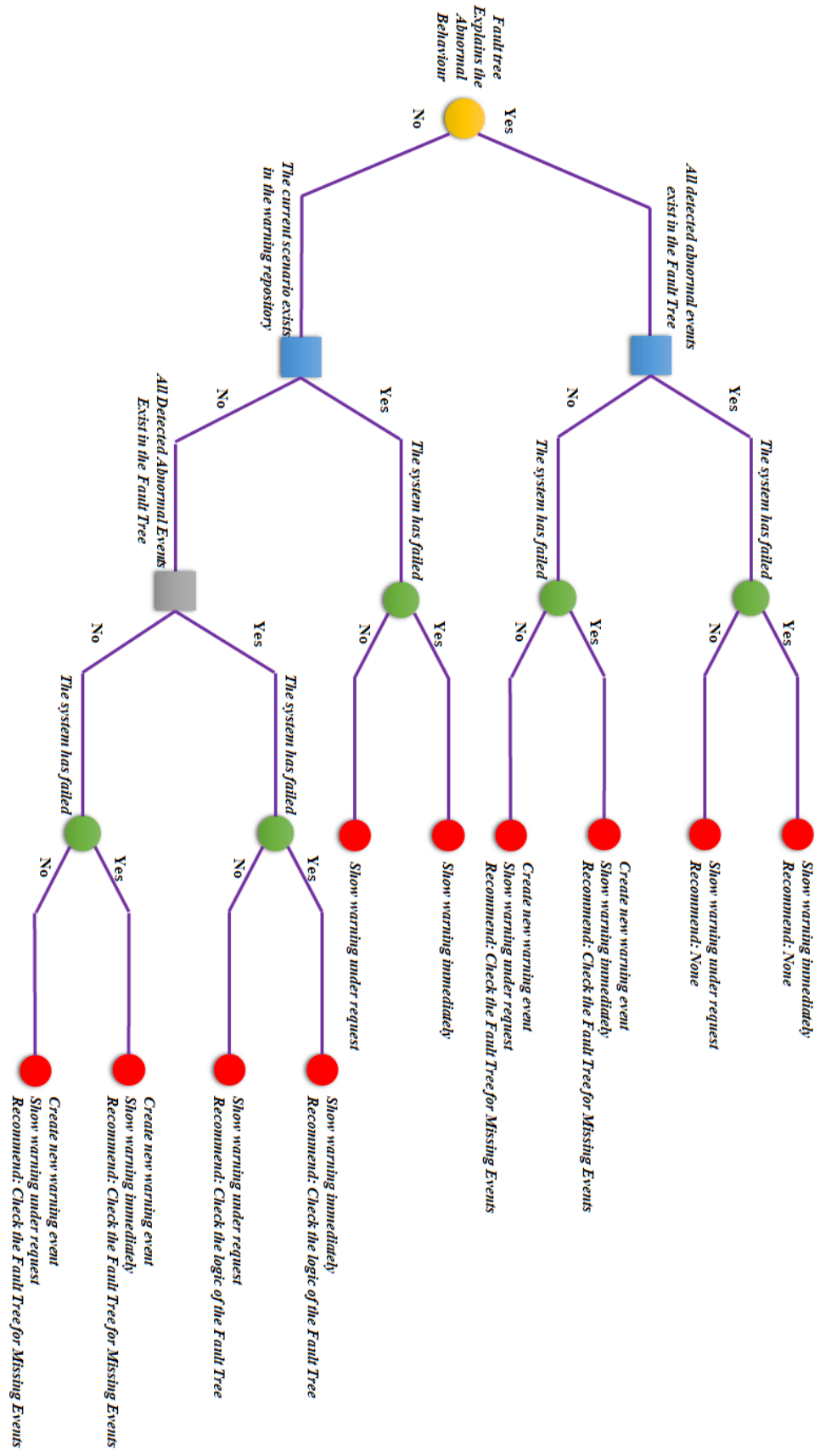

FIGURE 6. A decision tree for scenario classification.

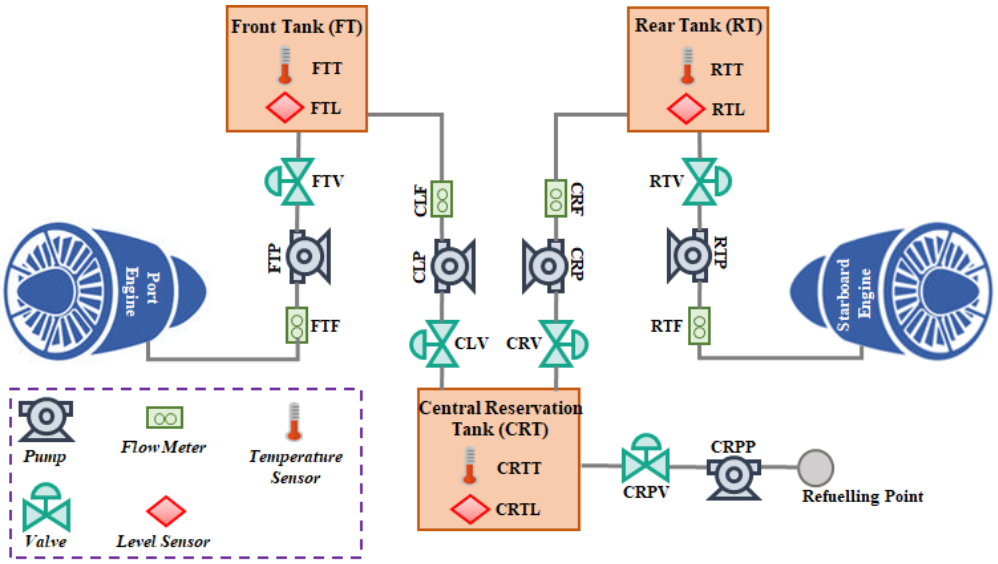

FIGURE 7. A schematic of aircraft fuel distribution system.

resources, such as the system fault tree, to check if the current abnormal scenario has been taken into consideration during the offline analysis. If not, action should be taken and warnings based on the system situation (fail or not) should be issued. After that, based on the final step of the branches, the system user is notified. Note that the warning is given immediately or on request based on the criticality of the scenario. If the detected anomalous behavior represents a system failure, then it is considered as critical. On the other hand, if it does not represent system failure, then the scenario is considered as less critical. When an abnormal system behavior is detected, the following two cases are possible.

Case 1: The fault tree of the system can explain this abnormal scenario. Although an explanation is found, the fault tree may or may not contain all the event(s) corresponding to these observed anomalies. If the fault tree contains all the related events, the decision-making block does not suggest any repair action for the fault tree. On the other hand, if there are some events missing in the fault tree then a recommendation is provided to check the fault tree for a potential repair to include the newly discovered events which were not seen during design stage. Note that whenever a recommendation is provided it is saved in a central repository for future use in a similar case.

Case 2: The fault tree cannot explain this abnormal behavior. That could either mean that during the design time creation of the fault tree the analysts were not able to foresee some events that are identified now, or they were able to identify all the events, but the logical relationships among the events were wrongly set in fault tree. Therefore, it is recommended either to repair the fault tree by including newly identified events or by correcting the logic of the fault tree. Note that, in this case, before generating any recommendation, the decision-making process first checks the warning repository to see if this particular scenario has been addressed in the past. If it was addressed before, then the warning is retrieved from the repository for reuse. If it is a new case, then a new appropriate warning event is created, and the required notification is provided based on the system situation (fail or not).

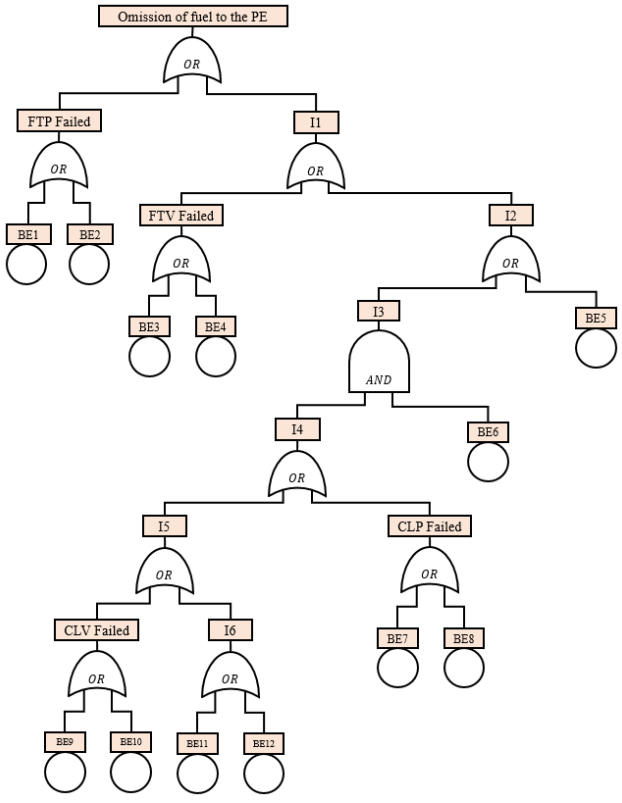

FIGURE 8. Fault tree of omission of fuel to the PE.

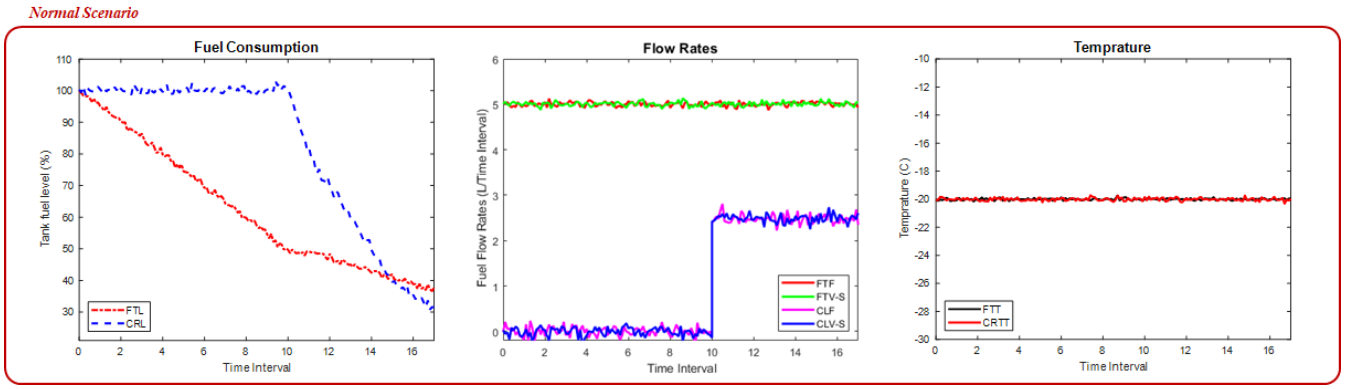

FIGURE 9. Readings from sensors under normal operating condition.

IV. CASE STUDY EVALUATION

To illustrate the concept proposed in this paper, we use a simplified version of an Aircraft Fuel Distribution System (AFDS) used in [43]. The system shown in Fig. 7 has two primary functions: storing fuel and distributing fuel to the engines. These functions are provided in refueling and consumption phases, respectively. During refueling, the fuel is first loaded in the Central Reservation Tank and then distributed to the Front and Rear Tanks. In the consumption phase, the two engines receive adequate level of fuel from the appropriate tanks. For instance, the Port Engine (PE) will receive fuel from Front Tank and the Starboard Engine (SE) will receive fuel from Rear Tank. Each of the tanks have a level sensor and a temperature sensor. They measure the level of fuel and the temperature of the tank, respectively. When the fuel level reaches to a pre-specified level in the Front and the Rear Tank, they can draw fuel from the central reservation tank. Similar to the tanks, the valves have their own sensor to measure the rate of flow through them. Additionally, there are flow sensors attached to the pipes to measure flow rate through the pipes.

As seen in the figure, the fuel flow paths to the PE and SE are identical and they only use a different set of components. For this reason, for illustrative purposes, in this paper we only consider the fuel flow path of the PE for further analysis. It was also assumed that the sensors are reliable; therefore, their failures are not considered in the analysis. A fault tree is derived by considering the *"Omission of Fuel to the Port Engine"* as the top event and shown in Fig. 8. Table 1 describes the meaning of the basic events and their associated components.

To illustrate the proposed idea, we consider that eight different sensors are used to monitor the real time behavior of the fuel flow path to the PE. The sensors are FTL and FTT (level and temperature sensor of front tank), CRTL and CRTT (level and temperature sensor of central tank), FTF and CLF (two flow sensors attached to two different points in the pipes), FTV-S (sensor on valve FTV), and CLV-S (sensor on valve CLV).

TABLE 1. Description of the basic and intermediate events of the fault tree in figure 8.

Component Name	Tags in FT	Failure Mode
FTP	BE1	FTP.Electromechanical_fail_stop
	BE2	FTP.Control_valve_stuck_at_zero
FTV	BE3	FTV.Stuck_Closed
	BE4	FTV.Comission-close_command
Front Tank	BE5	FT. Outlet_close
	BE6	FT.Leaked
CLP	BE7	CLP.Electromechanical_fail_stop
	BE8	CLP.Control_valve_stuck_at_zero
CLV	BE9	CLV.Stuck_Closed
	BE10	CLV.Comission-close_command
Central Reservation Tank	BE11	CRT. Outlet_close
	BE12	CRT.Leaked
	I1	No fuel flow through valve FTV
	I2	No fuel flow from front tank
	I3	Front tank's fuel level too low
	I4	No fuel flow through pump CLP
	I5	No fuel flow through valve CLV
	I6	No fuel flow from central tank

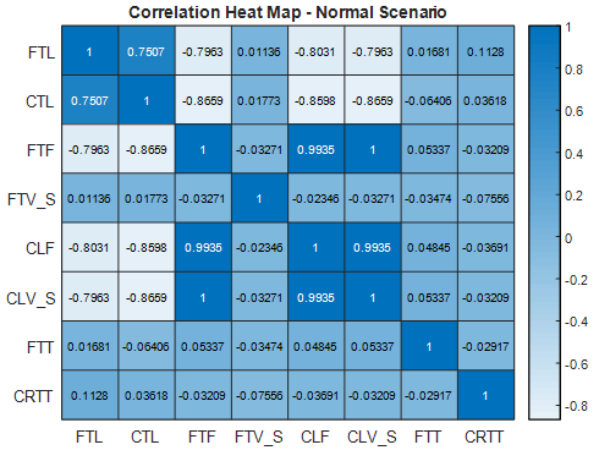

FIGURE 10. Correlation heat map for different variables in the normal operating condition.

Fig. 9 shows the readings from different sensors when the system works normally. Note that all the data used in this paper is hypothetical and used only for illustrative purposes.

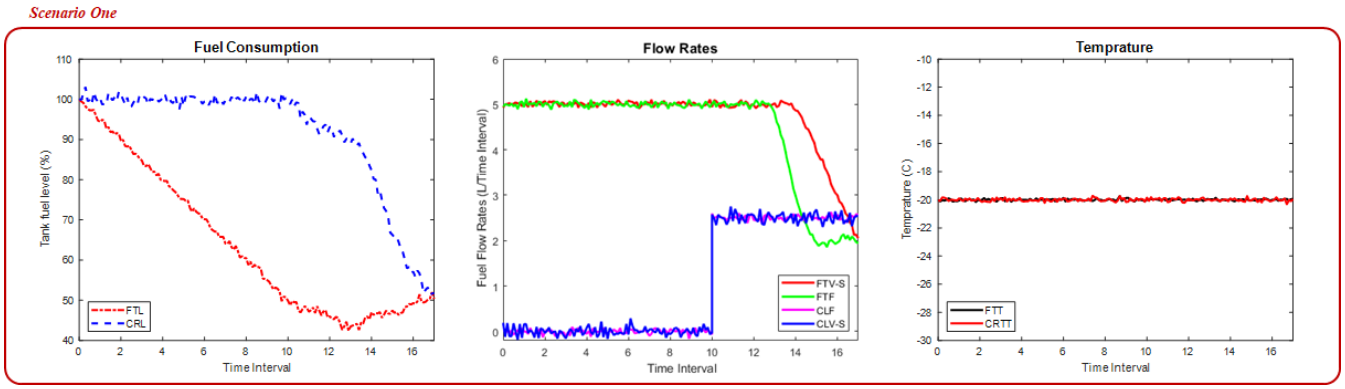

FIGURE 11. Readings from sensors in abnormal scenario 1.

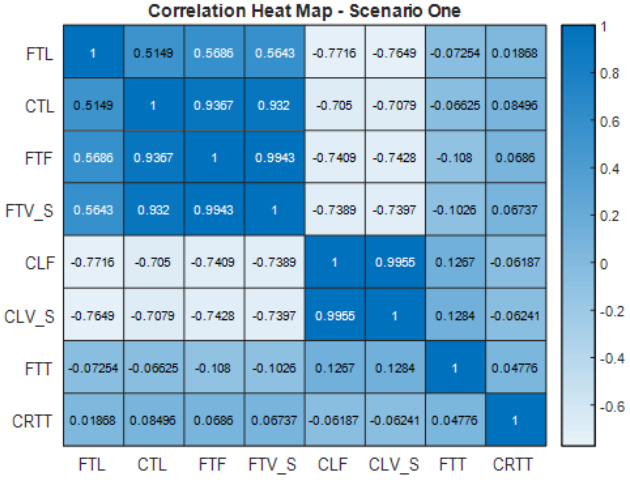

FIGURE 12. Correlation heat map for different variables in the scenario 1.

However, we believe that the real operational data from a practical system can be used in the same fashion. As seen in Fig. 9, it is assumed that in normal operating conditions, temperatures of the fuel in the front and the central tank are kept at $-20°$C. It is also seen that the front tank keeps providing fuel to the PE, without drawing any fuel from the central tank, until its fuel level drops to 50%. At time interval 10, the fuel level of front tank reaches to 50%, where it starts drawing fuel from the central tank, which is evident from the drop in the fuel level in the central tank and availability of fuel flow through CLF and CLV-S sensors. The central tank fuel level drops sharply compared to the left tank because it was assumed that both the left and right tanks are drawing fuel from the central tank at the same time. Moreover, after the time interval 10, the fuel level of the left tank drops less sharply than before because of the support it is receiving from the central tank. The correlation among different variables in the normal scenario is shown in Fig. 10.

To illustrate how the proposed approach will identify abnormal scenarios and suggest appropriate actions, we consider four different scenarios. Fig. 11 shows the first scenario. In this case, after time interval 13, the operators of the system notice that the port engine is starving of fuel. Independent of what the operators observe, the approach proposed in this paper detects an anomaly after time interval 13. The approach detects that the change in the readings from FTF and FTV-S cause the system to go outside its operational boundary. Given the abnormal behavior and the detected parameters, the approach finds correlations between these parameters with other parameters (see Fig. 12). From the

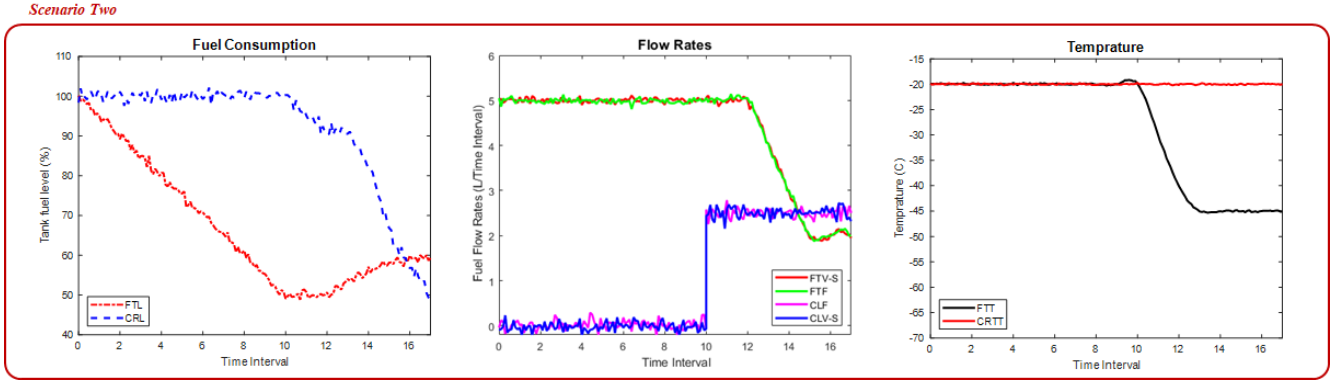

FIGURE 13. Readings from sensors in abnormal scenario 2.

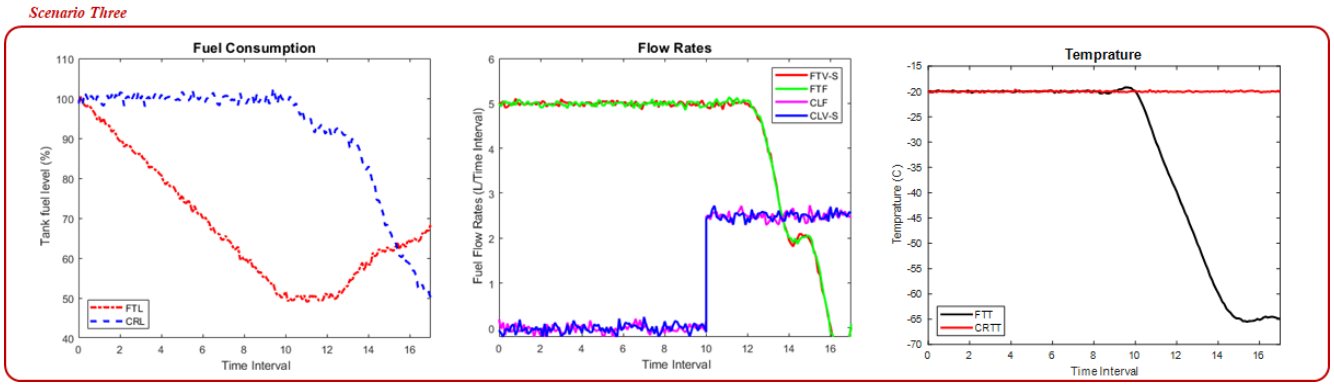

FIGURE 14. Readings from sensors in abnormal scenario 3.

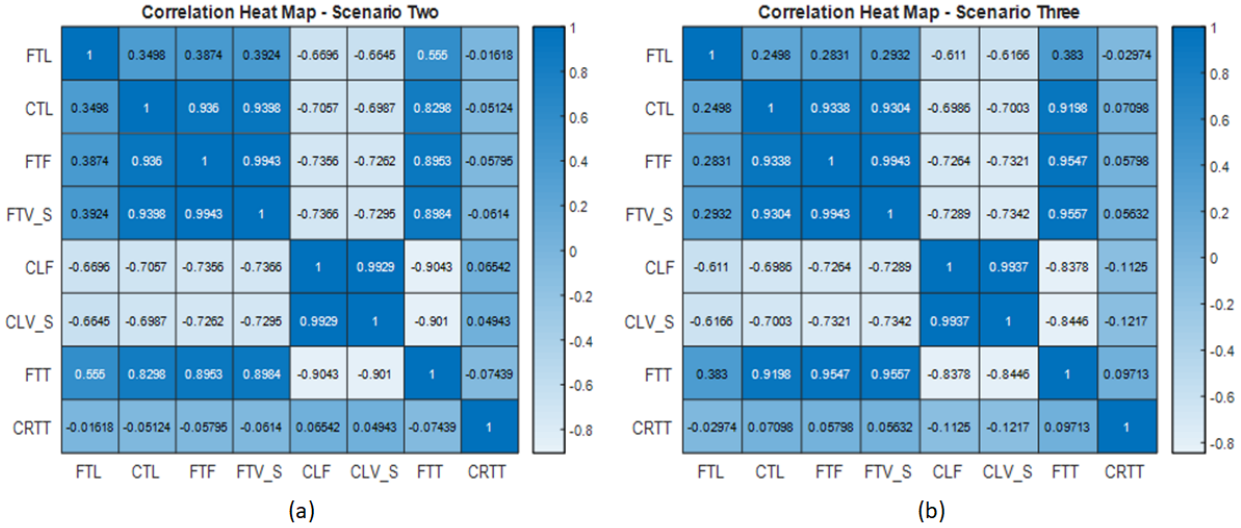

FIGURE 15. Correlation heat map for different variables in scenarios 2 and 3 (a) correlation heat map for different variables in the scenario 2 (b) correlation heat map for different variables in the scenario 3.

correlation heat map, it is seen that in this abnormal condition, the variables FTF and FTV-S themselves are highly correlated, i.e., a change in one of the variables may cause the problem in other variables. Moreover, these variables are highly correlated with CTL and FTL, which could potentially mean that the reduced rate of flow through pump FTP and valve FTV is responsible for reduced level of consumption from the front and central tank, consequently the reason for starvation of the port engine. The fault tree of Fig.8 can explain this scenario. This scenario corresponds to a case where the top event of the fault tree becomes true because of the occurrence of any of the BEs 1 to 4 (BEs associated with FTP and FTV). However, as Fig. 11 shows that the reduced fuel flow is detected by the FTF first, the position of this sensor in system (see Fig. 7) suggests that it is highly likely that the problem was with the pump FTP, i.e., BE 1 and/or BE2 in the fault tree of Fig. 8.

Figs. 13 and 14 represents two closely related scenarios. As seen in Fig. 13, like the normal operation mode (see Fig. 9), after time interval 10, the front tank starts drawing fuel from the central tank. However, unlike the normal operation mode, the temperature of the fuel in the front tank starts dropping after time interval 10. After time interval 12, the temperature recorded by FTT reaches to $-45°C$ and stays at the same level afterwards. At the same time, after time interval 12, the fuel flow rate recorded by FTF and FTV-S starts dropping. The proposed approach detects that the values recorded by FTT force the system to go outside its operational boundary. The correlation heat map in Fig. 15(a) shows that in scenario 2, FTT has very high correlation with FTF, FTV-S, CTL, and FTL. That means the drop-in temperature recorded by FTT is responsible for the abnormal system behavior. In this scenario, as the fuel system is operating in a degraded mode, a warning event would be generated by incorporating this newly found knowledge, and the warning will be shown on request.

Scenario 3, shown in Fig. 14, represents further degradation from scenario 2, where the temperature recorded by FTT dropped further to $-65°C$. At the same time, unlike scenario 2, the fuel flow rate recorded by FTF and FTV-S

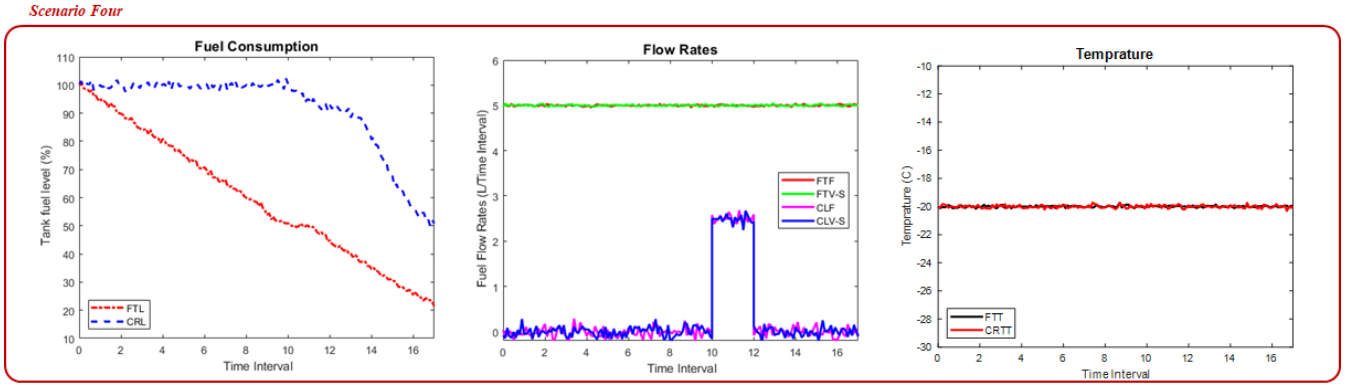

FIGURE 16. Readings from sensors in abnormal scenario 4.

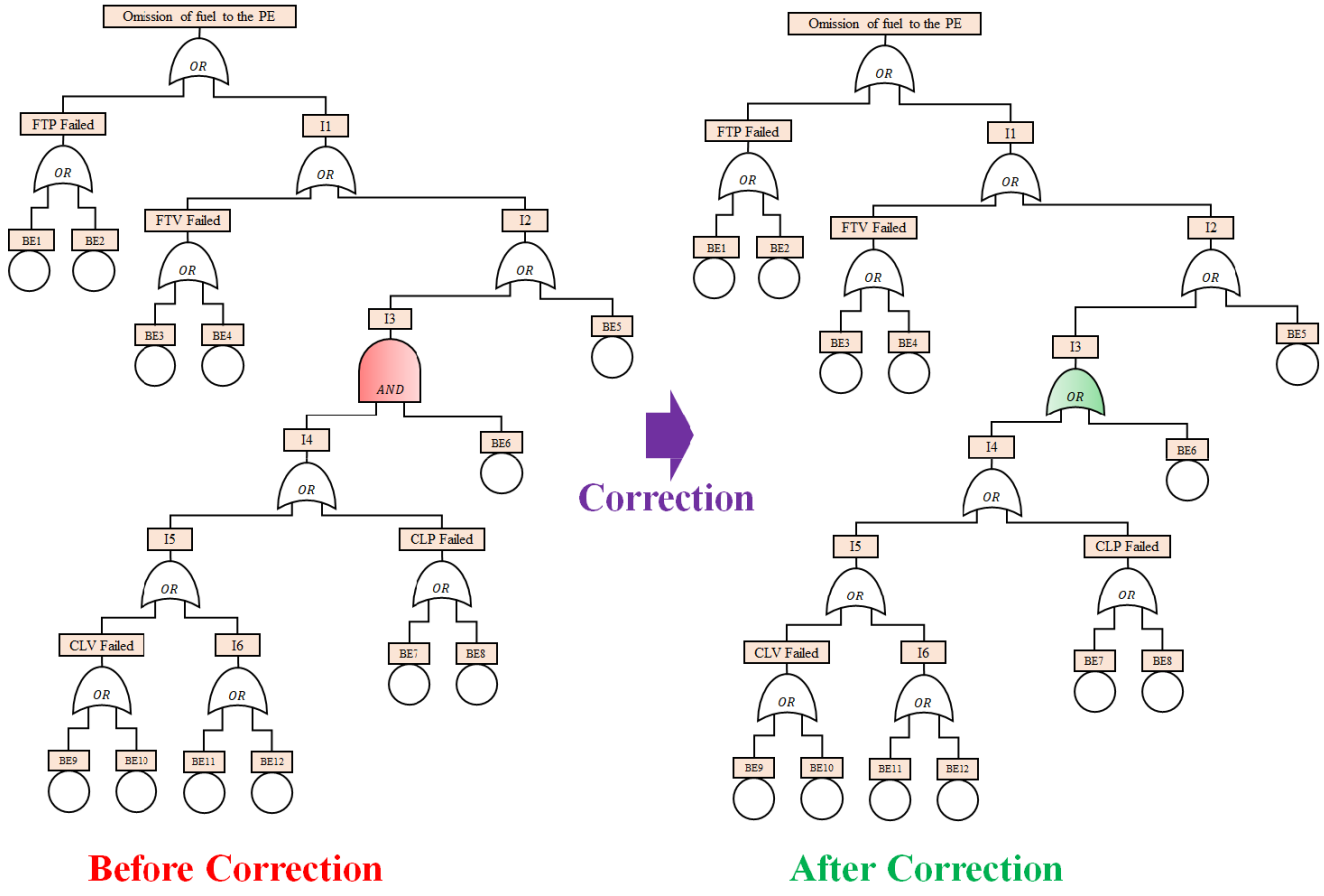

FIGURE 17. Fault Tree correction based on the recommendation of the proposed method for scenario 4.

reaches to zero, meaning that there is no fuel flow to the PE. The updated correlation heat map for this scenario is shown in Fig. 15(b). According to the heat map, correlation of FTT with FTF, FTV-S, and CTL increased further, which suggests that the temperature of the fuel in the front engine is responsible for this abnormality. An inspection of the fault tree of Fig.8 reveals that this scenario is not explainable by the fault tree because in the fault tree there is no event related to temperature of fuel in the front engine. As a result, a warning event is created with the suggestion to repair the fault tree by including an event related to the temperature of the fuel.

The scenario 4 (see Fig. 16) shows that at time interval 10, the central fuel tank starts providing fuel to the front tank, which is an expected behavior. However, after time interval 12, the central tank stopped providing fuel to the front tank as evidenced by no flow reading from CLF and CLV-S. No flow reading from CLF and CLV-S could be attributed to the failure of either or both of pump CLP and valve CLV.

Because of this, the front tank has to feed the PE alone, which results into very low fuel level at the front tank. Moreover, throughout this time, the readings at FTT and CRTT remain the same. In this case, an abnormal behavior is detected by the proposed approach and the variables detected for this abnormality are FTL, CLF, and CLV-S. Although an abnormality is detected by the proposed approach and there are events associated with all the detected abnormal parameters in the fault tree, the fault tree of Fig. 8 was still not able to explain this scenario. It is clear that this abnormality is because of the low fuel level at the front tank, which corresponds to event I3 (front tank's fuel level too low) in the fault tree. However, for some reason this event does not become true. Therefore, there may exist a problem with the logical relationship among the events that can cause intermediate event I3. For this reason, the proposed approach would create a warning event with the suggestion to check the fault tree for the correctness of the logic gate used for modelling the behavior of event I3 and other events downwards.

Based on the recommendation, the fault tree of Fig. 8 is corrected by replacing the AND gate below I3 by an OR gate (see Fig. 17). After this correction, the fault tree was able to explain the scenario 4.

V. CONCLUSION

This work utilises machine learning based approaches to capture the real time behavior of a system based on real-time operational data. During monitoring, if it is found that the system deviates from its normal behavior, a number of recommendations have been provided based on the nature of the detected abnormality and the ability of the safety artefacts of the system to explain the abnormality. These recommendations include the potential repair of the fault tree of the system via the inclusion of new basic events and/or via the correction of the logical structure of the fault tree.

From historical evidence, it is clear that during safety analysis in the design phase of system development, it is possible that the analysts may not foresee all possible causes of failure and they may develop safety artefacts based on the wrong assumptions and wrong understanding of the system behavior. Such limitations can only be uncovered during the operation of the system. The approach proposed in this paper is an attempt to address these issues by taking into account the monitoring data. The primary advantage of the approach is that it can provide additional knowledge about the safety model and the system behavior to the system user when an unknown scenario is encountered.

Currently, we demonstrated the effectiveness of the approach by applying it on a simplified aircraft fuel distribution system, based on hypothetical data and scenarios. However, in the future, we plan to verify the usefulness and scalability of the approach by applying it to more complex systems with real operational data. One challenge in this case would be the availability of operational data of a system and the willingness of the system owners to share the data.

AKNOWLEDGEMENT

The authors would like to thank EDF Energy R&D U.K. Centre and AURA Innovation Centre for their support.

REFERENCES

[1] G. Latif-Shabgahi, J. M. Bass, and S. Bennett, "A taxonomy for software voting algorithms used in safety-critical systems," *IEEE Trans. Rel.*, vol. 53, no. 3, pp. 319–328, Sep. 2004.
[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.
[3] K. S. Trivedi and A. Bobbio, *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[4] S. Kabir, M. Yazdi, J. I. Aizpurua, and Y. Papadopoulos, "Uncertainty-aware dynamic reliability analysis framework for complex systems," *IEEE Access*, vol. 6, pp. 29499–29515, 2018.
[5] Y. Yu and B. W. Johnson, "Safety assessment for safety-critical systems using Markov chain modular approach," *Int. J. Rel., Qual. Saf. Eng.*, vol. 18, no. 2, pp. 139–157, 2011.
[6] S. Distefano, F. Longo, and K. S. Trivedi, "Investigating dynamic reliability and availability through state–space models," *Comput. Math. Appl.*, vol. 64, pp. 3701–3716, Dec. 2012.
[7] K. S. Trivedi and A. Bobbio, *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[8] X. Wang, P. Jia, H. Lizhang, L. Wang, F. Yun, and H. Wang, "Reliability and safety modelling of the electrical control system of the subsea control module based on Markov and multiple beta factor model," *IEEE Access*, vol. 7, pp. 6194–6208, 2018.
[9] K. Aslansefat and G. Latif-Shabgahi, "A hierarchical approach for dynamic fault trees solution through semi-Markov process," *IEEE Trans. Rel.*, to be published. doi: 10.1109/TR.2019.2923893.
[10] S. Kabir, "An overview of fault tree analysis and its application in model based dependability analysis," *Expert Syst. Appl.*, vol. 77, pp. 114–135, Jul. 2017.
[11] *IET Code of Practice: Competence for Safety Related Systems Practitioners Covers*, Inst. Eng. Technol., Eng. Saf. Consultants Ltd., London, U.K., 2016.
[12] M. Manion, "The epistemology of fault tree analysis: An ethical critique," *Int. J. Risk Assessment Manage.*, vol. 7, pp. 382–430, Jan. 2007.
[13] A. Rae, J. McDermid, and R. Alexander, "The science and superstition of quantitative risk assessment," *J. Syst. Saf.*, vol. 48, pp. 28–38, 2012.
[14] A. Rae, R. Alexander, and J. McDermid, "Fixing the cracks in the crystal ball: A maturity model for quantitative risk assessment," *Rel. Eng. Syst. Saf.*, vol. 125, pp. 67–81, May 2014.
[15] J. Suokas and P. Pyy, "Evaluation of the validity of four hazard identification methods with event descriptions," VTT Tech. Res. Centre Finland, Espoo, Finland, 1988.
[16] G. Carter and S. D. Smith, "Safety hazard identification on construction projects," *J. Construct. Eng. Manage.*, vol. 132, no. 2, pp. 197–205, 2006.
[17] C. Perrow. (Apr. 1, 2011). *Fukushima, Risk, and Probability: Expect the Unexpected*. [Online]. Available: https://thebulletin.org/2011/04/fukushima-risk-and-probability-expect-the-unexpected-2/
[18] N. Onishi and J. Glanz. (Mar. 26, 2011). *Japanese Rules for Nuclear Plants Relied on Old Science*. [Online]. Available: https://www.nytimes.com/2011/03/27/world/asia/27nuke.html
[19] J. Joyce and K. Wong, "Hazard-driven testing of safety-related software," in *Proc. 21st Int. Syst. Saf. Conf.*, Vancouver, BC, Canada, 2003, pp. 1–10.
[20] N. Leveson, "A new accident model for engineering safer systems," *Saf. Sci.*, vil. 42, pp. 237–270, Apr. 2004.
[21] E. Lloyd and W. Tye, *Systematic Safety: Safety Assessment of Aircraft Systems*. London, U.K.: Civil Aviation Authority, 1982.
[22] *Safety Investigation Into the Accident on June 2009 to the Airbus A330–203, Flight AF447*, BEA, Paris, France, 2009.
[23] W. van der Aalst, "Data science in action," *Process Mining*, vol. 1, pp. 3–23, Apr. 2016. doi: 10.1007/978-3-662-49851-4_1.
[24] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004.

[25] G. Greco, A. Guzzo, L. Pontieri, and D. Sacca, "Discovering expressive process models by clustering log traces," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1010–1027, Aug. 2006.
[26] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wires Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 182–192, 2012.
[27] D. Fahland and W. M. P. van der Aalst, "Model repair—Aligning process models to reality," *Inf. Syst.*, vol. 47, pp. 220–243, Jan. 2015.
[28] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo, "Automated discovery of process models from event logs: Review and benchmark," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 686–705, Apr. 2018.
[29] E. E. Hurdle, L. M. Bartlett, and J. D. Andrews, "System fault diagnostics using fault tree analysis," *Proc. Inst. Mech. Eng. O, J. Risk Rel.*, vol. 221, no. 1, pp. 43–55, 2007.
[30] M. Lampis and J. Andrews, "Bayesian belief networks for system fault diagnostics," *Qual. Rel. Eng. Int.*, vol. 25, no. 4, pp. 409–426, 2009.
[31] B. Cai, Y. Liu, Y. Ma, Z. Liu, Y. Zhou, and J. Sun, "Real-time reliability evaluation methodology based on dynamic Bayesian networks: A case study of a subsea pipe ram BOP system," *ISA Trans.*, vol. 85, pp. 595–604, Sep. 2015.
[32] M. Askarian, R. Zarghami, F. Jalali-Farahani, and N. Mostoufi, "Fusion of micro-macro data for fault diagnosis of a sweetening unit using Bayesian network," *Chem. Eng. Res. Des.*, vol. 115, pp. 325–334, Nov. 2016.
[33] J. Z. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mech. Syst. Signal Process.*, vol. 25, no. 5, pp. 1803–1836, Jul. 2011.
[34] J. I. Aizpurua, V. M. Catterson, Y. Papadopoulos, F. Chiacchio, and G. Manno, "Improved dynamic dependability assessment through integration with prognostics," *IEEE Trans. Rel.*, vol. 66, no. 3, pp. 893–913, Sep. 2017.
[35] J. I. Aizpurua, V. M. Catterson, Y. Papadopoulos, F. Chiacchio, and D. D'Urso, "Supporting group maintenance through prognostics-enhanced dynamic dependability prediction," *Rel. Eng. Syst. Saf.*, vol. 168, pp. 171–188, Dec. 2017.
[36] G. Chen and Z. Ge, "Hierarchical Bayesian network modeling framework for large-scale process monitoring and decision making," *IEEE Trans. Control Syst. Technol.*, to be published.
[37] S. Getir, L. Grunske, A. van Hoorn, T. Kehrer, Y. Noller, and M. Tichy, "Supporting semi-automatic co-evolution of architecture and fault tree models," *J. Syst. Softw.*, vol. 142, pp. 115–135, Apr. 2018.
[38] C. Agrell, S. Eldevik, A. Hafver, F. Pedersen, E. Stensrud, and A. Huseby, "Pitfalls of machine learning for tail events in high risk environments," in *Safety and Reliability—Safe Societies in a Changing World*. Boca Raton, FL, USA: CRC Press, 2018, pp. 3043–3051.
[39] E. Simen, C. Agrell, A. Hafver, and F. B. Pedersen, "AI + SAFETY: Safety implications for artificial intelligence," DNV GL, Oslo, Norway, Tech. Rep., 2018. [Online]. Available: https://ai-and-safety.dnvgl.com
[40] J. Cheng, C. Zhu, W. Fu, C. Wang, and J. Sun, "An Imitation medical diagnosis method of hydro-turbine generating unit based on Bayesian network," *Trans. Inst. Meas. Control*, vol. 41, no. 12, pp. 3406–3420, 2019.
[41] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.
[42] J. A. Carino, M. Delgado-Prieto, D. Zurita, M. Millan, J. A. O. Redondo, and R. Romero-Troncoso, "Enhanced industrial machinery condition monitoring methodology based on novelty detection and multi-modal analysis," *IEEE Access*, vol. 4, pp. 7594–7604, 2016.
[43] S. Kabir, M. Walker, and Y. Papadopoulos, "Dynamic system safety analysis in HiP-HOPS with Petri nets and Bayesian networks," *Saf. Sci.*, vol. 105, pp. 55–70, Jun. 2018.

YOUCEF GHERAIBIA received the Ph.D. degree in computer engineering from the University of Annaba, Algeria, in 2016. He is currently a Research Associate of the Assuring Autonomy International Programme with the University of York. Prior to that, he was a Research Assistant with the Department of Computer Science and Technology, University of Hull. He has published over 20 articles on his research-related topics. His research interests include optimization, machine learning, probabilistic risk and safety analysis, and autonomous system safety. In 2016, he was a recipient of the National Innovation Award from the Algerian Government.

SOHAG KABIR received the Ph.D. degree in computer science and the M.Sc. degree in embedded systems from the University of Hull, U.K., in 2016 and 2012, respectively, where he is currently a Research Associate with the Dependable Intelligent Systems (DEIS) Research Group. He has worked in EU projects on safety including MAENAD and DEIS. His research interests include model-based safety assessment, probabilistic risk and safety analysis, dynamic safety and reliability analysis, and stochastic modelling and analysis.

KOOROSH ASLANSEFAT (M'16) was born in Tehran, Iran, in 1989. He received the B.Sc. degree in marine electronic and communication engineering from Chabahar Maritime University, Chabahar, Iran, in 2011, and the M.Sc. degree in control engineering from Shahid Beheshti University, Tehran, Iran, in 2014. He is currently pursuing the Ph.D. degree with the University of Hull, Hull, U.K., where he was involved in data-driven reliability-centered evolutionary and automated maintenance for offshore wind farms. His main research interests include Markov modeling, performance assessment, artificial intelligence, optimization, and stochastic modeling.

IOANNIS SOROKOS received the B.Sc. degree in computer science from the Athens University of Economics and Business, Greece, in 2011, and the M.Sc. and Ph.D. degrees in computer science from the University of Hull, in 2013 and 2017, respectively, where he is currently a Postdoctoral Researcher. His research interests include model-based dependability analysis and assurance, metaheuristic optimization, artificial intelligence, computer graphics, and computational game theory.

YIANNIS PAPADOPOULOS has pioneered work on model-based dependability assessment and evolutionary optimization of complex engineering systems known as hierarchically performed hazard origin and propagation studies (HiP-HOPS). He has coauthored EAST-ADL, an emerging automotive architecture description language with Volvo, Honda, Continental, Honeywell, and DNV-GL, among others. He is actively involved in two technical committees of IFAC (TC 1.3 and 5.1). He is also working on new metaheuristics inspired by the hunting behavior of penguins and developing technologies for self-certification of cyber-physical and autonomous systems. He is currently a Professor and the Leader of the Dependable Intelligent Systems Research Group with the University of Hull. His research interests include digital arts and various aspects of philosophy and its interactions with science.