POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

A Novel Data-Driven Fault Tree Methodology for Fault Diagnosis and Prognosis

Kerelous Refaat Latef Waghen

Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiae Doctor*

Génie industriel

Février 2020

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

A Novel Data-Driven Fault Tree Methodology for Fault Diagnosis and Prognosis

présentée par **Kerelous Refaat Latef WAGHEN**

en vue de l'obtention du diplôme de *Philosophiae Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Bernard CLÉMENT, président

Mohamed-Salah OUALI, membre et directeur de recherche

Tamer GAYED, membre

Dragan KOMLJIENOVIC, membre externe

DEDICATION

1 dedicate this research to my beloved parents Refaat and Mariam, my sister Christina, my brother Mina and my friend Mina Refaat for their support.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Mohamed-Salah Ouali, who dedicated himself wholehearted to my research. He tried to understand my ideas and helped me to improve them. I admire his patience, his comprehension, and his great support and devotion to me and my research. I also wish to express to him my most sincere gratitude for his emotional support.

Also, I would like to thank Dr. Mouloud Amazouz and Dr. Ahmed Ragab for their contribution on my research through providing me practical experiences in CanmetENERGY-Varennes that adapt my work to be more applicable in the industry.

RÉSUMÉ

La thèse développe une nouvelle méthodologie de diagnostic et de pronostic de défauts dans un système complexe, nommée Interpretable logic tree analysis (ILTA), qui combine les techniques d'extraction de connaissances à partir des bases de données « knowledge discovery in database (KDD) » et l'analyse d'arbre de défaut « fault tree analysis (FTA) ». La méthodologie capitalise les avantages des deux techniques pour appréhender la problématique de diagnostic et de pronostic de défauts. Bien que les arbres de défauts offrent des modèles interprétables pour déterminer les causes possibles à l'origine d'un défaut, leur utilisation pour le diagnostic de défauts dans un système industriel est limitée, en raison de la nécessité de faire appel à des connaissances expertes pour décrire les relations de cause-à-effet entre les processus internes du système. Cependant, il sera intéressant d'exploiter la puissance d'analyse des arbres de défaut mais construit à partir des connaissances explicites et non biaisées extraites directement des bases de données sur la causalité des fautes. Par conséquent, la méthodologie ILTA fonctionne de manière analogue à la logique du modèle d'analyse d'arbre de défaut (FTA) mais avec une implication minimale des experts. Cette approche de modélisation doit rejoindre la logique des experts pour représenter la structure hiérarchique des défauts dans un système complexe.

La méthodologie ILTA est appliquée à la gestion des risques de défaillance en fournissant deux modèles d'arborescence avancés interprétables à plusieurs niveaux (MILTA) et au cours du temps (ITCA). Le modèle MILTA est conçu pour accomplir la tâche de diagnostic de défaillance dans les systèmes complexes. Il est capable de décomposer un défaut complexe et de modéliser graphiquement sa structure de causalité dans un arbre à plusieurs niveaux. Par conséquent, un expert est en mesure de visualiser l'influence des relations hiérarchiques de cause à effet menant à la défaillance principale. De plus, quantifier ces causes en attribuant des probabilités aide à comprendre leur contribution dans l'occurrence de la défaillance du système. Le modèle ITCA est conçu pour réaliser la tâche de pronostic de défaillance dans les systèmes complexes. Basé sur une répartition des données au cours du temps, le modèle ITCA capture l'effet du vieillissement du système à travers de l'évolution de la structure de causalité des fautes. Ainsi, il décrit les changements de causalité résultant de la détérioration et du vieillissement au cours de la vie du système.

ABSTRACT

The thesis develops a new methodology for diagnosis and prognosis of faults in a complex system, called Interpretable logic tree analysis (ILTA), which combines knowledge extraction techniques from knowledge discovery in databases (KDD) and the fault tree analysis (FTA). The methodology combined the advantages of the both techniques for understanding the problem of diagnosis and prognosis of faults. Although fault trees provide interpretable models for determining the possible causes of a fault, its use for fault diagnosis in an industrial system is limited, due to the need for expert knowledge to describe cause-and-effect relationships between internal system processes. However, it will be interesting to exploit the analytical power of fault trees but built from explicit and unbiased knowledge extracted directly from databases on the causality of faults. Therefore, the ILTA methodology works analogously to the logic of the fault tree analysis model (FTA) but with minimal involvement of experts. This modeling approach joins the logic of experts to represent the hierarchical structure of faults in a complex system.

The ILTA methodology is applied to failure risk management by providing two interpretable advanced logic models: a multi-level tree (MILTA) and a multilevel tree over time (ITCA). The MILTA model is designed to accomplish the task of diagnosing failure in complex systems. It is able to decompose a complex defect and graphically model its causal structure in a tree on several levels. As a result, an expert is able to visualize the influence of hierarchical cause and effect relationships leading to the main failure. In addition, quantifying these causes by assigning probabilities helps to understand their contribution to the occurrence of system failure. The second model is a logical tree interpretable in time (ITCA), designed to perform the task of prognosis of failure in complex systems. Based on a distribution of data over time, the ITCA model captures the effect of the aging of the system through the evolution of the fault causation structure. Thus, it describes the causal changes resulting from deterioration and aging over the life of the system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADT	Accelerated Degradation Tests
AM	Asset Management
ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial neural networks
BIFIM	Bayesian intelligent fault inference model
BN	Bayesian network
BRB	Belief rules base
BLSTM	Bi-Directional Long Short-Term Memory
BSWI	B-spline wavelet on the interval
CA	Causality analysis
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CAFT	Computer-aided fault tree
CBM	Condition-based maintenance
CM	Corrective maintenance
DSS	Decision support system
DT	Decision tree
DBN	Deep belief networks
DNN	Deep neural networks
DDAG	Diagnostic directed acyclic graph
DG	Direct graph
DET	Dynamic event tree
DTW	Dynamic Time Warping
EM	Emergency maintenance
ETA	Event tree analysis
FD	Fault diagnosis
FT	Fault tree
FTA	Fault tree analysis

FTKD	Fault tree based on knowledge discovery
FFTA	Fuzzy fault tree framework
GC	Granger causality
HiP-HOPS	Hierarchically Performed Hazard Origin & Propagation Studies
ILTA	Interpretable logic tree
ILTCA	Interpretable time causality analysis
KPCA	Kernel principal component analysis
KNN	K-nearest-neighbors
KDD	Knowledge discovery in database
LAD	Logical analysis of data
MTBF	Mean time between failure
MTTF	Mean time to failure
MBK PLS	Multiblock kernel partial least squares
MILTA	Multi-level ILTA
NRS	Neighbourhood rough set
Op-SWPT	Optimized Stationary Wavelet Packet Transform
PCP	Parallel coordinates plot
PLS	Partial Least Squares
PRE	Prediction Rule Ensembles
PM	Preventive maintenance
PCA	Principal components analysis
PHM	Proportional hazards modelling
QDA	Quadratic discriminant analysis
RF	Random Forest
RUL	Remaining useful life
RBM	Restricted Boltzmann Machine
RST	Rough sets theory
SDG	Signed directed graph
SVM	Support vector machine
TNFS	Temporal neuro-fuzzy system

TBM	Time-based maintenance
TDMI	Time-delayed mutual information
TD-SDG	Time-delayed signed digraph
TFF	Time-to-failure
WPHM	Weibull Proportional Hazards Model
EWMA	Weighted moving average

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Manufacturing before the 18th century was occurring in people's homes, using very basic tools. In this era, manufacturing systems were very simple using few components, so their interrelations and behaviors were fully comprehensible and predictable (Taguchi & Chowdhury, 2004). During this century, a major industrial revolution took place and changed the manufacturing process. Consequently, the world's economy also shifted from handicrafts to an industrialized economy, which transformed machinery systems and resulted in mass production.

Over the past three centuries, innovations and development in an industrialized organization helped automate industrial processes to achieve more complex tasks. The implementation of cloud computing, internet of things (IOT), artificial intelligence (AI) and cyber-physical systems have revolutionized this era to build a new generation of systems known as "Industry 4.0", which are smarter systems with more complex structures and many connected subsystems and components (Martinez, Lara, Saucedo, & Fierro, 2018).

Although Industry 4.0 has had a dramatic impact on enhancing the productivity of industrial systems and their automated processes, this new concept has led to further complication to a system's structure (Mourtzis, Papakostas, & Makris, 2019). These new systems are mainly characterized by non-linear dynamics and have diverged from their superposition principles (Socolar, 2006). In addition, they are composed of many interconnected subsystems and components with a complex hierarchical structure. Their interrelations and interdependencies are complex to understand, compared to old systems (Torngren & Sellgren, 2018).

Therefore, the challenge is to sustain the performance of complex systems at a level that will maximize the system's availability and productivity. This objective can be achieved through effective risk management, which considers the different risk factors that result from a system's diversity and connectivity. This effectiveness is accomplished through the involvement of different approaches and tools that are conducted by the new technologies' capabilities and the power of data.

This chapter is organized as follows. Section 1.1 depicts a risk management roadmap and the possible types of risks that the industrial system could face. Then, focus is made to the fault risk, as the most important and avoidable risk that needs to be managed in order to sustain the system's performance at an acceptable level. In Section 1.2, the different types of maintenance plans are investigated in order to propose a reliable risk management strategy to manage fault risk. Accordingly, condition-based maintenance (CBM) is selected because of its advantages over other strategies in managing the fault through its continuous monitoring. In Section 1.3, CBM steps are discussed, where more focus is relayed to the decision-making step. The different approaches for constructing fault diagnosis and prognosis models are explored, where these models form the core of the CBM decision making for fault management. In Section 1.4, the problem statement that crystalizes this thesis is depicted and followed by the thesis objectives in Section 1.5. Finally, the last section represents the organization of the thesis.

1.1 Risk Management of a Complex System

Complicated and complex systems are fundamentally different systems in term of understanding their events causality, system linearity and controllability. Complicated system includes many subsystems that have independent relationship, its event causality has linear cause-and-effect pathways and every system output has a corresponding input. On the other hand, complex system includes a network of subsystems and components that characterized with complex dependences and multiple interacting causes that hard to recognize with a clear pattern (Grabowski & Strzalka, 2008).

Due to nowadays systems complexity, managing the system physical asset such as components, equipment and production line required diversified capabilities such as human, informational and finical capabilities. Asset Management (AM) focus on employs those relevant capabilities for managing the whole system physical asset portfolio. Through AM, the assets that have direct or indirect impacts on the customer (e.g. assets that part of production line) are managed by avoiding the failures and directly absorbing the consequences (Abbas & Shafiee, 2020). Risk management has turn to be integral part of an efficient AM implementation, as the levels of system complexity

that needs specialization increases. Dedicated Risk Management is designated to control the system risks myriad and diversity to which assets are exposed (Analouei, Taheriyoun, & Safavi, 2020).

Risk management exploits engineering and management knowledge to reduce and eliminate risks (Gilbert, Amalberti, Laroche, & Paries, 2007). This objective can be achieved by understanding the elements of the system and their related risks in order to modify or redesign the system until reaching an acceptable level of safety. The essential steps of risk management are depicted in Figure 1.1 according to ISO 31000 (Purdy, 2010).

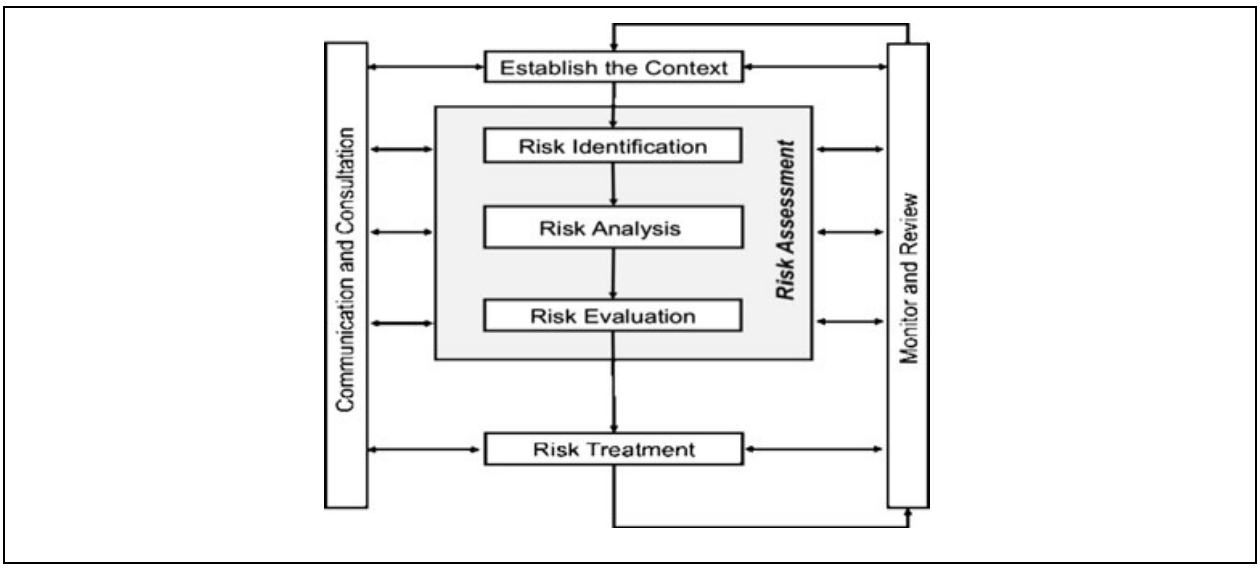


Figure 1.1 Schematic diagram for risk management ISO 31000:2009 (Purdy, 2010)

The “Establish the Context” step defines the context of the risks based on the system structure and the relations of its outputs & inputs. The “Risk Identification” step identifies the possible internal and external risks. The “Risk analysis” step evaluates and quantifies each identified risk and its impact on the system. The “Risk evaluation” step validates the risk feasibility and evaluate the cyber-risks acceptance. The “Risk treatment” step proposes a set of decisions in order to minimize the risk impacts on the system.

The Risk identification is the main step of the roadmap towards successful risk analysis and management. At this step, the risks are classified according to their probabilities and likelihood. The risks are later inspected at the risk analysis step. The diversity of aspects in a complex system, such as technical and socioeconomic, urges to propose new risk classifications, facilitate the risk identification step.

Institutions and enterprises define risks through different perceptions. Each perception is shaped based on a system's structure. In general, Kaplan, and Mikes (2012) summarized the different risk classifications into three main categories: strategic, external and avoidable risks.

Strategic risks happen due to failures in managerial and business decisions. External risks are hard to predict due the causes of the external factors. On the other hand, avoidable risks are internal abnormal events in the system that have direct impact on the system's performance and its production outcomes. Contrary to strategic and external risks, avoidable risks are derived from shortening in a system's internal dynamics and can be directly managed through system control parameters. The occurrences of avoidable risks can be quantified by analyzing a system's reliability (Yoon & Youn, 2019).

The most well-known, avoidable risk in industrial systems is the fault. The fault is an internal event caused from a deviation in the system behaviour and its expected normal response. It is treated in the same manner, as depicted in Figure 1.1. The fault risk is identified and its consequences are analyzed and evaluated to implement the risk management that minimizes its occurrence and its contribution to system failure (Zhi-jun & Yan, 2015).

The main objective of fault risk management is to sustain the system at acceptable performance and security levels by eliminating their occurrence. According to Kaplan, and Mikes (2012), the procedures of eliminating those risks can be achieved through different maintenance strategies. The strategies are depicted in the next subsection, along with their strengths and limitations for better managing the fault risk in industrial systems.

1.2 Maintenance strategies

Maintenance strategies are focused on eliminating the consequences of avoidable risks (faults) in industrial systems (He, X., Tong, & Chen, 2007). They integrate both reliability and safety in order to develop an optimal maintenance strategy for managing faults and minimizing the probability of a system failure that may occur from potential faults. This topic remains a critical research point for reliability and safety engineering that has pushed them to develop different maintenance strategies for maintaining the system at reliable and cost-effective operations (Telukdarie, Ndlovu, & Medoh, 2018). Various maintenance strategies over the past several years have been proposed.

According to the literature, they can be divided into two categories: “Corrective Maintenance” and “Preventive Maintenance,” as depicted in Figure 1.2.

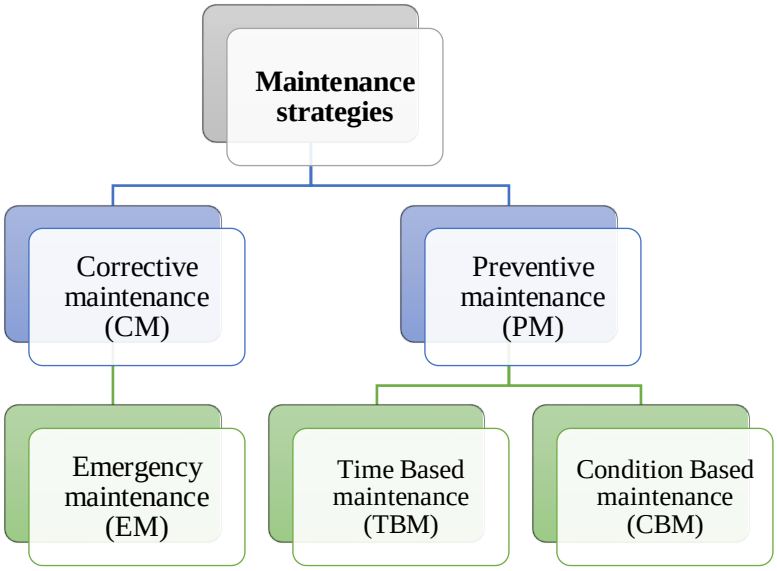


Figure 1.2 Main types of maintenance strategies

Corrective maintenance (CM) is described as a strategy performed after failure happens. The causes’ faults are identified, isolated and rectified to restore the system to a state that can perform its required functions normally (Kothamasu, Huang, & VerDuin, 2006). CM is a running to failure strategy that is applied once the failure occurs to restore the system to its recommended performance. It can be efficient when the failure’s consequences are limited and do not need urgent repairs (Polotski, Kenne, & Gharbi, 2019). On the other hand, preventive maintenance (PM) is described as a strategy that performs maintenance tasks, such as inspection and serving at predetermined intervals, or by monitoring the system performance to minimize the future system’s failure probabilities and to improve its degraded performance (Kothamasu et al., 2006). The key difference between the two main strategies is that CM is a reaction strategy that requires the failure to occur first, besides its unscheduled strategy, whereas PM aims to prevent the failure before it happens through a scheduled plan (Velmurugan & Dhingra, 2015).

Emergency maintenance (EM) is one of the more well-known CM strategies, which is applied only in emergency situations due to unexpected fault. It is not preferred to use the EM frequently, since it is usually more expensive than the normal maintenance. In addition, it leads to a longer system outage and dramatic impact on production (Mesenzhnik, Prut, Gnedin, & Bugrova, 1990).

PM subtypes are divided into two main categories: time-based maintenance (TBM) and condition-based maintenance (CBM). TBM aims to restore the system's reliability at fixed time intervals to prevent the occurrence of failure by predicting the mean time between failure (MTBF) (de Jonge, B., Dijkstra, & Romeijnders, 2015). It assumes that failures are mainly age related, which is an assumption that is too simple for the complexity of present-day failures. On the other hand, CBM considers the faults to be warning indicators for abnormalities in the system process that could lead to a failure. Therefore, it monitors the system performance by looking for evidence related to the occurrence of faults, in order to recommend a maintenance action to treat those faults (Kumar, S., Goyal, Dang, Dhami, & Pabla, 2018).

Deciding between CM and PM strategies is achieved based on the system failure rate, which has different behaviors over the system's life time, and requires different maintenance strategies to adapt to changes over time. Figure 1.3 depicts the failure rate trend versus the lifetime of the industrial system, which is known as the bathtub curve.

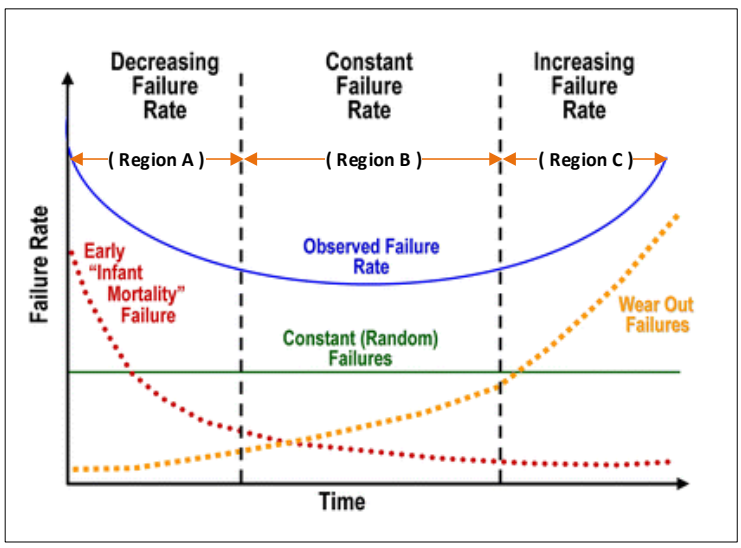


Figure 1.3 Typical bathtub curve (Suhir, 2015)

It has been observed that the failure rate trend can be divided into three main patterns, which defines three categories of the failure: early infant mortality (Region A), constant (Region B) and wear-out failures (Region C).

- **Region A** is characterized by an initial decreasing failure rate that happens due to the lack in design and lack of adequate controls. The failures can be eliminated through corrective

maintenance, since the intervention actions are only limited to correcting the limitations in system design.

- **Region B** is characterized by the lowest constant failure rate, which is considered to be the useful lifetime of the system. The failures mainly happen due to chance, after redesigning and configuring the system well. They can be eliminated through corrective maintenance strategy.
- **Region C** is characterized by an increase in failure rate as a result of system deterioration due to wear-out. The increase in failures can be held up and the system lifetime can be extended via a preventive maintenance strategy that focuses on minimizing the possible failure occurrences (Jardine & Tsang, 2005).

  Wear-out is a degradation and deterioration in system elements due to an impact of the fault appearance and its development. It is a very critical stage in the system lifetime, as the deficiencies in its management lead to a complete system failure (Wilkins, 2002). Regarding system performance in region C, it can be categorized into two working states: the normal and degradation state. According to Ahmadi (2012), the system can deliver its main functions in both the normal and degradation states, as follows:

  - **The normal state** can be maintained if the system is kept at the beginning of region C. The system leads to fewer production defects and malfunctions. The objective of PM in that region is to maintain the system in good working condition as long as possible so that its failure rate still close to the lower failure rate of region B. This objective can be achieved by monitoring the system's health condition through the assigned thresholds that reflecting the system degradation state. The system will enter into a degradation state if the conditions exceed those thresholds.
  - **The degradation state** occurs when the system enters deeply in the wear-out region. Consequently, the PM performs a set of maintenance actions by overpassing this threshold to return the system to its normal state.

The best PM strategy for region C is CBM, since maintenance actions are only performed when evidence related to system deterioration is detected due to the occurrence of faults. The CBM strategy performs more than the other PM strategies because it ensures the system's

reliability and the safety of the system. In addition, it minimizes the overall cost by continually monitoring the system's health condition and the degradation state.

Thanks to the great development of wireless sensors (!!! INVALID CITATION !!!), engineers can easily verify the system health conditions to periodically measure a huge quantity of data and sends them to a data center (DC). With the support of this collected data in DC, the knowledge of the system's health state is extracted and the implementation of CBM provides efficient, continuous system monitoring. In addition, sustaining the system's reliability through CBM is more effective by considering the current system state rather than performing repairs at calculated elapsed times, as in TBM. Moreover, the system's complexity impacts the accurate estimation of MTBF in TBM, based on the failure distribution for precisely defining the predetermined maintenance intervals (Lai, Jiang, & Jackson, 2019). In the next section, the CBM steps that manage the faults in region C (wear-out) are discussed.

1.3 Condition Based Maintenance (CBM)

Nowadays, a huge shift in maintenance strategies has been directed towards CBM due to its adequate solutions for scheduled preventive actions. These actions optimize the system's availability against its degradation as long as possible (Liu, Yunpeng, Xu, Li, Xia, & Gao, 2019). The CBM combines different methods and tools that go through procedures in order to distinguish the fault events for triggering preventive actions (Tahan, Tsoutsanis, Muhammad, & Abdul Karim, 2017). Briefly, the CBM procedures are done through perceiving, first the fault event, then understanding and anticipating its future manifestation by assigning preventive actions accordingly (Bianchini, Rossi, & Antipodi, 2018). The success of CBM tasks are centred on identifying and controlling the fault root-causes that affect the fault evolution over time.

The fault diagnosis and prognosis are the two main pillars of a CBM strategy. They are analytical procedures that provide expert with comprehensive knowledge to understand, identify and anticipate the fault's behavior. Fault diagnosis is a causality analysis process that involves the system's historical failures, which are deeply analyzed through fault detection, isolation and identification steps, as follows:

- **Fault Detection:** The fault detection step is a task that recognizes the fault's occurrence. This is realized by determining and verifying specific system indicators related to the fault inside the system process (Habibi, Howard, & Simani, 2019).
- **Fault Isolation:** The fault isolation step is a task that separates the faulty internal event from the normal internal events of the system. This isolation is done by assigning a set of particular causes related to the faulty event occurrence (Javed, Chen, Farrag, & Xu, 2019).
- **Fault Identification:** The fault identification focuses on how the fault impacts the system by understanding the fault causality structure and its cause-effect relationship (Kordes, Wurm, Hozhabrpour, & Wismuller, 2018).

Hence, the fault diagnosis aims to provide essential causality knowledge about the fault event occurrence at a certain period over the system's life. However, the development of a future fault and its impact on the system are still missing. Since complex industrial systems are subject to deterioration, those events are subject to developing over the period as the system ages. Consequently, the limitation of the fault diagnosis is complemented by the fault prognosis for understanding the future behavior of the fault. The fault prognosis is an anticipatory analysis that predicts the future development behavior of the fault event. It uses available knowledge about the system to forecast future fault behaviour (Jiang, Wu, Lu, & Mao, 2018).

Both a fault's diagnosis and prognosis are combined to mange the fault through the CBM strategy. They provide essential knowledge extracted from the previous system's evidence of faults and are treated as posterior comprehension knowledge. This knowledge anticipates the manifestation of a future fault and forms the decision-making core of the CBM strategy by deploying preventive actions. This extracted knowledge is preserved and captured in the CBM by the fault diagnosis and prognosis embedded models. Recognizing and analyzing those events are executed by continually monitoring the state of the system health conditions. The system's stream data is verified by those models to identify the faulty events and then the related decision actions are taken accordingly. As depicted in figure 1.4, the CBM strategy adapts three main steps to achieve this objective (Jardine, Lin, & Banjevic, 2006).

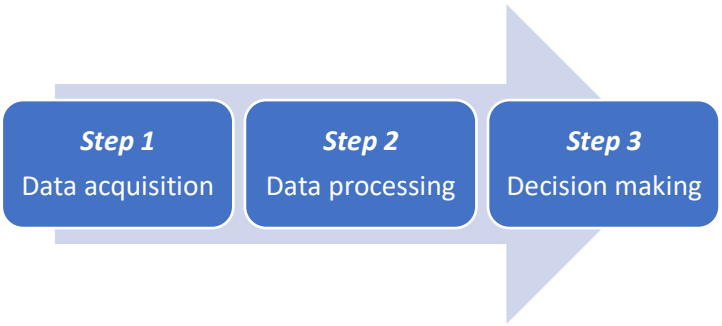


Figure 1.4 Main steps of CBM

Step 1 - Data acquisition: Data is organized and collected from the system's operational control histories and its measured performance sensors network is distributed over the system processes to explain its general state.

Step 2 - Data processing: This consists of two tasks: signal processing and data cleaning. Signal processing is more related to the sensors' signal outputs, where signal pre-processing techniques and feature extraction are applied to preserve only useful knowledge. Data cleaning is applied to all of the collected data, and outliers' observations are verified and classified into meaningful and noisy observations. In addition, this task removes redundant variables and deals with the missing values in data observations (Loukopoulos et al., 2017).

Step 3 - Decision making: The system's faulty states are verified through the processed data by the fault diagnosis and prognosis of embedded models. Those models are trained to recognize the deviation in the system's operation states outside its recommended limitations. According to the model's description and analysis results, sets of preventive and intrusive actions are performed to treat the causes of the fault and control its future consequences (Manling et al., 2019). The fault diagnosis and prognosis of embedded models in CBM can be built and trained based on two different approaches: model-based and data-driven.

The model-based approach is built based on the physics of the system. It is an analytical model that captures the relation between inputs and outputs of the system. Moreover, it can be developed from a set of equations or graphical causal models such as diagrams and trees. It is constructed based on the expert's experiences. The model's represented knowledge is mainly influenced by experts' understanding of the system (Luo et al., 2005).

As reported in (Ayyub, 2014; Baig, Ruzli, & Buang, 2013), the fault tree analysis (FTA) is one of the more well-known model-based approaches that is commonly used in risk management and reliability analysis. The FTA provides many advantages. First, it is easy to understand and interpret. Second, it is a flexible approach to represent the fault hierarchical structure. Third, the FTA is a probability-based method that depicts the probability in a bottom-up approach, starting from basic and intermediate event probabilities, reaching up to the probability of the top event. These probabilities provide the decision makers with a quantitative estimation of how the cause events are combined together to influence the probability of the top faulty event's occurrence.

On the other hand, **the data-driven approach** is built based on the system's historical data in order to discover the relations and dependences of the system elements on each other. It uses machine learning and pattern recognition techniques to extract knowledge that map the data measurements with the fault event occurrence. It is self-constructed in models where an expert's intervention is limited to validation and improvement, unlike the model-based approaches in which the expert will need to be involved during the construction process (Yuchen, Shen, & Kaynak, 2018).

Granger causality is one of the well-known causality analysis techniques for fault diagnosis (Alizadeh, E., El Koujok, M., Ragab, A., & Amazouz, M., 2018; Ntalampiras, 2018). It describes the statistical dependencies between variables depicted with arcs in a graph. The graph describes the interaction between system variables and information flow. This causal relationship between the system processes measurements is very useful to understand the fault occurrence and determine its related root-causes. Moreover, the represented knowledge is unbiased and an essential advantage for complex systems when the expert's knowledge has become too limited to understand the system's complex structure.

1.4 Problem statement

Significant development had been made in the fault diagnosis and prognosis domain that has helped the experts to build a reliable model that can be implemented and form the core of the CBM decision-making process. Model-based approaches are able to represent the fault causality knowledge in an interpretable manner, which facilitates the decision-making process.

However, building a model by the expert with an accurate understanding of representing complex fault events with its related causes is a challenging task. In addition, the construction of such a

model is a tedious and a time-consuming task, as long as the complexity of the system increases. Such a task requires the involvement of expertise from different domains. With a simple system, the faults have a straightforward relationship with the associated causes and building its model is relatively accurate.

On the other hand, the graphical data-driven approach is able to extract useful and unbiased knowledge easily from complex systems. It summarizes the huge amount of data in a useful predictive model or meaningful knowledge. The data-driven graphical models attempt to represent the fault's causality knowledge with a graph or tree to describe the interaction between system variables. However, those models lack the decomposition capability of the model-based to represent the fault into hierarchical causes. Therefore, those models are relatively accurate in predicting the fault's quantification and only represents its initial causes. Table 1.1 summarizes the advantages and limitations for the both model-based and data-driven approaches.

Table 1.1 Model-based and data-driven advantages and limitations

	Model-based approach	**Data-driven approach**
Advantages	Easily demonstrating the fault causality structure.	Automatic constructed models that extract unbiased causality knowledge from the data.
Limitations	Limited in complex systems and could reflect biased knowledge.	Missing the expert logic to easily demonstrating the causality structure.

Accordingly, this thesis addresses the following two major challenges:

1. In complex systems, building a model based on the experts is very complex, as it is hard to depict the fault causality structure that finds the relationship between the main event, the intermediate events and the root-causes. Therefore, a new methodology that is able to automatically construct the models for complex systems has to be developed to take advantage of the interpretability of the model-based approach.

2. In contrast, the current data-driven approaches lack the capability to interpret the industrial faults problems, discover dependencies and build the causality structure, which represent crucial links between the preserved knowledge in the data and the expert's understanding. Therefore, the need to develop a new methodology is strongly recommended, to unlock the data value in an appropriate, interpretable manner to represent and discover the hidden fault causality from the data.

1.5 Main objective and contributions of the thesis

This thesis aims to propose an approach for building a graphical data-driven fault tree model that links the model-based with the data-driven approaches. The novelty of this thesis is to use the FTA logic for decomposing the fault to its causality structure as a graphical interface that represents the automatically discovered fault knowledge by the data-driven patterns. Therefore, the proposed methodology introduces an automatic construction approach for a model that is similar to FTA, but builds directly from the data. This new, hybrid version of a model will be able to overcome the limitations of the data-driven and model-based approaches to maximize the diagnosis and prognosis of a fault in a complex system.

The construction procedures start by extracting the patterns from the dataset and connecting them using Boolean logic gates to model and describe the main fault causality. The patterns that represent the different fault causality events are extracted automatically from the system's database. Consequently, the expert's role is only to validate or enrich the obtained model.

In this thesis, the proposed approach aims to build models that are able to well diagnose and prognose the fault through achieving a set of objectives. The easiness of model construction and the easiness of demonstrating the fault hierarchical causality structure for the fault diagnosis task. While, modeling the degradation over the fault causality structure and modeling the changes in that causality structure over time for the fault prognosis task.

This thesis proposes three contributions toward understanding the problem statement and achieving the main objectives noted above. Three scientific papers were prepared and submitted in the Expert Systems with Applications journal, where one of the papers has already been published and the two others are under review.

Contribution 1. One-level interpretable logic tree analysis model (ILTA-model)

Contribution 1 develops the one-level ILTA-model, which is able to construct a one level tree that diagnoses the root-causes of simple systems. This paper proposes an automatic approach for constructing the fault tree based on the extracted knowledge from the system database. It provides an estimation of fault occurrence with a set of control rules for managing the consequences of the fault occurrence. The model's performance is validated based on simulated data from a simple actuator system.

Contribution 2. Multi-level interpretable logic tree analysis model (MILTA-model)

Contribution 2 develops the MILTA-model that addresses the fault diagnosis challenges in complex systems. The model improves the ILTA-model to address the fault diagnosis where the causality structure between the root-causes, intermediate causes and the fault event is complex. The model construction and exploitation are validated using a complex fault of a real actuator installed in sugar production processes.

Contribution 3. Interpretable time causality analysis model (ITCA-model)

Contribution 3 develops the ITCA-model that addresses the fault prognosis challenges in complex systems over time. The ITCA-model considers the effect of the system aging on the changing of the fault causality structure at different periods of time. The model integrates the results of the ILTA and MILTA models over the system's life. The model performance is validated based on the NASA dataset for turbofan engine performance degradation. Table 1.2 summarizes and links the thesis objectives with contributions.

Table 1.2 Thesis contributions

<table>
<tr><th></th><th colspan="2">Fault diagnosis</th><th colspan="2">Fault prognosis</th></tr>
<tr><td>Objectives</td><td>Easiness of model construction</td><td>Easiness of demonstrating the fault hierarchical causality structure</td><td>Modelling the degradation over the fault causality structure</td><td>Modelling the changing in the causality structure over the time</td></tr>
<tr><td rowspan="3">Model name</td><td>ILTA model</td><td></td><td></td><td></td></tr>
<tr><td colspan="2">MILTA model</td><td></td><td></td></tr>
<tr><td colspan="4">ITCA model</td></tr>
<tr><td>Contribution</td><td>Construct the FTA directly from the data with the minimal expert involvements.</td><td>The fault decomposition process is achieved with the same manner as the expert did.</td><td>The degradation and its effect on the fault causality structure is automatically captured over time.</td><td>The changing in the fault causality structure over time are graphically represented in one model.</td></tr>
</table>

1.6 Thesis organization

This thesis is divided into seven chapters. Chapter 2 reviews the methods and techniques of model-based and data-driven approaches for fault diagnosis and prognosis. It focuses on recent related developments and highlights the research gaps. Chapter 3 provides an overview of the proposed methodology that addresses the three aforementioned objectives, as well as the common thread between the three models. Chapter 4 presents the first contribution of the thesis, the ILTA-model construction, validation and exploitation. Chapter 5 presents the second contribution of the thesis, the MILTA-model for fault diagnosis in complex systems. Chapter 6 presents the third contribution of the thesis, the ITCA-model, which provides a time causality analysis of a complex fault. Finally, Chapter 7 and 8 provides a general discussion and the conclusions about the three models proposed, followed by future research.

CHAPTER 2 CRITICAL LITERATURE REVIEW

A fault is an internal system event that concerns one or more parts of a system, which results in the system response deviating from its expected behavior. Consequently, a fault produces performance drifts from the expected outputs such as defective products, low quality and service outage reaching to accidents.

In this chapter, the current fault diagnosis and prognosis approaches and models are investigated to manage the fault risk and its consequences in complex industrial systems. Section 2.1 depicts the fault diagnosis and prognosis procedures and the main challenges of their application in real situations. Section 2.2 reviews the data-driven and model-based approaches in fault diagnosis and prognosis domains. The main differences between these two main approaches are highlighted and the data exploitation to build their models is discussed. Sections 2.3 and 2.4 focus on the literature review of fault diagnosis and prognosis using the data-driven models. Similarly, sections 2.5 and 2.6 address the fault diagnosis and prognosis model-based models. Finally, Section 2.7 summarizes the limitations and weaknesses in the different fault diagnosis and prognosis models, leading to the conclusion about the research motivation for this thesis.

2.1 Fault diagnosis and prognosis approaches

Fault events may occur due to internal or external causes. External causes could be harsh and abnormal operating conditions or factors of a working environment, such as extreme temperature and humidity. Whereas the internal causes could be, for instance, missing lubrication that leads to early wear-out (Isermann, 2006). Fault occurrence could be classified as an abrupt, incipient or intermittent fault. An abrupt fault appears suddenly and has a continuous effect on the system (e.g. connection cut off). While an incipient fault happens gradually (e.g. gradually increasing the wear-out of mechanical gear) and has a progressive effect on the system. Finally, an intermittent fault occurs and disappears frequently (e.g. a computer software fault).

Meanwhile, the fault impact on the system's performance could be classified into permanent and temporary. The permanent impact cannot disappear, while the temporary impact may be resolved with corrective action (Zolghadri, Henry, Cieslak, Efimov, & Goupil, 2014). The evolution of the faults events over time without interaction with the correction or prevention actions could lead to

a failure, which is a set of functions and objectives that the system failed to achieve them due to the fault occurrence and development (Hamill & Goseva-Popstojanova, 2009). Figure 2.1 depicts a flowchart for supervising and controlling the causes of a fault and its impact on the system according to (Isermann, 2006).

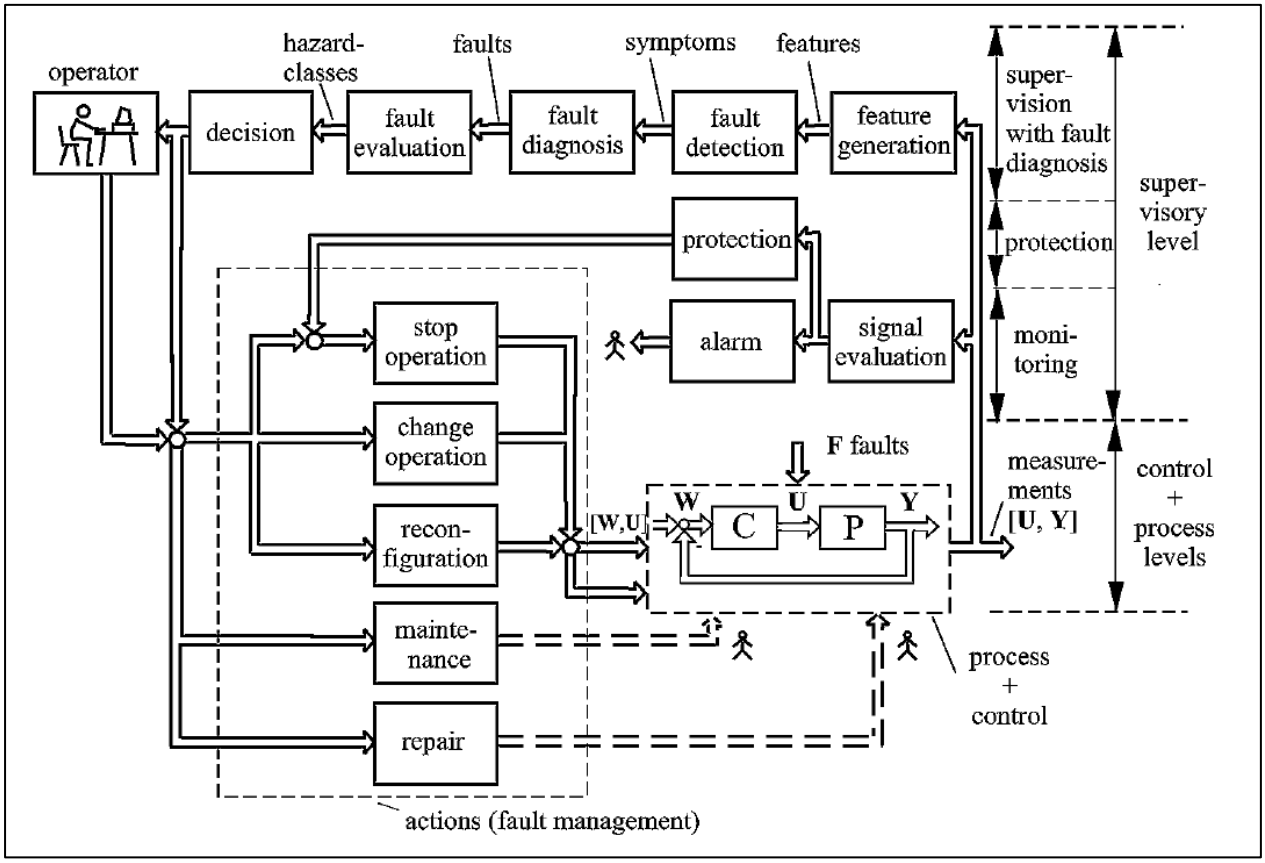


Figure 2.1 Supervision and control flowchart (Isermann, 2006)

With regards to monitoring and protection, only certain variables are selected from the system's internal processes that have a direct relation with the fault occurrence (fault indicators), such as the variables (U and Y), as depicted in Figure 2.1. The fault indicators are employed to define the fault tolerance threshold that will trigger the system's alarm to recall the operators (Amin & Hasan, 2019; Cui et al., 2018). A protection subsystem is automatically activated once those fault tolerance thresholds are violated (Rui & Lie, 2018).

However, monitoring and protection provides an expert with fault knowledge to indicate the fault occurrence without deeply investigating the fault corresponding causality. This is very superficial

knowledge to support the expert in understanding the fault's root causes and taking an appropriate action.

Supervision with CBM fault diagnosis and prognosis provides an expert with all of the required knowledge about the fault's characteristics, occurrence and behaviour. CBM helps propose more fault management actions. The fault indicators such as (U and V) with other included process variables are collected, since this level needs more explainable variables to provide deep fault analysis (Wurzel & Hasbroucq, 2015). The analysis result diagnoses the fault via its root causes (Omidi & Liu, 2018) and forecasts its future development and impacts on the system (Zhengxin, Xiaosheng, Changhua, & Yaguo, 2018). Based on an understanding of the current system's situation, an operator can take a set of preventive actions such as repairing, maintenance, reconfiguration, change operation or protective actions.

2.1.1 Fault diagnosis and prognosis methods

Fault management using CBM fault diagnosis and prognosis provides much better knowledge about the fault occurrence and fosters the expert's decisions taken through its deep knowledge representing the fault causality. The fault diagnosis and prognosis procedures in CBM include a sequential steps flowchart, as depicted in the flowchart in Figure 2.2 (Sikorska, Hodkiewicz, and Ma (2011). These procedures are divided into two main steps. The fault diagnosis step defines the affected components and the related causes for the fault's appearance. The fault prognosis step describes the evolution of the fault and its expected impact on a system's health condition. The prognosis step heavy relies on the diagnosis step output (e.g., the fault type, root causes and the affected components).

Fault diagnosis objectives are achieved through three sequential tasks: fault detection, fault isolation and fault identification (Liu, B., Ding, Wu, & Yao, 2019). Fault detection recognizes the fault's appearance by monitoring the system's operation conditions (Miljković, 2011). Fault isolation insolates the fault's occurrence in the system by determining the affected components and sub systems (Biswas, Dash, Choudhury, & Sahoo, 2018). The fault identification characterizes the fault's nature and mode, besides discovering its main root causes (Gao, Jie & Zhao, 2019).

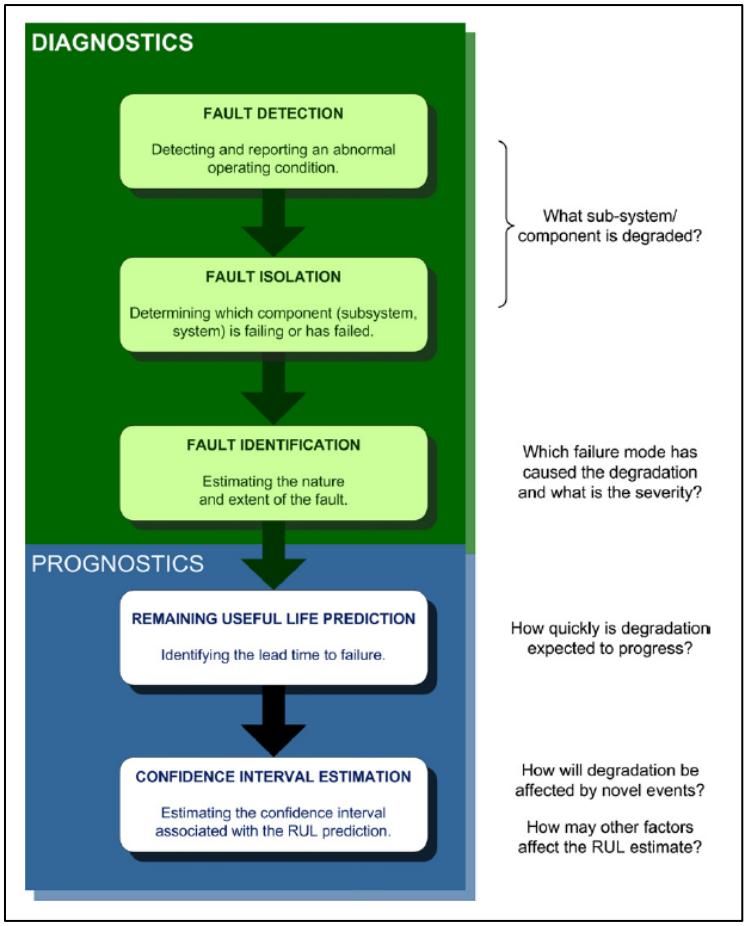


Figure 2.2 Fault diagnosis and prognosis flowchart (Sikorska et al., 2011)

Those three tasks can be performed through various methods in which an expert's involvement and the data utilisation can be different. One of the methods mainly relies on an expert's knowledge and his experience with the system's structure and the possible causes of fault occurrences. The expert builds a model, such as FTA, that reflects his understanding of how the fault may be detected, isolated and quantified (Srivastava & Sinha, 2012). However, by increasing the system's complexity, he can partially rely on the data by selecting certain variables to assign the trend check or the threshold range for more precise fault detection (Amin & Hasan, 2019). In addition, he can use fault pattern recognition methods such as a neural network or self-organizing map as a support tool to isolate and identify the evidence of the fault. On the other hand, the task of a diagnosis can be performed by mainly using the system's data, where expert systems extract knowledge from the data to isolate, identify and characterize the fault's occurrence (Guo, Y. et al., 2019), while the human expert's role is limited to only supervision.

After completing the fault diagnosis, the fault prognosis comes as a complemented anticipation analysis task to predict the diagnosis results. The future fault impact on the system behavior is investigated to understand how quickly the affected components and subsystems are degraded and progressed for complete failure over time (Sikorska et al., 2011). With the same analogy of the fault diagnosis task, the fault prognosis task can be achieved using various methods that involve specific expert roles and data exploration.

In a fault prognosis, the expert relies on his knowledge to build a mathematical model that describes the degradation phenomenon within the system and then uses such a model to estimate the remaining useful life (RUL) of the system. However, such a method is too limited for simple degradation phenomenon (Lin, Wen-Chiao & Ghoneim, Youssef A., 2016). Meanwhile, the expert can rely more on the data to determine the expected lifetime of some critical components or subsystem individually by using statistical models such as the proportional hazards model (PHM) (Ding & He, 2010). Moreover, the expert can train a neural network to automatically calculate the RUL of those components or subsystems (Wu, Q., Ding, & Huang, 2018). From another side, the human expert can only monitor expert or fuzzy systems that compare the similarity between the current fault situation and the historical system failure to explain the system RUL (Zarei, Shasadeghi, & Ramezani, 2014).

2.1.2 Fault diagnosis and prognosis in complex systems

Achieving CBM fault diagnosis and prognosis tasks in complex systems creates a new challenge that necessitates the proposal of appropriate solutions. A complex system can be seen as multiple interdependent subsystems that imply interrelated internal processes, which are very difficult to understand.

Figure 2.3 depicts two examples of Canada's oil production and the pipeline transportation systems. Gas and oil are extracted from the wellheads and travel through the gathering pipelines to gas processing or oil batteries. The crude oil and the liquid gas are processed and continue traveling through the feeder pipelines, which deliver them to the transmission pipelines, which is the main gateway to the pipeline consuming network. From the transmission pipelines, the oil can be refined or shipped, whereas the gas can be distributed to homes, businesses and various industries.

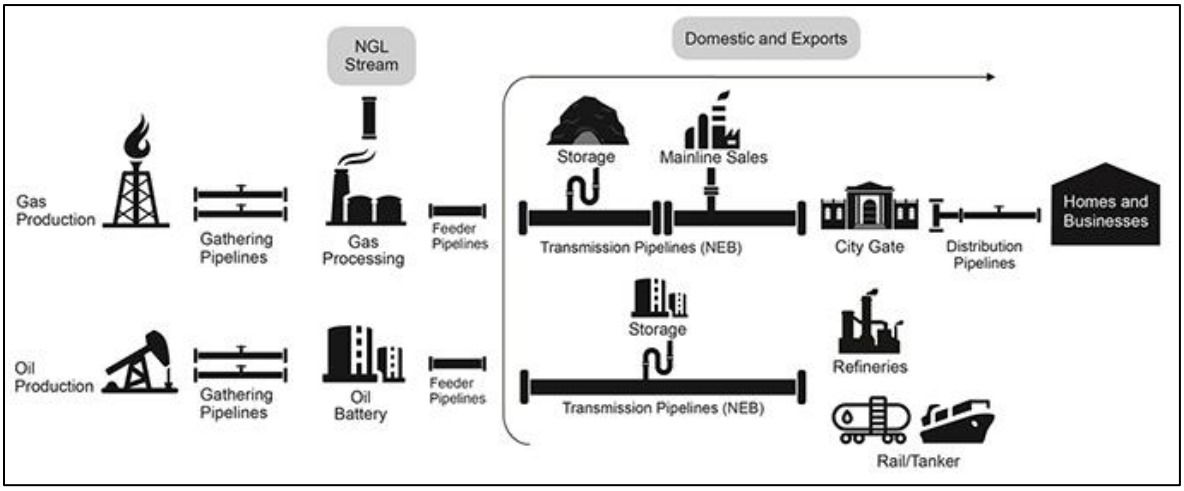


Figure 2.3 Oil production and pipeline system overview (Board, 2017)

When one of the fault causes (e.g. missing the lubrication) happens in an internal process of the explained system, it triggers the occurrence of a corresponding fault event. The fault can be detected by a change in pressure or vibrations. Thus, the instability of this internal process affects the output state of the external process, which can be measured with the change in mass flow. Depending on the significant role of this internal process, the system output can be affected, for example, with defection in the production output. Without understanding the fault causality structure and the interdependence between those internal processes, inappropriate maintenance actions can be taken. Therefore, the fault can progress and become a contagious event that effects other internal processes, which leads to a massive fault and service outage of the system (Isermann, 2006).

In complex systems, the faults and failures occur due to the joint appearance of multiple internal faults over the different interconnected internal processes. Meanwhile, the diagnosis and prognosis of those complex faults independently lead to a partial understanding of the failure causality. Hence, a successful fault diagnosis and prognosis in a complex system have to consider this joint causality knowledge that represents the processes' interdependency as well as the internal faults' influences. Therefore, fault diagnosis and prognosis models have to strongly rely on the collected data over the system's peripherals. Since the extracted knowledge from the data enables the discovery of hidden fault phenomena, this helps an expert grasp fault causality in an interpretable manner (Cook, 1998).

2.1.3 Industrial data for fault management

Previous research has focused on the use of the domain knowledge to build CBM fault diagnosis and prognosis embedded models. Those models offer several advantages such as simplicity, interpretability and ease in development (Duvvuri, 2019). With an increase in system complexity, such models are susceptible to disturbances in detecting, characterizing and anticipating the fault in those systems. The drawback of the expert's knowledge on complex systems underlines the limitation in understanding dependent processes, in addition to a bias analysis due to the partial understanding of the fault causality.

From another point of view, the increase in system automation, connectivity and machine intelligence have helped to exponentially grow the amount of collected data (Xu, Sun, Wan, Liu, & Song, 2017). Therefore, massive amounts of data have been employed to fit the gap and the limitations of the domain expert to describe complex systems. Thus, a shift in research orientation is happening; instead of completely relying on an expert to manage a fault, the collected data is used to extract hidden knowledge and phenomena on the fault occurrence. This will assist the expert with understanding the complex fault structure (Bingamil, Alsyouf, & Cheaitou, 2017). An emphasis on relying on data and its extracted knowledge for fault diagnosis and prognosis overcomes an expert's limited knowledge. Moreover, using data boosts the automatization and the intelligence of the system when managing the faults by CBM.

The collected system data can be grouped into two main classes: structured and unstructured data. The structured data is organized information that has a specific architecture according to the defined objectives that the data has been collected for. The highly arranged structure of the data enables analytics and expertise to easy extract knowledge and hidden phenomena related to the analysis objectives (Yuan, Zhang, & Duan, 2018). The unstructured data comes from leveraging different data sources that have different structures. It includes a huge diversity in data formats, such as image, text and numbers, which mainly caused by the system connectivity and its IOT over a system network. It represents the raw data source from which different structures of data with different objectives can be defined (Corcoglioniti, Rospocher, Cattoni, Magnini, & Serafini, 2018).

Most industrial data has a structured form, related to defined objectives and collected from different sources. Wang, J., Zhang, Shi, Duan, and Liu (2018) review the structured data types for different

objectives such as CBM, energy management, supply chain, customers and services. They have classified the collected data for CBM objectives into five categories:

(1) Environmental data, which mainly includes the working operation conditions such as temperature, humidity and topography.

(2) The task data, which contains the type of the performed task by the machine or internal process.

(3) The operation data, involving measurements and reading from sensors that are distributed over the different systems' internal processes in order to represent the current system state.

(4) The performance data, which are state data collected from sensors that are specially placed to capture a system's performance indicator measurements, such as production flow.

(5) The maintenance data, which are related to all of the maintenance activities and the preventive actions.

The five data categories are employed to create rational data through a data management framework (e.g. NoSQL) that connects different structured data based on their relations. This data includes valuable information and knowledge about the hidden degradation and the fault patterns that improve the fault diagnosis and prognosis models. For instance, the operations data that represent the current system operation conditions can be labelled with a state variable from the performance data to extract useful knowledge about the different system states.

The rational data improve its performance and availability in close loop, where the data are used to construct and improve the CBM fault models that accordingly decisions are taken. The system reaction regarding the taken decision is capture by the new collected data, which are reanalyzed by the CBM information system. Figure 2.4 depicts the architecture of the CBM information system and knowledge flow for the complex system.

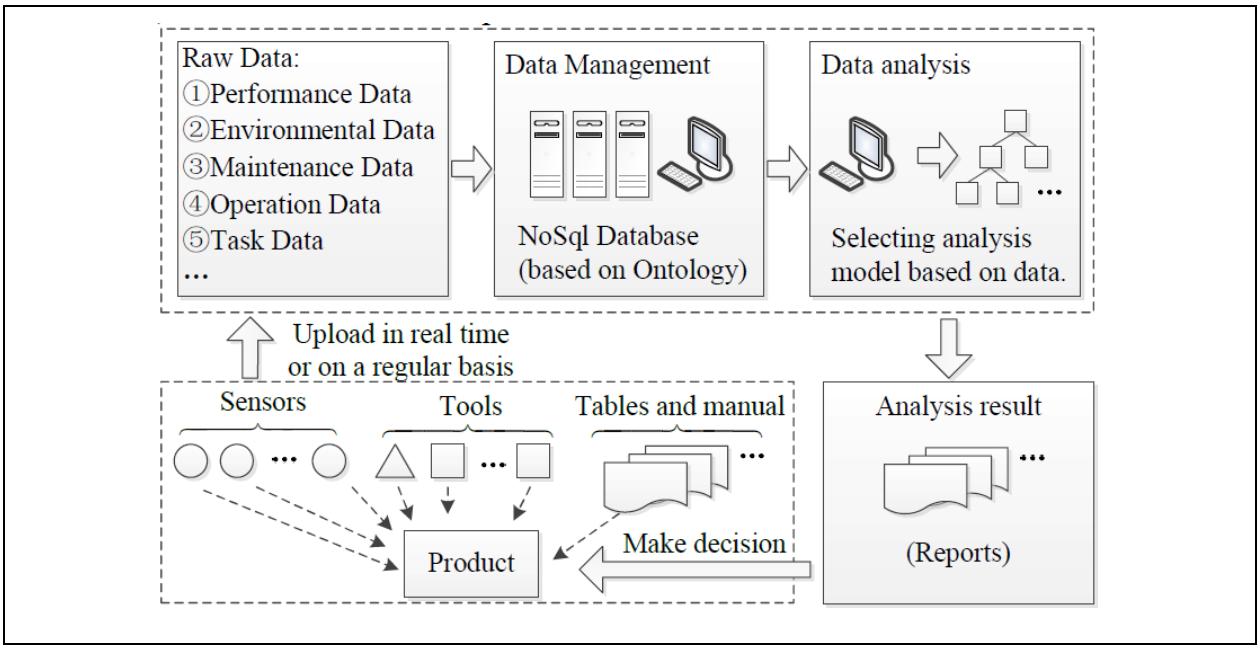


Figure 2.4 The architecture of CBM information system (Wang, L., Qian, Li, & Liu, 2017)

Accordingly, the five structured data categories are collected in the form of datasets that represent the system's current state. The datasets are employed as inputs of the data management system to perform a fault diagnosis and prognosis analysis. Thus, the hidden knowledge and phenomena about the fault causality and its impacts can be extracted to support a CBM decision maker to enhance the current system's state.

2.2 Fault diagnosis and prognosis models

Fault diagnosis and prognosis models for complex systems can rely on system data to better manage the consequences of a fault. This dependence on the data is utilized in two forms: historical and streaming data.

Historical data is utilized for constructing or improving the fault diagnosis and prognosis models in order to adapt with the system's complexity and its changes. On the other hand, streaming data is employed to verify and describe the current system's state and to validate the fault occurrence according to the constructed fault diagnosis and prognosis model (Kumar, A., Shankar, & Thakur, 2018).

Generally, constructing the fault diagnosis and prognosis models can be done using data-driven and model-based techniques. The data-driven techniques transform the large amount of historical

data into useful knowledge. The algorithms and statistical analysis explore and extract the hidden patterns, trends and behaviors from the data to form the model's knowledge for the fault diagnosis and prognosis (Md Nor, Che Hassan, & Hussain, 2018). On the other hand, the model-based techniques mainly use an expert's prior knowledge about the system's physics, its internal process structures and the dependency between them to build a mathematical formula or graphical models to diagnose and prognose the fault. The expert's knowledge, which is the cornerstone for those models, can be enriched by extracting the hidden phenomena from the system's historical observations to adapt to the system's complexity (Balasubramanian & Muthu, 2017). Figure 2.5 depicts common and different features between the two fault diagnosis and prognosis techniques.

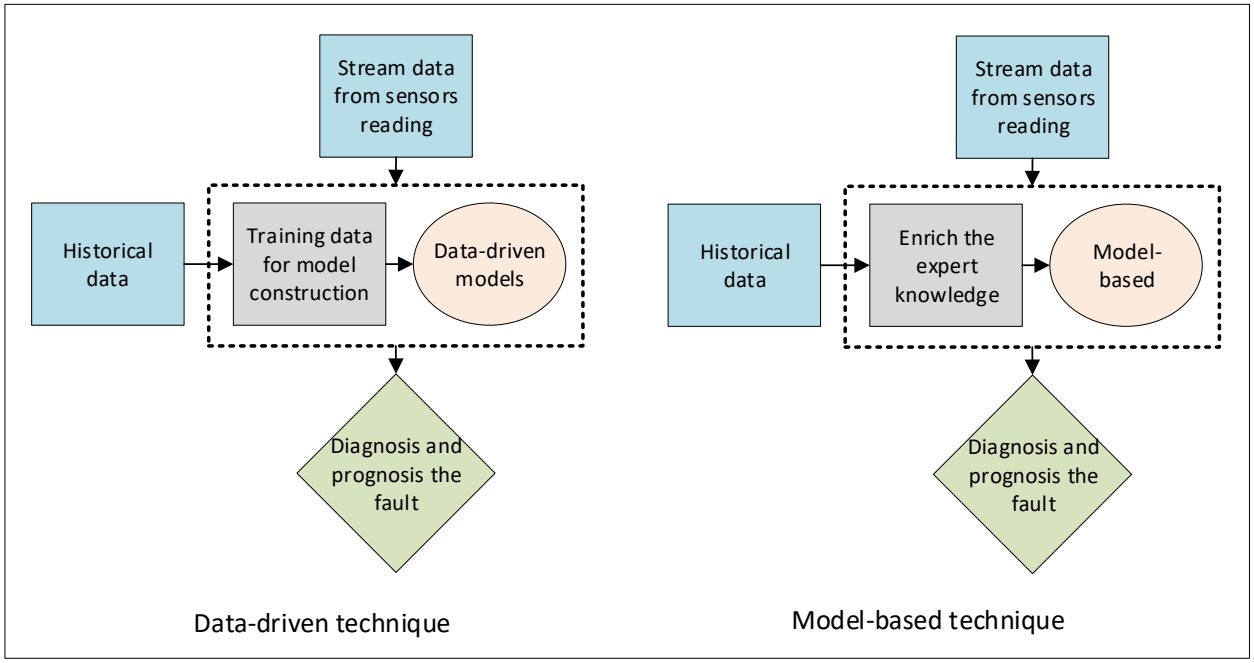


Figure 2.5 Data-driven and Model-based techniques

The historical data is mainly employed for constructing the data-driven models, while in the model-based, this historical data is used to enhance the expert's prior knowledge to improve the constructed models. Therefore, the historical data produces two different kinds of knowledge used: training data for model construction in data-driven techniques, and enriching the expert's knowledge in model-based techniques. Thus, the model processing, verification and validation have a direct and significant impact on CBM decision making.

From another point of view, constructing the data-driven models based on historical data does not need any modelling effort on the part of the expert. This is because the mapping process between the system measurements, with its different health states, is mainly done based on the system database (Md Nor et al., 2018). Consequently, these models provide unbiased knowledge and are very reliable in predicting and anticipating the fault in a complex system if the collected data samples are representative of the different system states (Tao & Alves, 2019).

In addition, interpretable knowledge can be extracted from the historical data, which adds explainability to those models, aside from their predictability power (Guo, Y. et al., 2019). This knowledge is able to explain certain fault causality structures and their dependence individually. However, they may miss the logic that explains the fault causality structure by decomposing the fault to its root-causes, in addition to the change of that structure over the system's life. Therefore, we may strongly rely on data-driven models to predict the fault states in a complex system.

On the other hand, historical data in the model-based techniques is used to assist an expert in building models that reflect his understanding about the system's physics that explain a fault occurrence. Those models are mainly constructed based on the expert's knowledge, which is able to provide an interpretable model capable of thoroughly explaining the fault's hierarchical structure. To overcome the expert's knowledge limitations in complex system, fault knowledge is extracted from historical data, which enriches the expert's prior knowledge on the complex system structure and its possible faults. The extracted knowledge contributes to discovering the fault's hidden phenomena, which could be overlooked by an expert due to the system's complexity (Badida, Balasubramaniam, & Jayaprakash, 2019).

When building a model-based model, the expert has to model his prior knowledge, which explains the essential fault causality, and then integrate the knowledge discovered from the data to improve the constructed model. However, due to the system's complexity, he might face challenges in combining this discovered knowledge in his constructed model. In this case, we can strongly rely on the model-based in understanding the fault causality structure for a subsystem or certain processes in a complex system.

Figure 2.6 depicts the different models for fault diagnosis and prognosis. The statistical and artificial intelligence algorithms are used to build the variant fault data-driven models, while quantitative and qualitative analysis is employed to build variant fault model-based models. In the

next section, recent achievements in both techniques will be discussed, and their models' advantages and limitations will be highlighted.

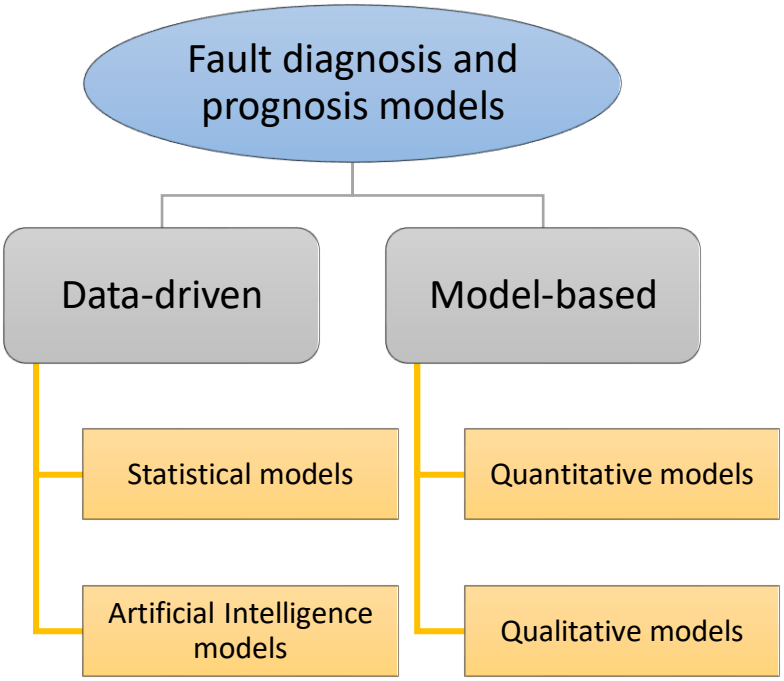


Figure 2.6 Fault diagnosis and prognosis models' tree

2.3 Data-driven models for fault diagnosis

Several data-driven techniques are proposed to achieve the three fault diagnosis steps, which can be classified into statistical and artificial intelligence models.

2.3.1 Statistical models

Statistical fault diagnosis models are mainly deployed for fault monitoring based on statistical analysis. The models can be constructed using univariate and multivariate statistical analysis.

The univariate analysis evaluates the related variables in the fault occurrence, which are commonly known as fault indicators, to discover specific trends related to the fault occurrence and to assign the control limits to identify the fault event. The Shewhart control charts, the cumulative sum (CUSUM) control charts and the exponentially weighted moving average (EWMA) control charts

are the most common univariate analyses for statistical process control (SPC) (Alauddin, Khan, Imtiaz, & Ahmed, 2018; Vera do Carmo, Lopes, & Souza, 2004).

To address the complex dependency of a system's internal processes, a multivariate analysis considers the cumulative effects of the data variables that dominate their dependency and correlation for diagnosis. Principal components analysis (PCA) and Partial Least Squares (PLS) are the most well-known multivariate analysis methods for fault diagnosis (Qin, 2009). The main advantage that supports their implementation concerns their ability to handle highly correlated data, which includes error measurements and missing values. Those algorithms transform the original data into another, lower dimension domain, and they do not require defined inputs and outputs data (Alauddin et al., 2018).

The PCA algorithm is mainly applied in fault detection and isolation in complex systems. Bakdi, and Kouadri (2017) addressed the problem of a false alarm in a complex system by fault monitoring a PCA. The assigned PCA threshold at a fixed significant level needs to be more flexible when adapting dynamics in complex system. Thus, an adaptive threshold based on an exponentially weighted moving average (EWMA) control chart statistic is developed to reinforce PCA performance. Zhong, Dong, and Ye (2018) combined PCA with a BP neural network to build a robust fault diagnosis model that is less sensitive to the noise. The extracted principal components (PCs) are used to train the BP neural network instead of the original data to mitigate the noise sensitivity problem.

A major drawback on PCA is that it cannot work with non-gaussian and non-linear data, which limits its application, especially in complex systems. This problem is addressed by using kernel methods such as kernel principal component analysis (KPCA) (Lawrence, 2005; Liang & Lee, 2013). Similar to PCA, PLS suffers from the non-gaussian and dynamic problems. Accordingly, Rui, and Lie (2018) introduced a new statistical approach based on kernel partial least squares for improving the fault detection process. The data first mapped into a feature space to overcome the non-gaussian challenge, then two statistical methods are used to monitor the fault's appearance. Zhang, Yingwei, Zhou, Qin, and Chai (2010) proposed a fault diagnosis framework based on the multiblock kernel partial least squares (MBK PLS). The contribution of the MBK PLS is able to present non-linear interpretation for fault diagnosis, aside from fault monitoring, by dividing the system variables into several significant blocks.

2.3.2 Artificial Intelligence models

Using artificial intelligence, the data-driven models are able to achieve efficient fault diagnosis tasks compared to the statistical models by including optimization and statistical learning. Regarding the fault detection task, machine learning classification algorithms are used. He, D., Li, and Zhu (2012) employed the k nearest neighbor in the plastic bearing fault detection, in which first the frequency and time domain features are extracted by the envelope analysis and EMD, and then the frequency domain features are used to identify bearing outer race faults, and time domain features are used to build a k-NN classifier to identify other types of bearing faults. Gul, Imran, and Khan (2018) applied the support vector machine (SVM) to the vibration signals to detect and classify faults in rotating machines. To address the challenge of changes in data over time due to system wear out, the authors use an incremental approach to the SVM.

Regarding the fault isolation task, the interpretable knowledge (if-then rules) are used to isolate the fault from the system processes. Mortada, M. A., Yacout, S., and Lakis, A. (2014) apply the logical analysis of data (LAD) algorithm on the power transformers' historical data to distinguish the difference between the normal and fault operations modes by extracting a set of patterns that characterize each mode. On the other hand, Yang, and Yu (2019) integrate the rough sets theory (RST) with the neural network (NN) to simplify the complexity of an NN structure in the fault diagnosis task. The RST preprocesses the data by removing its redundancy, then uses it to build the NN model.

Regarding the fault identification task, graphical models can be constructed to understand the possible fault causality and the cause and effect relationships between the variables of the system. Han, He, Zheng, and Wang (2019) introduce hybrid approach based on a three-layer Bayesian intelligent fault inference model (BIFIM) for inverters, in which the first layer includes the main fault event, the second layer expresses the possible symptoms of the fault, and the final layer contains the possible causes, which are enriched by the expert's domain knowledge. Abid, Sallem, and Braham (2019) integrate Optimized Stationary Wavelet Packet Transform (Op-SWPT) with the decision tree (DT) to diagnose the bearing faults. The data is first transformed by Op-SWPT to relevant knowledge, then is used to build a DT model that represents the fault's relevant knowledge in an interpretable graphical model.

2.4 Data-driven models for fault prognosis

The majority of the data-driven techniques for the fault prognosis task focus on the prediction of the remaining useful time of the system's life, using both statistical and artificial intelligence models.

2.4.1 Statistical models

Quantifying the system's performance using the reliability knowledge information using time-to-failure data is one of the most common statistical applied techniques in the industry to prognose a fault occurrence. These techniques are able to work with censored and small data sets. Based on the probability density function for a group of identical components, the occurrence of the system failure can be predicted. Several parametric and non-parametric functions are used to build the density function. Zhicai, Dongfeng, and Xinfa (2014) introduce a three-step methodology to predict the RUL of equipment using the Weibull Proportional Hazards Model (WPHM) to adjust the state parameters to the data and the regression model to fit the failure rate curve and then to predict the RUL of the equipment. Sun et al. (2019) aggregate the electric and mechanical failures for describing the failures in a CNC machine. First, the Weibull distribution parameters are estimated based on the maximum likelihood method, which combines both failure types. Then, the Pearson correlation coefficient is used to determine the dependency of the failure types on the CNC failure.

Forecasting future system deterioration based on historical data is another alternative for prognosis of the fault. The Autoregressive integrated moving average (ARIMA) is a very common technique for capturing development trends in time series data. It is able to deal with temporal trends and autocorrelation through the following three recursive steps: model identification, parameter estimation and model validation (Mehrmolaei & Keyvanpour, 2016). Yi, Yun, Chuan, and Yan-Ni (2018) apply ARIMA to forecast the fault trend in gearbox using nonlinear and non-stationer data. Meanwhile, the obtained fault trend through the ARIMA model is validated by comparing the extracted knowledge with the outlet pressure of the gearbox oil pump. Yuhensky, Munadi, and Hafiddudin (2016) proposed a methodology that models the fault formulation in customer premise equipment (CPE) segment on broadband network system. The ARIMA is selected thanks to its lower cumulative mean square error (CMSE) that is able to forecast the fault amounts per type of disturbances in the broadband network segment. Bezerra Viana, Sandoval Goes, and Conceicao

Rocha (2013) proposed a hybrid approach for the fault prognosis based on time series indicative parameter that denote the equipment condition. The fault development is divided into different scenarios. The ARIMA model is constructed for each valid scenario to provide a fault prognosis model.

Proportional Hazards Modelling (PHM) is another model used to prognose the fault. PHM extracts a set of covariates that express the contribution of the model covariates on the equipment's lifetime. It considers the multiplicative relationship in the system as the product of a linear function and the baseline hazard rate. The linear function normally refers to the operation conditions or environment (Kumar, D. & Klefsjo, 1994). Guo, C.-x., He, Zhang, Lu, and Yang (2014) use the PHM to characterize the faults in an oil-immersed transformer. The model focuses on temperature-based aging, which denotes the deterioration of classifiable health status. The obtained comprehensive knowledge from the model demonstrates thermal aging and its effect on a system's health. Jian, Huifang, Dongyang, and Benteng (2016) proposed a stratified proportional hazard model (PHM) that maximizes the equipment service age through extracting a set of covariances. The extracted covariances are employed then to categorize the equipment life cycles into multitype recurrent events. Xiaochuan, Fang, Bennett, and Mba (2018) identify the fault root-causes and the expected lifetime for reciprocating compressors in oil and gas industries. The proposed methodology integrates canonical variate analysis (CVA), cox proportional hazard (CPHM) and support vector regression (SVR) models to determine the fault features importance and forecasting the equipment remaining useful time.

2.4.2 Artificial Intelligence models

Expert system (ES) is a program or a set of programs that imitate human logic to manage fault diagnosis and prognosis. The ES decision consists of a set of rules that produce a set of particular outputs from the delivered inputs. Acquiring these rules requires the involvement of one or more experts over the years in order to develop a solid and adaptable set of rules that represent the system's behaviour (Chidaravalli, Jenila Livingston, & Manjunath, 2017). However, due to system complexity, human experts' knowledge is replaced with fuzzy logic to overcome that problem (Debnath, Reddy, Jagadish, & Das, 2019). Garga et al. (2001) propose a hybrid reasoning approach for an industrial gear box to predict the RUL. The approach integrates the domain knowledge with fuzzy logic and neural network to introduce an automated reasoning method. Mahdaoui et al.

(2019) propose a temporal neuro-fuzzy system (TNFS) to predict the RUL in preheater cement cyclones plant. The core of the TNFS is a set of temporal fuzzy rules that interpret the fault causality and fit the prognosis task. The time factor is added to the rules to capture the dynamics of the process and estimate the RUL. Boukra (2015) address the challenge of selecting relevant features in the fault prognosis task and to estimate the RUL. The proposed method uses the Particle Swarm Optimization algorithm to select the relevant features and particle filtering for forecasting the RUL, which is later integrated to the Neuro-Fuzzy System.

Artificial Neural Networks (ANN) is a very efficient and effective technique for both fault diagnosis and prognosis. It is able to deal with different input types of data to train the ANN network (Chen, Y. et al., 2019). Kui, Laghrouche, and Djerdir (2018) build a predictive model that predicts the degradation of the proton exchange membrane fuel cell (PEMFC) and estimates its RUL. The back propagation neural network model has 4 neural layers with 2 hidden layers. Moreover, the model highlights the important variables for describing the system degradation such as stack current, stack temperature, air pressure, hydrogen pressure and air humidity. Zangenehmadar, and Moselhi (2016) estimate the deterioration rates in water distribution networks based on predicting the RUL in the pipeline. ANN model is constructed based on Levenberg-Marquardt backpropagation algorithm. Accordingly, the model identifies the pipeline age, condition, length, diameter, material, and breakage rate as the contributed variables on the pipelines' degradation.

Deep learning is one of the most recent machine learning techniques that shows great performance in overcoming the deficiencies in current data-driven models due to its state-of-the-art accuracy (Rengasamy, Morvan, & Figueredo, 2018). Jin, W. (2016) addresses the challenge of a lack of complete lifecycle data in predicting the RUL based on Accelerated Degradation Tests (ADT). The model improves the accuracy of the RUL, since the degradation trends have become evidence at the end of the system's life. The feature enhancement is based on the Restricted Boltzmann Machine (RBM) method and similarity-based method to predict the RUL. Cheng Geng, Xiang Yu, Hong Zhong, and Yan Feng (2019) address the effect of the interference and noise in operational conditions' data to predict the RUL. The proposed prognostic method is based on Bi-Directional Long Short-Term Memory (BLSTM) network. The operation data is first preprocessed to sequences of data with a fixed length, and then it feeds the BLSTM. Jichao, Zhenpo, and Yongtao (2019) proposed a predictive maintenance framework based on mechanical equipment and

recurrent neural network (RNN). The RNN uses the long short-term memory (LSTM) to forecast the faults in air booster compressor (ABC) motor and suggests a set of predictive maintenance actions. Jichao et al. (2019) proposed a methodology for predicting the battery system states in electric vehicles. The battery voltage is selected to be the main characterisation parameters for forecasting the battery faults. Long short-term memory(LSTM) recurrent neural network model is built to anticipate the fault and predict the possible battery states over time. The RNN model is constructed based on real data collected from electric taxi belonging to the Service and Management Center for Electric Vehicles (SMC-EV) in Beijing.

Based on the literature, statistical and AI models have been able to predict fault occurrence and forecast its impact on the system efficiently. However, the above-mentioned techniques are considered to be black box, where their mapping process needs to be unlocked and represented in an interpretable manner. This limitation is addressed by the rules-based and graphical models (glass box) that enable the discovery of possible fault causalities in the form of interpretable knowledge. Their exploitation in complex systems are too limited, since the models do not address the fault's hierarchical structures. In addition, capturing its influences of the system wear-out and the changes in the causality structure over time is a big challenge.

2.5 Model-based fault diagnosis

Model-based fault diagnosis captures fault physics, using a mathematical expression or graphical model. According to the literature, quantitative and qualitative models are employed in model-based fault diagnosis.

2.5.1 Quantitative models

Quantitative models are mainly used for monitoring the fault occurrence, in which a mathematical model is employed as a reference to capture the current system's behavior and its deviation. Residual analysis is one of the most common types of analysis for quantitative models, in which residual, or the symptom signal, is generated based on the differences between the system's inputs and prior knowledge represented by the mathematical model. Several techniques are employed for monitoring the fault occurrence, such as parity relations and Kalman filters (Venkatasubramanian, Rengaswamy, Yin, & Kavuri, 2003).

Sun et al. (2019) propose a hybrid approach for fault detection in closed loop control systems based on parity space. The original stream data is transformed into the parity space based on a stable kernel matrix to overcome the noise in the data and to improve the residual analysis results. Rigatos et al. (2019) implement the differential flatness theory based on a Kalman filter for faults in the gas-turbine and of a synchronous generator in the electric power generators. The Kalman filter is employed as a linearized equivalent model to the system simulation, while the system's output is compared with the current turbine output to generate the residuals that are later used for monitoring and detecting the fault.

2.5.2 Qualitative models

Unlike quantitative analysis, qualitative analysis underlines the fault identification and quantification steps regarding the fault causality structure and the cause-and-effect relationships between the fault events. This knowledge is usually represented by graphical models, which represent a fault's causality and qualify its occurrence in an interpretable manner (Mutlu, Arnold, Franchek, & Meraz, 2017).

Fault tree analysis (FTA) is an analytical qualitative model that selects an undesired event in the system and then performs an analysis, such as checking the system's environment and operating conditions to find all of the possible ways that led to the occurrence of that selected event. It is a graphical model of parallel and sequential faults, which leads to the selected event. The FTA uses Boolean logic gates to describe how those parallel and sequential faults are connected together to describe the occurrence of the selected event (Srivastava & Sinha, 2012). Halloul, Chiban, and Awad (2019) introduce an adaptive fuzzy fault tree hybrid approach to enrich the expert's knowledge using the fuzzy set theory to overcome the lack of probability calculations for basic events. Mukherjee, and Chakraborty (2007) proposed a hybrid approach for enhancing the construction of the FTA model by analyzing the related knowledge from maintenance data in the form of text reports. The proposed methodology addresses the challenge of diagnosing the complex faults using the system's historical maintenance reports.

Additionally, the causal graphs focus on the physical cause-and-effect relationship between the system process variables, where nodes represent the variables and directed edges depict the causality relationship between the variables (Huang, Gao, & Gao, 2013). Jie, Mengyuan, Xiong, Liang, and Kaixiang (2017) propose a hybrid approach that quantifies a causal graph based on an

expert's prior knowledge. After the construction of the causal graph, the correlation index (CI) based on the partial correlation coefficient is employed to quantify the variable cause-and-effect relationship.

2.6 Model-based fault prognosis

Model-based fault prognosis relies heavily on domain knowledge to describe the faulty conditions of a system over time (Lin, Wen-Chiao & Ghoneim, Youssef A, 2016). It is classified into qualitative and quantitative models (Schwabacher, 2005).

2.6.1 Quantitative models

Celaya, Kulkarni, Biswas, Saha, and Goebel (2011) introduce a capacitor empirical degradation model based on accelerated data. The model is able to predict and quantify the RUL while determining the capacitor lumped-parameter and the capacitance equivalent series resistance (ESR) during the degradation process. Xiang, Chen, He, Li, and He (2005) propose a model-based to quantify the nature frequent crack faults by determining the crack parameters based on experimental data. Based on the crack element of B-spline wavelet on the interval (BSWI), the crack equivalent strength and its location data are employed as input data to quantify the crack size and location. Kawatsu (2019) develops a hybrid approach model-based for a rocket engine to optimize maintenance costs as well as to sustain its reliability and safety. Due to the limited amount of rocket engine failure data, the model-based quantitative assessment is built based on the Modelica modelling language. The model simulates the operation of the engine, and the generated data is compared with actual measurements of sensors using the Dynamic Time Warping (DTW) algorithm. Then, the hierarchical clustering technique is applied to categorize the possible failure model based on this dissimilarity. Hu, Zhang, and Wang (2016) introduced a quantitative safety framework for fault prognosis and early warning. In that framework, the expert represents the fault causality chain based on the GTST-DMLD (goal tree success tree, GTST; dynamic master logic diagram, DMLD) models for anticipating the future fault behavior and quantify its associated risks in the petroleum system.

2.6.2 Qualitative models

The qualitative models focus more on qualitative functions, centred on different units in a system. Kim, and Mylaraswamy (2006) propose an algorithm to monitor fault development over time in gas turbine engines. The model is able to generate evidence based on three selected events that monitor the speed at peak EGT (exhaust gas temperature). Lu, Jiang, Wang, Lu, and Chen (2012) address the drawback of the system downtime due to fault evolution in complex industrial process. The proposed hybrid methodology estimates the time delay in the process industry. First, time-delayed mutual information (TDMI) is employed to model the fault causality in the form of a time-delayed signed digraph (TD-SDG) mode. Then, a general fault prognosis strategy is used to optimize the system's downtime based on TD-SDG and PCA. Chen, B., Matthews, and Tavner (2013) combine the expert's prior knowledge with the data to improve fault prognosis in wind turbine fans based on a hybrid approach. The expert's prior knowledge is modeled and enriched by unseen or overlooked events that are discovered from the data.

Djeziri, Ananou, and Ouladsine (2013) proposed a hybrid approach for managing the possible faults in mechatronic system. The RUL is estimated based on the fault trend reconstruction by integrating the Principal Component Analysis (PCA) and the fault direction matrix with qualitative multi-physical model. Liu, S., Zhu, Zhang, and Wu (2016) proposed a methodology for health status assessment in the high-speed railway catenaries system. The health index is calculated based on the AHP and entropy method, subjective and objective weights of the catenary. Then, grey clustering method is employed to define the different health states that each state is analyzed through qualitative and quantitative analysis to understand the state evolution over time.

Indeed, model-based fault prognosis is interpretable and a relatively accurate model that could build from the first principle of the system's faults. It is mainly applicable on a simple system with well-known causes, for which the human knowledge about the faults, their occurrence and development are clear. Its limited implementation in complex system was overcome by enriching those models based on data-driven techniques, in which the unseen events are discovered and added to the model's prior knowledge. However, forming the model skeleton prior knowledge by the expert in complex system is a challenging task to identify the principal causality structure of the faulty situation and combining the extracted hidden fault knowledge from the data.

2.7 Research motivations

Data-driven and model-based research goes in parallel to construct efficient fault diagnosis and prognosis models, which form the decision-making engine in the condition-based maintenance (CBM) for fault management. This research field aims to automate the fault management process with minimal involvement of experts.

Data-driven models are able to accurately predict the fault state due to the power of data in industrial systems. However, to diagnose the fault in a complex system well, human experts look for models that are able to explain and represent the fault causality structure in addition to having prediction capability. Understanding the fault causality structure based on the represented interpretable knowledge is an essential feature in complex systems, of which experts and operators can understand the dependencies and the complex relations between faults in internal processes. Ensuring that the fault and its impact and consequences are well represented to human experts guarantee optimal CBM decisions and preventive actions.

Another challenge in a complex system with regards to data-driven fault prognosis models is graphically modeling the deterioration and performance degradation. Consequently, the fault causality structure can be changed over a system's life. Therefore, the complex systems need models that are able to capture these changes in an interpretable manner. It is a crucial feature that helps anticipate the impacts of a fault and provides more precise knowledge about the processes that will be affected in the future by a currently occurring fault.

These limitations can be partially overcome by model-based models, by providing interpretable models that summarize an expert's knowledge and his experience. This research field aims to maximize understanding of the fault management process. However, the expert's enriched knowledge is limited when it comes to demonstrating the fault causality structure and its development well over time. Although mapping system inputs and outputs is done in an interpretable manner by an expert, it is not accurate, as it done in data driven models, especially in highly complex system structures. In addition, it is biased knowledge, reflecting the expert's own beliefs and understanding of the system.

In nutshell, the following features are required by complex systems for fault diagnosis and prognosis models:

1. Models should be able to model complex fault causality in an interpretable manner to represent the collected system data for fault diagnosis tasks.
2. Models should be able to represent, in an interpretable manner, the changes in a complex fault causality structure over time for fault prognosis tasks.

The proposed ILTA in this thesis aims to link the data-driven approach with the model-based approach in order to overcome their limitations and to achieve a complex system's needs, in which fault knowledge is extracted directly from data and represented in an interpretable manner, similar to model-based representations. Its methodology is able to automatically build a data-driven graphical tree model, similar to the classic FTA, in which the fault is decomposed to its hierarchical causality structure (indicator, intermediate and root causes). In addition, the Boolean logic gates are employed to connect those different causes for exhibiting the fault occurrence in a top-down manner. Furthermore, quantifying its occurrence is done by assigning probabilities for the included causes to the tree and deriving a set of control rules to manage its consequences. ILTA methodology is applied in fault management by providing two advanced tree models, which are the following: the Multi-level interpretable logic tree analysis (MILTA) for fault diagnosis and Interpretable time causality analysis (ITCA) for fault prognosis in a complex system. In the next chapter, the research methodology will be illustrated in order to depict the generation of those graphical trees for diagnosis and prognosis tasks.

CHAPTER 3 SYNTHESIS OF THE WORD

This thesis provides graphical, data-driven fault tree models for achieving fault diagnosis and prognosis in complex system. The proposed methodology utilizes the FTA model-based interface logic, which can easily depict fault causality to represent extracted knowledge from databases (KDD). Therefore, the FTA model-based limitations, such as dependency on an expert's biased knowledge, is overcome through the unbiased KDD. Meanwhile, the challenges regarding constructing interpretable data-driven models is overcome by representing data knowledge through the FTA interface logic.

This chapter depicts an overview of the methodology for constructing the proposed data-driven fault trees. Section 3.1 introduces the three data-driven fault trees, the ILTA-model, the MILTA-model and the ITCA-model. The ILTA-model is a one-level tree that represents the fault causality in simple system, whereas the MILTA-model is a multi-level tree designed for a complex fault diagnosis task. Similarly, the ITCA-model is a multi-level tree, developed for the complex fault prognosis task. The ITCA-model considers the effect of time on the changes in the fault causality structure. Section 3.2 presents an overview of the three construction methodologies, in which the ILTA-model is used as the construction engine to build the other two advanced trees: the MILTA and ITCA models. Accordingly, the MILTA and ITCA models are a collection of selected ILTA-models that depict the complex fault causality structure in a multi-level tree.

3.1 Basic elements of the data-driven fault tree

The proposed methodology introduces three data-driven fault trees models, ILTA, MILTA and ITCA. The ILTA-model provides a one-level tree where the fault event has a direct relation with the root-causes. The fault is represented through three layers: the solution, pattern and condition layers, which represent the fault causality knowledge in a bottom-up manner.

The condition (C) is the basic block to form the KDD, it includes three elements; data variable, inequality sign and cut point value (e.g. $C_1: X_1 > 10$). The combination of those three elements are able to isolate certain data observations and summarize their common characteristics. The pattern (P) is a conjunction of some conditions that discriminate one class of observations from the other classes (e.g. $P_1: \{C_1: X_1 > 10\}\, AND\, \{C_2: X_3 > 200\}$). The solution (S) is defined as a

combination of certain patterns that cover the observations of the same class (e.g. $S_1 = P_1: \{C_1: X_1 > 10\}\ AND\ \{C_2: X_3 > 200\}\ \ OR\ P_2: \{C_3: X_1 \leq 10\}\ AND\ \{C_4: X_3 \leq 200\}$).

The solution (S) selection is done by choosing the best patterns (P) that maximize the class observation coverage. Therefore, this solution is capable of depicting the class causality knowledge based on its included patterns and conditions. Finally, visualizing the solution is achieved through the three accumulated layers: the solution, pattern and condition layer. Figure 3.1 depicts an example of the ILTA tree structure with the solution, pattern and condition layers.

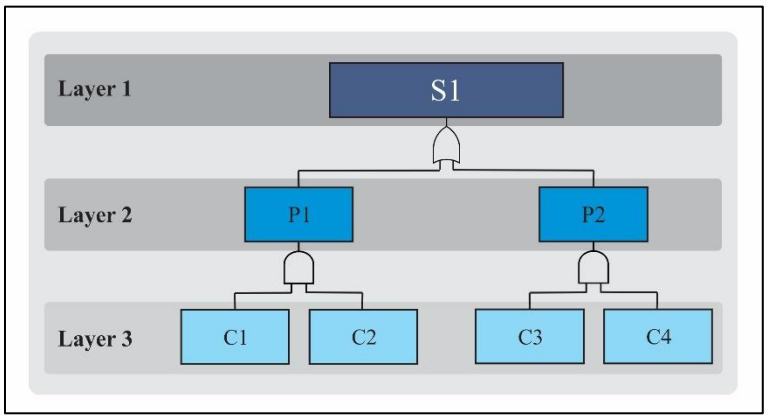


Figure 3.1 The ITLA-model construction layers

Indeed, the ILTA-model one level tree is employed in this methodology as the construction block for building the MILTA and ITCA trees. Each tree uses several ILTA-models that are connected together to represent the cause-and-effect relationship of faults. The MILTA-model is developed for fault diagnosis in complex system, which mimics the classic FTA graphical representation. The MILTA-model depicts the discovered fault causality knowledge and its cause-and-effect relationship directly from the system databases using a multilevel tree. The obtained tree characterizes the causality hierarchical structure related to the fault at a certain time period. The MILTA-model is composed of different connected ILTA-models, where the cause-and-effect chain is represented by the relation between the lower decomposition level and the upper ones, until reaching the fault event. Moreover, the first levels of the MILTA-model carry the causality knowledge about the fault indicators followed by the causality knowledge of the intermediate causes and then by the root causes.

The ITCA-model is developed for fault prognosis in complex systems, which replicate the MILTA-model in time. Thus, the ITCA-model characterizes the fault causality evolution and the changes

of the causality structure over several periods of time to achieve the fault prognosis task. The ITCA tree has similar structure and functionality of the MILTA-model to diagnose the fault at a certain time, but it replicates the MILTA-model over time to catch the evolution of the causality structure over time.

3.2 Methodology overview

The three trees can be put in order according to the simplicity of their structure, starting with an ILTA-model that provides only a one-level tree for simple faults. Then, an MILTA-model characterizes the complex fault causality structure without considering the system wear-out effect (aging) on a change in the tree structure. Reaching the ITCA-model, this has the most complex tree, since it includes a time multi-level causality tree that captures the evolution of the fault causality at different periods of time over the system aging.

Since the ILTA-model is the simplest tree (Figure 3.1), it is employed in the MILTA-model to decompose the complex fault into sequential, simple sub-trees that are connected at different decomposition levels. Figure 3.2 depicts an example of the ILTA-model's functionality in building the MILTA-model. Starting from the main event, the ILTA1-model is constructed to initiate the first level of the MILTA-model. Then, the discovered root-causes (C_1 and C_2) consider the new events that need two new ILTA-models. The ILTA2 and ILTA3 models characterize the causality structures related to C_1 and C_2 at level 2, respectively. This decomposition process continues until the root causes of the fault are found.

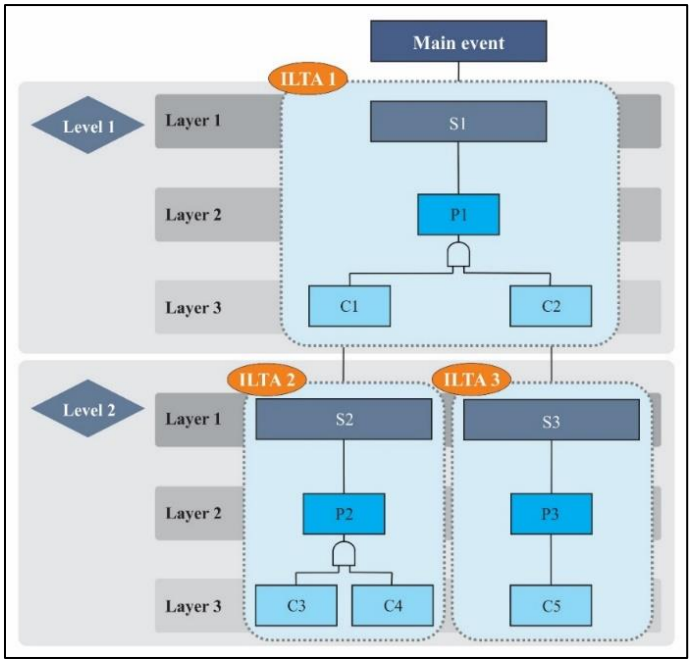


Figure 3.2 The MILTA-model construction process

With the same analogy, the ITCA-model characterizes the fault causality structures at different periods of time over the fault evolution to capture the effect of the system's aging on the causality structure's changes. Figure 3.3 presents an example of the ITCA-model construction process over three periods of time Δ_1, $\Delta_2$and Δ_3, which uses the ILTA-model functionalities at each period and decomposition level.

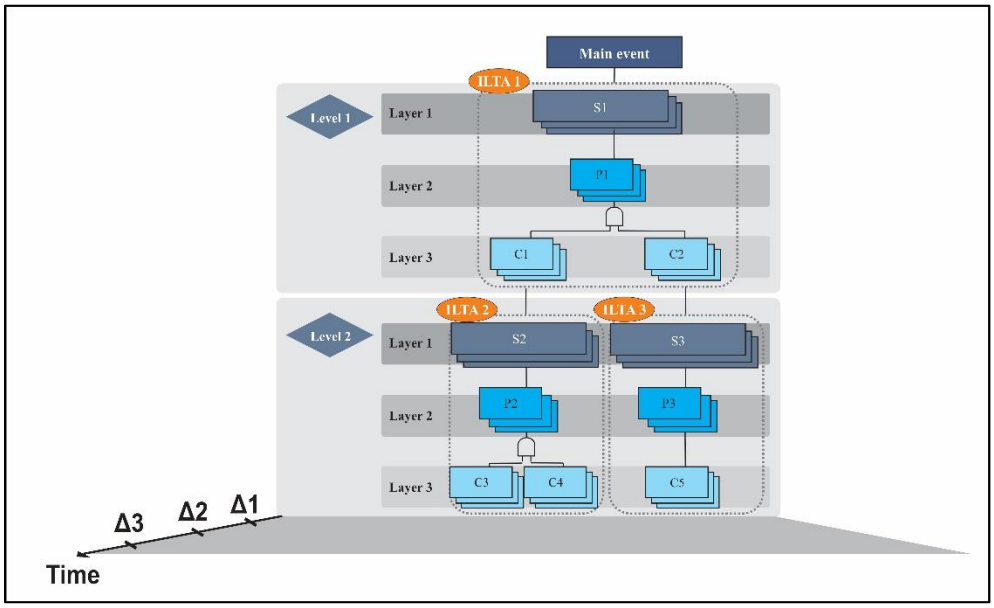


Figure 3.3 ITCA construction process

The next three chapters (chapter 4, 5 and 6) will present the three construction methodologies in detail. The results of the ILTA, MILTA and ITCA models are discussed. In addition, a case study is performed for each model to illustrate its usefulness in a fault causality analysis.

CHAPTER 4 ARTICLE 1: INTERPRETABLE LOGIC TREE ANALYSIS: A DATA-DRIVEN FAULT TREE METHODOLOGY FOR CAUSALITY ANALYSIS

Kerelous Waghen & Mohamed-Salah Ouali*

Mathematics and Industrial Engineering Department, Polytechnique Montréal

2500 Chemin de Polytechnique, Montréal, QC, H3T 1J4, Canada

kerelous.waghen@polymtl.ca; mohamed-salah.souali@polymtl.ca

*Member of CIRRELT (https://www.cirrelt.ca) and IVADO (https://ivado.ca)

Published at Expert systems with Application journal

(https://www.sciencedirect.com/science/article/pii/S0957417419304427)

Abstract:

This paper proposes an effective hybrid-based methodology, called interpretable logic tree analysis (ILTA), which characterizes and quantifies event causality occurring in engineering systems with the minimum involvement of human experts. It integrates two concepts: knowledge discovery in database (KDD) and fault tree analysis (FTA). The KDD extracts the root-causes in the form of a set of interpretable (meaningful) patterns and then is exploited to automatically construct a logic tree. Only the feasible solutions consisting of non-redundant patterns that cover the maximum number of observations in the dataset are selected using a burn-and-build algorithm. These solutions are employed first to visualize the discovered knowledge under the interpretable logic tree and second, to estimate the probability of an event given the occurrence of its root-causes. An actuator system dataset is used to illustrate and validate the proposed methodology. Moreover, the ILTA methodology allows the tuning of the system states based on Bayesian control rules that characterize the nature of the discovered root-causes.

Keywords: Causality Analysis; Fault Tree Analysis (FTA); Knowledge Discovery in Database (KDD); Decision Support System (DSS).

4.1 Introduction

Managing the health of complex systems such as those found in telecommunications, military, airplanes, chemical plants and heavy equipment involves a high level of human expertise to diagnose the system behaviour over time. The diagnosis aims to analyze the system abnormalities (faults) compared to the normal situation and to identify the main root-causes that have a serious impact on the overall system reliability and safety (Liu, R., Yang, Zio, & Chen, 2018). It supports the decision-maker with timely, actionable information that improves the performance of the targeted process (Li, H.-C., Wu, Gao, & Zhang, 2006; Yin, Ding, & Zhou, 2016).

Fault tree analysis (FTA) is a probability-based technique in which the Boolean logic gates are used to estimate the risk of an undesired top event in the system based on the probability of its root-causes (Baig et al., 2013). Two main advantages are supporting its wide use in system engineering. First, the FTA is a team-based problem modelling technique that can pool multiple expert opinions to identify the problem and its related root-causes (Levy, 1997). Second, the FTA searches for specific combinations of the root-causes and their probabilities of providing a consensual analysis that improves the targeted system reliability (Mayer & Hennings, 2008). The FTA is commonly applied in a multitude of domains such as risk management (Chemweno, Pintelon, Muchiri, & Van Horenbeek, 2018), reliability engineering (Mi, Li, Peng, & Huang, 2018; Zampino & Packard, 2005) and system safety (Peeters, Basten, & Tinga, 2018) to improve both the design and certainty of systems. The FTA helps engineers understand how faults occur in the system so that they can provide efficient solutions to avoid or reduce the effects of the identified root-causes (Ayyub, 2014).

Although the FTA provides relevant analysis of system faults, it requires the involvement of experts from different fields with detailed system knowledge to properly build a fault tree (FT). For many applications, this requirement is costly, particularly when they have to consider the outsourcing of domain experts. Moreover, the expert's knowledge can limit the deployment of the FT by focusing only on certain sequential causalities of root-causes and omitting others. Accordingly, the obtained FT may lead to partial identification and analysis for the root-causes, which can be inefficient in diagnosing the system conditions. An improvement on the FT construction approach that automatically captures hidden knowledge within the system's historical dataset is therefore needed.

Knowledge discovery in database (KDD) is an emerging approach that allows extracting useful information from a system's past experiences. The cornerstone of the KDD approach is predictive and descriptive machine learning methods that discover the hidden phenomena by exploiting the system's historical data (Choudhary, Harding, & Tiwari, 2009). Logical analysis of data (LAD) (Boros, Endre et al., 2000) and rough set theory (RST) (Pawlak, 1982) are two common descriptive machine learning methods that are based on discovering patterns within a labelled dataset. The discovered patterns are represented as sets of if-then rules that are readable (interpretable) to the human. The coverage of a pattern is an important parameter, defined as the number of data observations that are covered by that pattern. Those patterns represent the system characteristics, boundary and causality between the explicative variables in the dataset (Chikalov et al., 2013). An advantage of LAD over RST is that LAD allows discovering comprehensive and comprehensible rules in many real-world classification problems (Jocelyn, Chinniah, Ouali, & Yacout, 2017; Ragab, Ahmed, El-Koujok, Poulin, Amazouz, & Yacout, 2018) (Mortada, Yacout, & Lakis, 2011).

LAD is applied to industrial chemical processes in (Ragab, Ahmed et al., 2018) and shows great performance over other comparable methods to analyze the faulty states due to the structure of its interpretable patterns. Mortada et al. (2011) and Jocelyn et al. (2017) applied LAD to diagnose faults in rolling bearings by using a modified pattern generation method and machinery-related occupational accidents, respectively. For large-scale systems with a large number of interacting variables in the data, the number of extracted patterns becomes large. Some of these patterns may represent the main root-causes, while other patterns are redundant and represent irrelevant knowledge. As reported in (Kiraly, Laiho, Abonyi, & Gyenesei, 2014; Lucchese, Orlando, & Perego, 2010), pattern overlapping is a quality measurement for the presence of repetition and redundancy in the discovered knowledge. Therefore, the selection of relevant patterns that capture the major part of hidden knowledge in the system becomes a big challenge, which requires a tedious amount of effort to summarize the discovered knowledge.

This paper aims to develop an effective hybrid-based methodology, called interpretable logic tree analysis (ILTA), which merges the KDD represented by LAD with the FTA to characterize and quantify the system causality with a minimum expert involvement. Unlike conventional FTA, the proposed ILTA methodology can automatically explore the system's event causality in an efficient way. For a given system state (referred later to a class), the LAD method is employed to extract the hidden root-causes, in the form of interpretable patterns, within the system's historical dataset.

The overlapping and coverage of patterns are used as quality criteria in this work to select the relevant patterns. They are mathematical artifices, which help rank the discovered patterns according to their significance. The selection problem is formulated such that it only selects some of the combinations of patterns that maximize the coverage in a certain class while minimizing the overlap between selected patterns. An algorithm called burn-and-build is developed in this paper to solve the pattern selection formulation.

The occurrence probability of a given top event occurring in the system can be calculated as the coverage of the selected patterns resulting from the obtained feasible solution. The calculated probability can therefore fully or partially describe the system state according to the selected pattern coverage. Moreover, these probabilities allow a quantitative causality analysis to be performed on the system top event. The probabilities of patterns are calculated in (Jocelyn, Ouali, & Chinniah, 2018) using the occurrence of its constituent attributes. In that method, the patterns are ranked according to their coverages (probabilities) that reflect the risk of harm severity. In the proposed ILTA methodology, the above estimation technique is modified to calculate the probability of each combination of patterns that characterize a given event using Bayes' theorem.

The proposed hybrid-based methodology has two basic benefits. First, it automatically visualizes and interprets the system behaviour (cause-effect) without any expert involvement during the model construction. Second, it enables the occurrence probability of system events to be set by tuning their root-cause variabilities for the system diagnosis. The ILTA methodology is comprised of four stages. Stage 1 extracts the patterns from the system's historical dataset using the LAD method. Each pattern is a conjunction of conditions (root-causes). Stage 2 searches for feasible solutions for the pattern selection formulation while maximizing the observation coverage in a certain class. Stage 3 uses the Boolean logic gates of the FT as building blocks to construct the ILTA model based on the involved conditions, patterns and feasible solutions. Finally, Stage 4 assigns the occurrence probability of each system state based on the probabilities of its conditions, patterns and solutions.

The rest of the paper is organized into six sections. Section 2 classifies and reviews the application of the FTA models and their limitations. Section 3 provides an overview of the LAD approach and describes its terminologies. Section 4 presents the details of the proposed ILTA methodology and its constituent stages. Section 5 illustrates the proposed methodology using an actuator system

database and discusses the robustness of the obtained ILTA model in the system control. Section 6 concludes the paper and discusses the pros and cons of the proposed methodology along with future research.

4.2 Review of Fault Tree Analysis models

The FTA model was first developed at Bell Laboratories in 1962 by H.A. Watson, later improved by Boeing and then applied in several domains such as nuclear power plants (Kwag, Oh, & Lee, 2018), chemical processes (Xiao-Ping & Wei-Hua, 2017), and power electronics (Saponara, Ciarpi, & Fanucci, 2018). The FTA requires human expertise to identify a system's event causality in a deductive top-down way based on the detailed knowledge of each system (Beresh, Ciufo, & Anders, 2007). According to the literature and practice, three categories in the FTA have been identified: the classical fault tree analysis (FTA), computer-aided fault tree (CAFT), and more recently the fault tree based on knowledge discovery (FTKD).

The fundamental concept of the FTA is to transform the physical state description for a system into a logic diagram. The root-causes of a selected top event in the system are connected to the Boolean logic gates in a deductive way. There are two main characteristics of the FTA. First, it provides a qualitative evaluation of the system's state by finding the minimal paths that lead to such a top event (Zeng, Kang, Wen, & Zio, 2018). Second, it provides a quantitative evaluation by assessing the likelihood probabilities of the root-causes, therefore the expected occurrence probability of the top event can be determined (Kabir, 2017; Lee, Grosh, Tillman, & Lie, 1985).

The FTA was used extensively in the literature to address the risk management issues related to distinct events. Burkhalter, Martani, and Adey (2018) used the FTA to determine the optimal risk-reducing intervention strategy for railway lines based on the system states and budget availability. Sihombing, and Torbol (2018) proposed a FTA model for the risk assessment of components and subsystems of nuclear power plants. Melani, Murad, Caminada Netto, Souza, and Nabeta (2018) developed a FTA model to identify the critical components within a system and to prioritize the maintenance actions. Sule, Khan, Butt, and Yang (2018) studied the safety and reliability assessment of a managed pressure drilling operation by investigating the kick control operation based on a FTA model. Thapaliya, Jeong, and Kwon (2018) characterized the failure state of a railway level crossing system using a FTA model and highlighted preventive maintenance actions

to reduce the probability of failure. Bensaci, Zennir, Pomorski, and Mechhoud (2017) applied the failure mode effects and analysis (FMEA) method and the FTA to ensure high precision of the robot's navigation to avoid dangerous accidents when moving chemical products.

The limitation of building an FTA model is that it is time-consuming and it requires tremendous effort from the human experts from multiple engineering domains to investigate the root-causes, based on detailed knowledge and laboratory tests. Moreover, the FT becomes very tedious to build when many events are considered under the study. Human experts have to select high risk and frequent events. Therefore, some significant and unexpected fault scenarios may be overlooked. To address this limitation, the computer-aided fault tree (CAFT) methods are proposed in the literature. The CAFT automatically generates an FT using a set of predefined blocks. Therefore, the CAFT becomes more suitable to analyze multiple dependent failure events and helps engineers to track common root-causes that add relevant diagnosis information related to such events.

Similar to the FTA, the CAFT model needs a full system description. Two strategies of the CAFT models are distinguished in the literature. The first philosophy uses only a predefined set of standard failure modes for the automatic FT construction. Fussell (1973) is one of the first instigators of this strategy. He constructed the FT using the system boundary conditions and the failure transfer function to describe the failure modes. Lapp, and Powers (1977) developed an FT construction method based on an oriented digraph. The second category of CAFT improves the interaction between the analyst and the model, such that he or she can formalize abnormal situations and control the entire system’s behaviour. Mhenni, Nguyen, and Choley (2014) generated a CAFT for a system based on the SysML models. Zhang, Yanhua, Ren, Liu, and Wang (2015) used the so-called Go Model method to build the FT. The method defines a set of mapping rules from the common Go operators to FT nodes. Then an algorithm is used to transform the complete Go graph into its equivalent FT. Bhagavatula, Tao, Dunnett, and Bell (2016) developed a method based on the multi-state input/output tables, the component library and mark library to redraw the system description in a graphical form, then build the CAFT. Although the CAFT model is able to generate a generalized FTA model for a given system, it requires a full system description. This may not be flexible enough to model the complexity of present-day industrial systems. It poses another limitation with regards to the interaction between the analyst and the CAFT model since some phenomena could be overlooked depending on an expert’s knowledge and background.

The third category of FTA is motivated by the revolution in the domain of knowledge discovery in database (KDD). It is the fault tree based on knowledge discovery techniques (FTKD). The main concept of this category is to use descriptive machine learning and artificial intelligence (AI) methods to build an FTA that reflects the hidden phenomena in the dataset. The FTKD helps the expert enrich the obtained FTA models. Khoo, Tor, and Li (2001) used rough set theory as a descriptive machine learning method to rank the basic events within an FTA based on their importance. The method allows learning from experience and expert knowledge. Li, R., Li, and Su (2005) applied the rough set theory to FTA in order to calculate the relative significance of the bottom events and the occurrence probability of a top event. Tu, Duan, and Dong (2010) developed a fault tree using a rough set theory that work proposes a fuzzy clustering algorithm to select the most important data variables and to draw a dynamic clustering map for the fault samples and then transform this map into a fault tree. Yiu, Cheung, and Lok (2015) introduced a fuzzy fault tree framework (FFTA) to conceptualize the root-causes of the failure of negotiation in a construction dispute. Papadopoulos et al. (2011) developed an automated construction tool of Fault Trees and Failure Modes and Effects Analyses (FMEAs) based on Hierarchically Performed Hazard Origin & Propagation Studies (HiP-HOPS). The tool optimizes the selection of the system components based on a compromise between the improvement of the system reliability and the reduction of solution cost.

In the FTKD models, the role of the KDD is to help highlight the root-causes that are difficult for human experts to observe. Such causes can be observed from the system's historical data; the role of experts is to validate the obtained tree and analyze its applicability. However, the automation of the FT construction process based on KDD without the involvement of a committee of experts is strongly needed in real applications and has not been addressed in the literature. This is the main objective of the current paper. We automate the FT construction process based on the interpretable patterns of LAD.

4.3 Logical Analysis of Data for KDD

Logical Analysis of Data (LAD) is a supervised pattern generation and classification method introduced by Peter L. Hammer in 1986 (Hammer, P., 1986). The main concept of LAD is to extract human interpretable patterns from a labelled dataset. It is based on some concepts from the theory of Boolean functions, artificial intelligence and combinatorial optimization (Chikalov et al., 2013).

It consists of three main steps; data binarization, pattern generation and theory formation (Boros, Endre et al., 2000). In data binarization, LAD converts the variables in the datasets, whether numerical or categorical, to binary attributes using a set of cut points. More details about the binarization step and the optimization of the number of cut points are found in (Boros, Endre, Hammer, Ibaraki, & Kogan, 1997). The pattern generation is the most significant step in LAD. The aim is to extract a set of positive (+) and negative (-) patterns that distinguish between (+) and (-) observations. In the theory formation step, the extracted patterns are used to build a decision model that classifies the new observations that have not been seen in the training data. Interested readers can refer to (Boros, Endre et al., 2000) for more details about these steps. In what follows, the pattern definition and its main properties are presented.

The pattern is defined as a logical conjunction of some literals. The literal characterizes a binary attribute or its negation in a pattern. The extracted patterns could be pure or non-pure (or fuzzy) (Bonates, Hammer, & Kogan, 2008). The pure pattern covers some observations of one class and none of the observations in the opposite class. The fuzzy pattern covers some observations in one class and is allowed to cover some others in the opposite class due to constraining relaxation. Three parameters are used to define the pattern quality: degree, prevalence, and homogeneity. The *degree* is the number of literals that constitute the pattern. A high degree pattern is more likely to cover a small set of observations, while a low degree pattern is more likely to cover a larger set of observations (Boros, Endre et al., 2000). The *absolute prevalence* of the pattern is defined as the number of observations covered by that pattern. The *relative prevalence* of the pattern (coverage) is defined as the ratio between the absolute prevalence and the number of observations in the entire dataset. The *homogeneity* is the ratio of the absolute positive prevalence to the absolute prevalence of the pattern (Alexe & Hammer, 2006). LAD pattern types are defined according to the characteristics found in (Hammer, Peter L., Kogan, Simeone, & Szedmák, 2004): *The prime pattern* is such that if we remove any of its literal, it will no longer be a pattern. *The strong pattern* is a pattern that has the largest coverage with respect to the other patterns. More details about the types of patterns are found in (Hammer, Peter L. et al., 2004).

In this paper, the notion of literal in patterns is adapted to be used and understood by the researchers and practitioners in the domains of condition monitoring and fault diagnosis. Accordingly, the pattern is represented as a conjunction (simultaneous occurrences) of certain conditions. Each condition compares the variable with the cut point which represents the transition between the

different classes of data (condition: variable ≤ (or >) cut point). Thus, the condition represents the range of variables in a certain class.

There are three common methods to generate patterns: enumeration-based methods (Alexe & Hammer, 2006), heuristics-based methods (Bonates et al., 2008) and mixed integer linear prograV2g (MILP)-based methods (Ryoo & Jang, 2009). The enumeration-based methods are convenient for small datasets only since they are time-consuming. The heuristic-based methods extract the patterns iteratively until all the observations in the dataset are covered. However, these methods do not give optimal solutions but can give feasible ones, without requiring computational efforts such as the MILP-based methods. The MILP have the ability to extract strong patterns that satisfy user-specified requirements related to the prevalence, homogeneity and complexity quality criteria (Ryoo & Jang, 2009). The accuracy of LAD over the other common machine learning techniques, namely artificial neural networks (ANN), Decision Tree (DT), Random Forest (RF), k-nearest-neighbors (kNN), quadratic discriminant analysis (QDA) and support vector machine (SVM), is highlighted in (Ragab, Ahmed et al., 2018) and (Mortada, M.-A., Yacout, S., & Lakis, A., 2014).

4.4 The Proposed ILTA Methodology

The proposed ILTA methodology enables building a logic tree model automatically based on a set of patterns extracted from the dataset using LAD. The main strength of the methodology is that it does not require the involvement of any human expert in the construction stage. The expert should verify the relevance of obtained model and keep or remove redundant discovered knowledge if necessary. In this paper, we focus on the construction of only one-level tree as a simple case in order to exhibit the effectiveness of the proposed methodology. The one-level ILTA model consists of three layers (i.e. condition, pattern and solution) that explain each event class. For complex systems, the one-level ILTA model will be employed in future research as an engine module to build a multi-level ILTA model. Figure 4.1 depicts the one-level ILTA construction methodology. It consists of four sequential stages.

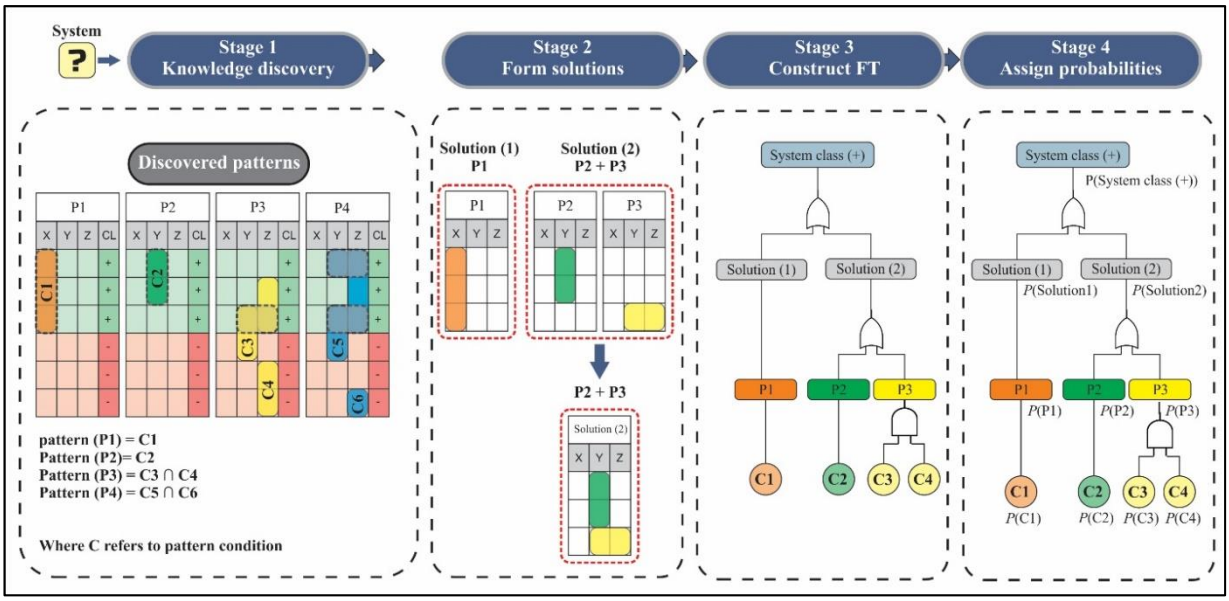


Figure 4.1 The proposed four-stage ILTA methodology

4.4.1 Stage 1-Knowledge discovery in the system's historical data

The purpose of this stage is to discover hidden knowledge within the dataset using the two-class LAD as pattern generation and selection technique. The extracted patterns characterize a specific class of observations representing a certain phenomenon against the opposite class. The patterns describe some combinations of conditions during the system's operation. The software LAD-WEKA is an open source software available at (Gomes & Bonates, 2014b), which extracts the patterns by using heuristics-based methods. The number of discovered patterns using LAD-WEKA is usually high. However, it can be constrained by tuning a parameter called the minimum class coverage, which is the minimal acceptable pattern coverage for its class to consider it in the final pattern set.

The discovered patterns may include two types of variables in their conditions: independent and dependent. The independent variables are the ones that can be directly control or influence the system without depending on any other variables. They represent independent root-causes in the dataset and are a way to define the system control boundaries. The dependent variable characterizes an output variable of a hidden relationship or interaction between some independent variables or not within the dataset. It can be used to define the system alert boundary. An extracted pattern may contain both independent and dependent variables within its conditions.

As mentioned previously, LAD can generate both types of patterns, pure and fuzzy. We exploit the concept of fuzziness and purity on the pattern conditions. However, in this case, the intersection between two or more fuzzy conditions can form a pure pattern. As shown in Figure 4.1, pattern 3 in the first stage consists of two fuzzy conditions for the variables Y and Z, as each fuzzy condition covers observations in both classes (+ and -). Both of the two fuzzy conditions form a pure pattern P_3.

4.4.2 Stage 2-Obtaining feasible solutions

This stage selects feasible solutions from a set of formed ones, where a solution is a group of selected patterns, discovered in Stage 1. Each pattern involves some specific root-causes that can represent partial casualty for a given system state. The solution is defined as a combination of some patterns that cover the observations of the same class. It explains how the patterns are combined to depict the class event. Each solution can be characterized by its coverage and overlap percentages. The solution coverage denotes the percentage of observations that are covered by any patterns of the formed solution. It may be interpreted as the representability of discovered knowledge. While the solution overlap denotes the percentage of the class observations, which are covered by more than one pattern of the solution. It may be interpreted as redundant knowledge within the formed solution. The feasible solution is defined as a solution which maximizes the coverage percentage and minimizes the overlap percentage. The search for all the feasible solutions allows the ILTA model to maximize the representability of the discovered knowledge while reducing their redundancy.

Figure 4.2 explains how the solution are formed and the feasible solutions are selected using the all combinations of the discovered patterns in Stage 1. From the example of Figure 4.1, there are 4 discovered patterns (P_1, P_2, P_3 *and* P_4) that explain the system class (+). Accordingly, when $n=1$, meaning that each solution consists of a single pattern, there are 4 possible solutions (S_1 to S_4). For each solution, the coverage (*Cov*) and the overlap (*OL*) percentages are calculated and saved. For example, S_1 consists of pattern P_1 which covers 3/3 observations of the class (+) with 0/3 overlap. However, S_4 covers 2/3 observations with 0/3 overlap. When $n=2$, each pair of patterns forms a solution. There are 6 solutions (S_5 to S_{10}). For example, S_7 combines P_1 and P_4. It covers 3/3 of the observations with 2/3 overlap. There are 4 solutions (S_{11} to S_{14}) that combine three patterns ($n=3$). For example, the S_{12} superposes P_1, P_2 and P_4. It covers 3/3 observations with 3/3 overlap. Finally,

there is a single solution S_{15} that combines the four discovered patterns (*n=4*). S_{15} covers 3/3 observations with 3/3 overlap. Based on the coverage and overlap percentages of the above-enumerated solutions, S_1 and S_8 are the two only feasible solutions. Indeed, they maximize the coverage of the observations of the system state (+) (*Cov=3/3*) with a minimum overlap (*OL=0/3*).

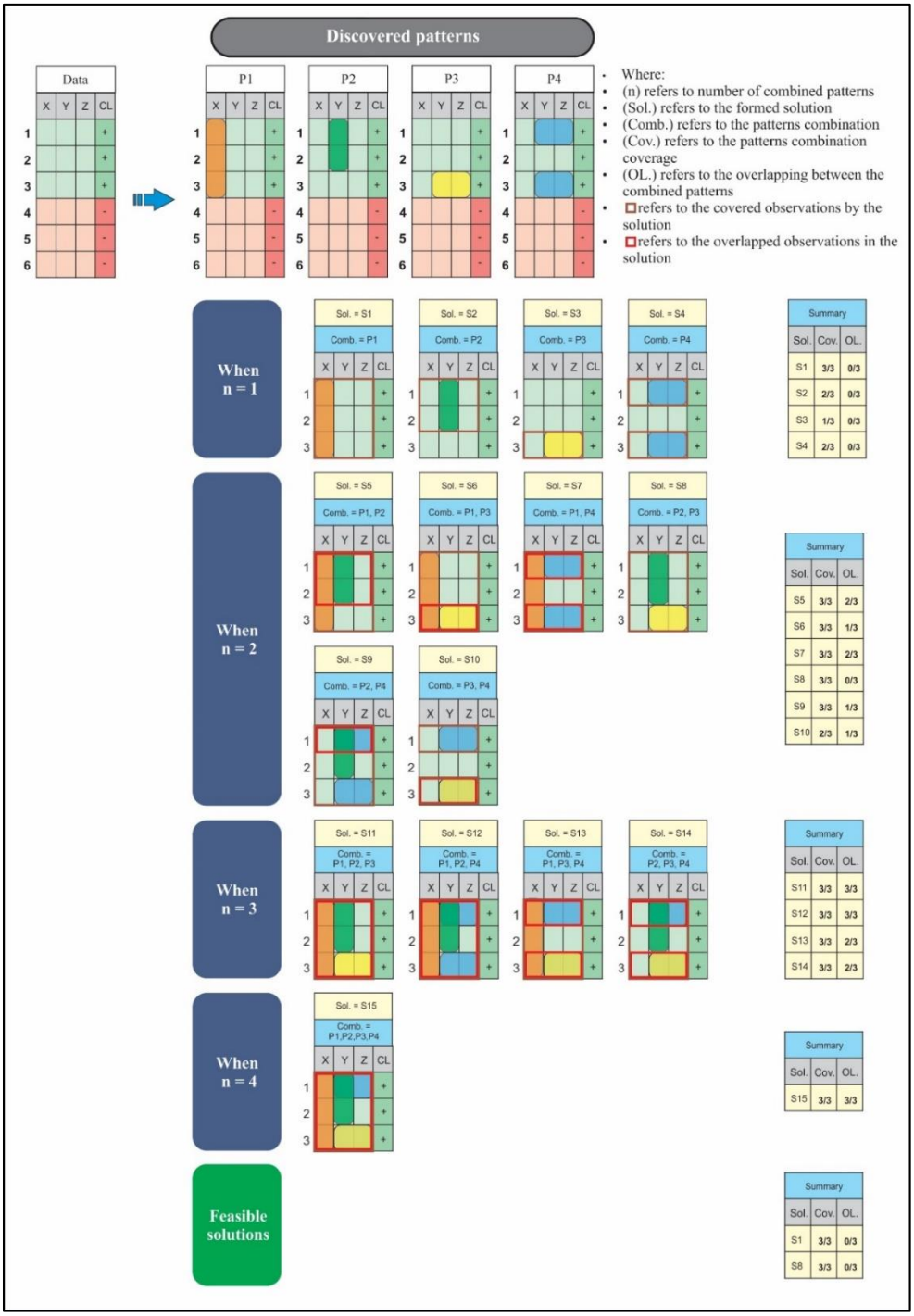


Figure 4.2 Form and select feasible solutions using the all combinations of discovered patterns

When the number of discovered patterns increases, the all pattern combination method becomes time-consuming. To overcome this problem, an iterative Burn-and-build algorithm is proposed. It finds almost all the feasible solutions in efficient way. Figure 4.3 depicts the algorithm flowchart in three steps. Step 1 initializes the algorithm parameters: the set of discovered patterns (*Pgen*); the overlap threshold (δ); the pattern index ($i=1$); and the number of combined patterns ($n=1$). Step 2 forms the solutions. Starting with a selected pattern P_i, the algorithm calculates the coverage percentage of P_i and assigns P_i to the solution $S_{i,n}$. The algorithm removes (burns) all the patterns P_j ($j \neq i$) that overlap P_i according to the predefined threshold δ. For each solution that combines P_i with its non-overlapped patterns ($n>1$), the algorithm chooses (builds) $S_{i,n}$ that gives the maximum coverage. This step is repeated for each starting pattern P_i until the all the formed solutions $S_{i,n}$ are obtained. Finally, Step 3 selects the feasible solutions $S_{i,n}$ that maximize the coverage over all the formed solutions.

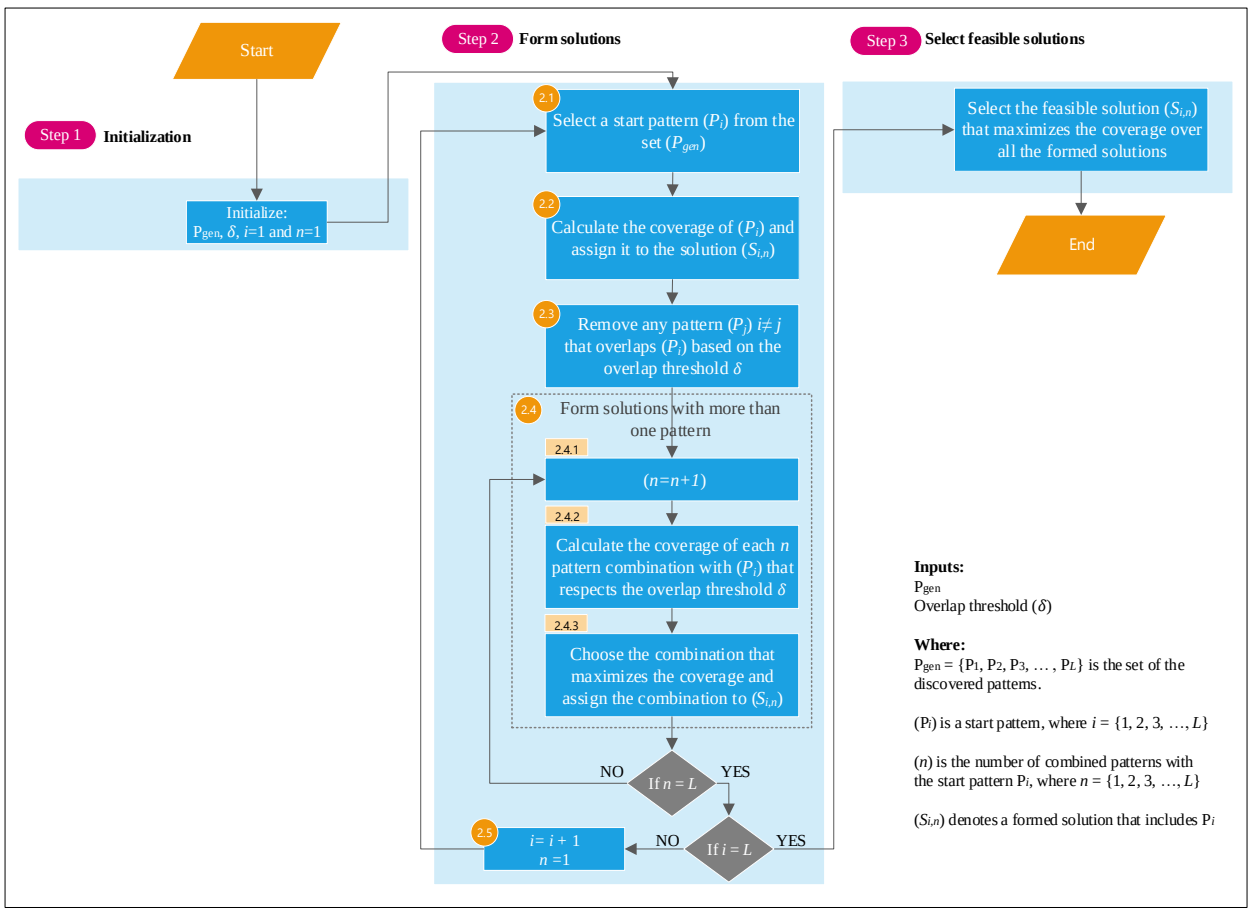


Figure 4.3 Form and select feasible solutions using the Burn-and-Build algorithm

Figure 4.4 illustrates the search for the two feasible solutions of the example of Figure 4.2 using the proposed algorithm. At the sub-step 2.1, consider for example discovered pattern P_2 $(i=2)$ as a start pattern. Then, at the sub-step 2.2 we calculate its coverage percentage and form the first solution $S_{2,1}$ $(n=1)$. At the sub-step 2.3, we remove the patterns P_1 and P_4 because they overlap P_2 according to the overlap threshold $\delta=0\%$. Then we build the solution $S_{2,2}$ that combines P_2 and P_3 $(n=2)$ at the sub-step 2.4. Once all the solutions are formed regarding to each start pattern, we select $S_{1,1}$ and $S_{2,2}$ as the only two feasible solutions because they maximize the coverage of the class observations over all the formed solutions.

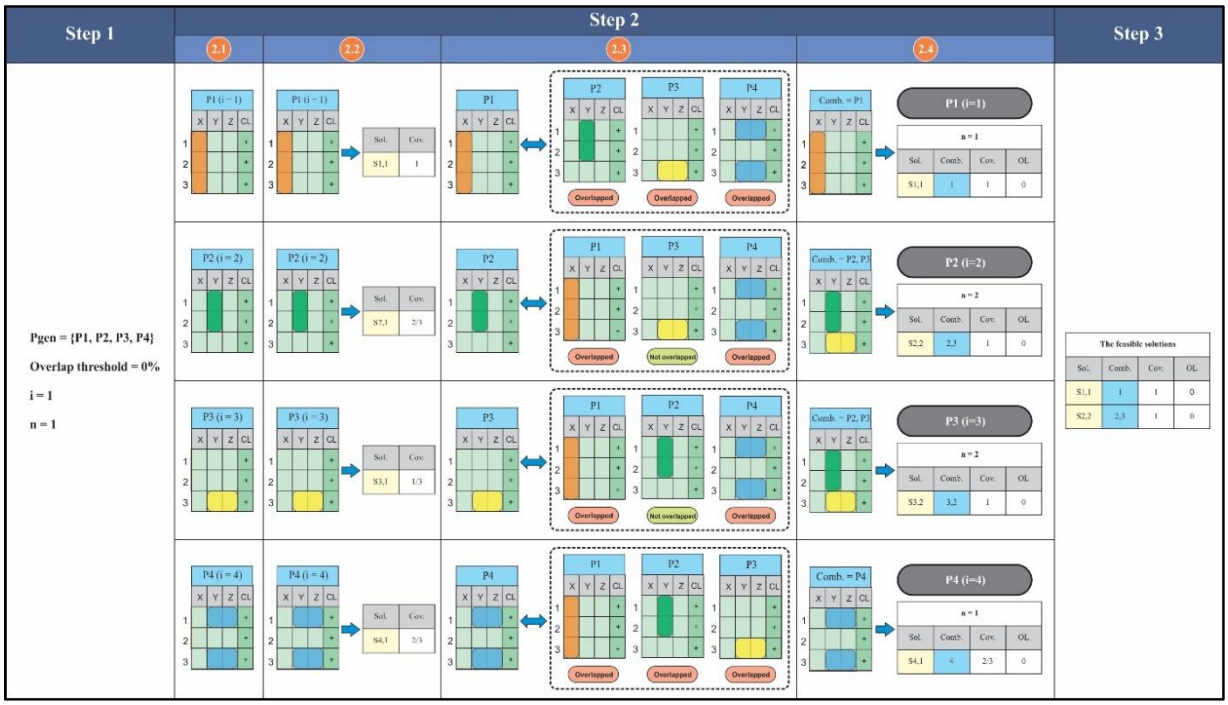


Figure 4.4 Illustration of the Burn-and-build algorithm

4.4.3 Stage 3-Logic tree construction

The feasible solutions found by the Burn-and-build algorithm are used in this stage to construct the one-level ILTA model using the OR and AND gates. The ILTA model is constructed in a top-down approach in three layers: solution, pattern and condition. At the solution layer, an upper OR gate is used, expressing the multiple feasible solutions related to the occurrence of the system class. The solution that has only one pattern is connected directly to that OR gate (see P_1 in orange, Figure 4.1, Stage 3). At the pattern layer, for any feasible solution that has a combination of more than

one pattern, such patterns are connected to solution with an OR gate (see P_2 in green and P_3 in yellow). At the condition layer, if the pattern has only one condition, such condition is connected directly to the previous OR gate (solution or pattern OR gate, see C_1 in light orange and C_2 in light green). If the pattern has more than one condition, those conditions are connected together using an AND gate since they have to occur simultaneously (see C_3 and C_4 in light yellow). Figure 4.5 shows the flowchart of the ILTA construction stage methodology.

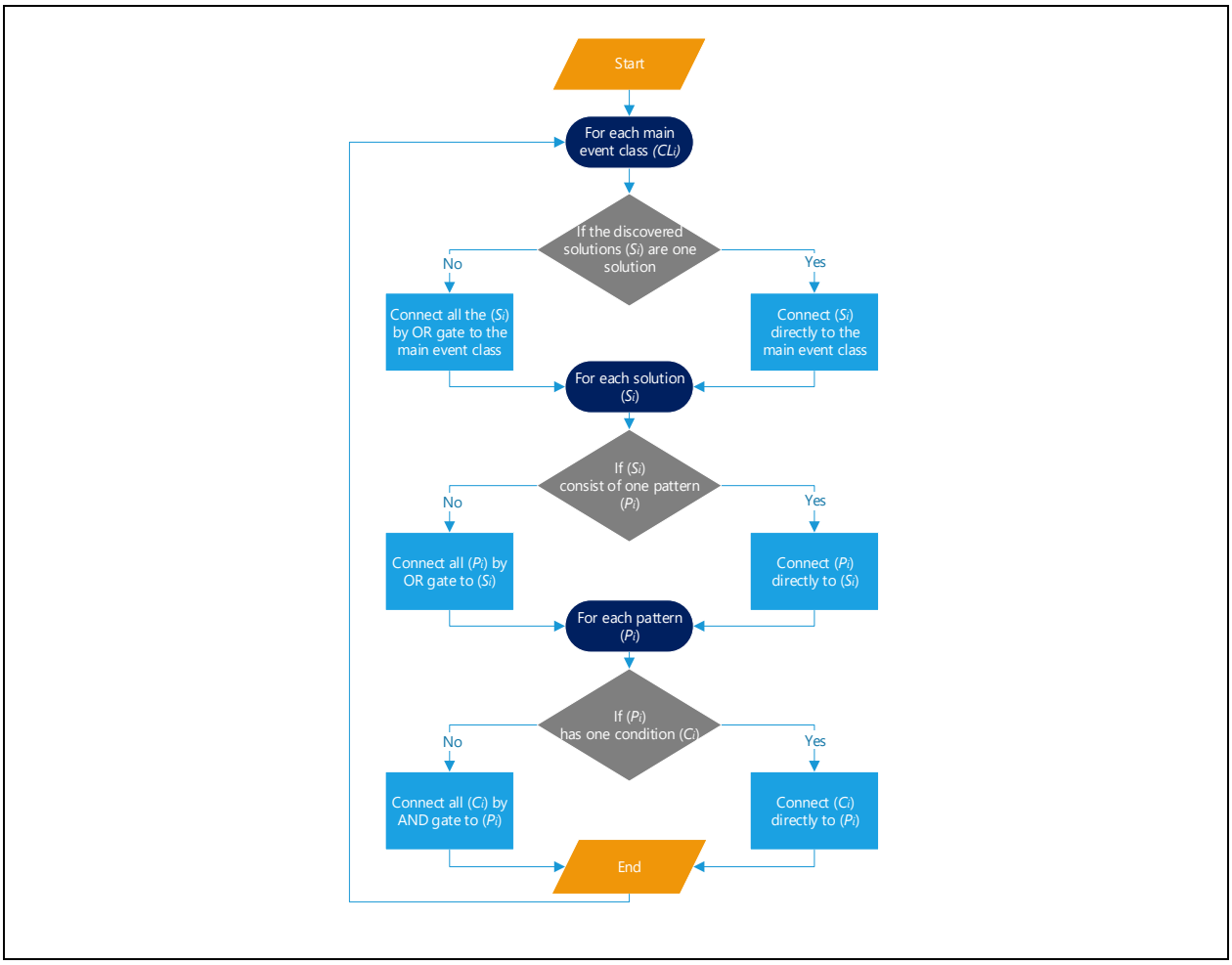


Figure 4.5 Time-OR gate functionality in the ITCA model

To illustrate the flowchart of Figure 4.5, let us consider a toy example consisting of 10 observations divided in two classes (Table 4.1). Observations 1 through 5 belong to the class (+), the other ones belong to the class (-). $X_1$and X_2 are two variables. C_1 through C_5 represent the five conditions that characterize the three discovered patterns P_1, P_2 and P_3. Therefore, the green cells represent the observations that are covered by each condition. While the green cells with the red boarders represent the conjunction of the conditions of each pattern that cover only the observations of the

positive event class (+). The ILTA model is constructed from two feasible solutions, namely S_1 and S_2, each one fully covers the event class (+). The solution S_1 consists of only one pattern P_1 identified by a conjunction of the two fuzzy conditions (C_1 and C_2). The pattern P_1 covers all observations of the class (+), defined as: $P_1: C_1 \cap C_2$. The solution S_2 comprises the following two patterns: P_2 is identified by one pure condition C_3 and P_3 is identified by a conjunction of the two fuzzy conditions (C_4 and C_5). The pattern P_2 covers the observations (1, 2, 4, and 5) and the pattern P_3 covers observation 3.

Table 4.1 Observations and obtained feasible solutions of the toy example

	S1+ = P1		S2+ = P2 ∪ P3				
	P1		P2		P3		
Obs.	**X1**	**X2**	**X1**	**X2**	**X1**	**X2**	**Class**
	L1	**L2**	**L3**		**L4**	**L5**	
1	4.8	0.3	4.8	0.3	4.8	0.3	+
2	5.1	0.4	5.1	0.4	5.1	0.4	+
3	5.5	0.2	5.5	0.2	5.5	0.2	+
4	4.8	0.2	4.8	0.2	4.8	0.2	+
5	4.7	0.2	4.7	0.2	4.7	0.2	+
6	6	1	6	1	6	1	-
7	6.1	1.2	6.1	1.2	6.1	1.2	-
8	5.7	1.2	5.7	1.2	5.7	1.2	-
9	5.5	1.3	5.5	1.3	5.5	1.3	-
10	5.6	1.3	5.6	1.3	5.6	1.3	-

P1: (X1 ≤ 5.55) AND (X2 ≤ 1) P2: (X1 <= 5.3) P3: (X1 > 5.3) AND (X2 ≤ 0.7)

Figure 4.6 visualizes the two solutions S_1 and S_2 in the form of a one-level logic tree based on the flowchart of Figure 4.5.

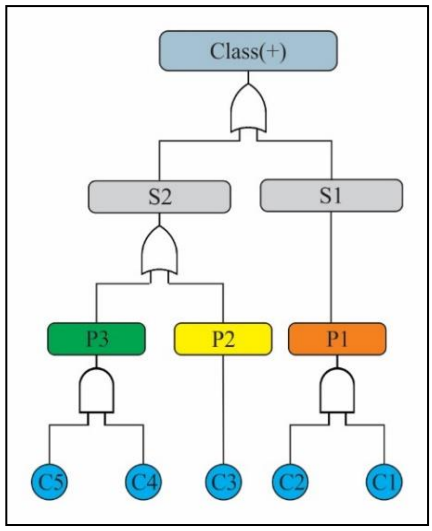


Figure 4.6 Visualization of the one-level ILTA model of the toy example

4.4.4 Stage 4-Probability calculations

After constructing the logic tree by connecting the solutions, patterns and conditions using the Boolean logic gates, the occurrence probability of each solution is then estimated using the occurrence of its constituent patterns and conditions. At the condition layer, the probability of each condition C_i, $i = 1..I$ is equal to the ratio given by the number of observations N_i covered by C_i divided by the total number of observations N_T in the dataset sample.

$$\mathcal{P}(C_i) = \frac{N_i}{N_T} \qquad (1)$$

At the pattern layer, the probability of each pattern P_j, $j = 1..J$ is equal to the product (intersection) of the probabilities of its conditions, where n_j is the number of conditions in P_j:

$$\mathcal{P}(P_j) = \prod_{i=1}^{n_j-1} \mathcal{P}(C_i \cap C_{i+1}) \qquad (2)$$

Using Baye's theorem (Bernardo & Smith, 2001), the pattern probability $\mathcal{P}(P_j)$ is given by:

$$\mathcal{P}(P_j) = \prod_{i=1}^{n_j-1} \mathcal{P}(C_i|C_{i+1}).\mathcal{P}(C_{i+1}) \qquad (3)$$

The value of $\mathcal{P}(C_i|C_{i+1})$ represents the degree of intersection (dependency) between the two conditions C_i and C_{i+1} with respect to the occurrence of C_{i+1}. Equation (3) represents the occurrence of the pattern by using only one of its conditions. Therefore, controlling the occurrence of the pattern can be done in different ways based on its conditions. This is an advantage of the proposed methodology.

At the solution layer, the probability of a feasible solution is calculated in terms of the union of probabilities of its combined patterns by using the Poincaré formula (Chelson, 1971). That formula states that for *n* events P_1, P_2, ... P_n pertaining to *S* and having intersections or not, the probability of their union is given by:

$$\mathcal{P}\left[\bigcup_{j=1}^{J} P_j\right] = \sum_{k=1}^{J}\left((-1)^{k+1} \sum_{1\leq j_1<j_2<\cdots<j_k\leq n} \mathcal{P}(P_{j1} \cap P_{j2} \cap \ldots \cap P_{jk})\right) \qquad (4)$$

Finally, the probability of a given event class C_W is given by the union of all probabilities of its feasible solutions $S_{lq}(q = 1,2, \ldots Q)$ that are calculated using Equation (4). For Q feasible solutions, the probability $\mathcal{P}(CL_W)$ of the top event class CL_W is given by:

$$\mathcal{P}(CL_W) = \mathcal{P}\left[\bigcup_{q=1}^{Q} S_q\right] \qquad (5)$$

4.5 Illustrative example

This section applies the ILTA methodology on an actuator system dataset generated from the DAMADICS benchmark simulator model (DAMADICS, 2002). The intention is to validate the constructed ILTA model since the simulator documentation provides the prior knowledge about the fault generation based on a logic equation that includes the involved root-causes. Therefore, it will be easy to compare the obtained root-causes based on the ILTA model and the ones already used by the simulator. In addition, we can estimate the accuracy of the obtained ILTA model based on a random sampling of the labelled observations.

The actuator system consists of two main parts: pneumatic linear servomotor and positioner (Figure 4.7). The pneumatic linear servomotor adjusts the valve flow according to the positioner control signal.

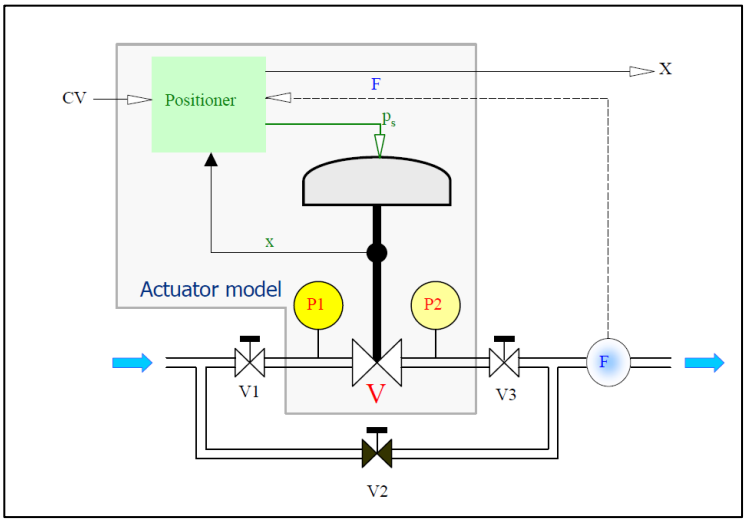


Figure 4.7 The actuator system (DAMADICS, 2002)

4.5.1 Data simulation

The DAMADICS Actuator Benchmark Library (DABLib) is a Simulink Matlab library that includes different tools. We generate a two-class labelled data: normal and fault. Regarding the fault class, we choose the fault *F7* (medium evaporation or critical flow) among 19 possible faults. Because *F7* is a simple and quite explicit fault. According to the benchmark, the physical interpretation of *F7* is "*two-phase flow (a mixture of fluid and steam) caused when local fluid pressure drops down to steam evaporation pressure level. This manifests in flashing or cavity phenomenon*" (DAMADICS, 2002). The database comprises six numerical variables (*CV*, *P1*, *P2*, *X*, *F* and *T1*) and one categorical (*F7)* (Table 4.2). Note that the numerical variables are normalized. Table 4.3 lists a sample of eight observations chosen randomly from the generated database: the first four observations are normal and the others belong to the fault class.

Table 4.2 Description of the actuator database variables

Variable ID	Description
CV	Control variable is the output signal from the controller.
P1	The value of the pressure on the control valve inlet.
P2	The value of the pressure on the control valve outlet.
X	The disturbed value of the rod displacement.
F	The disturbed medium flow.
T1	The temperature of medium.
Fault (F7)	Class variable: Medium evaporation or critical flow fault

Table 4.3 A sample from the actuator simulation data

CV	***P1***	***P2***	***X***	***F***	***T1***	***F7***
0.5	0.87683974	0.65083184	0.0015904	1	0.2145533	normal
0.51569763	0.90065326	0.65603315	0	1	0.2150963	normal
0.53133330	0.9156675	0.64501173	2.26E-05	0.999757	0.2142761	normal
0.546845329	0.91753540	0.65034119	0	1	0.21405824	normal
…	…	…	…	…	…	…
0.51569763	0.90193477	0.64356580	0.0006583	1	0.99925627	fault
0.53133330	0.91587598	0.65442129	0.0015826	0.9962099	1	fault
0.54684532	0.91528355	0.65277880	0.0013790	1	0.99848085	fault
0.56217247	0.90064185	0.64421054	0.0012165	1	1	fault
…	…	…	…	…	…	…

Figure 4.8 depicts the first two components based on the principal component analysis (PCA) (Abdi & Williams, 2010). It visualizes all the generated data in 2D space according to the normal and fault classes.

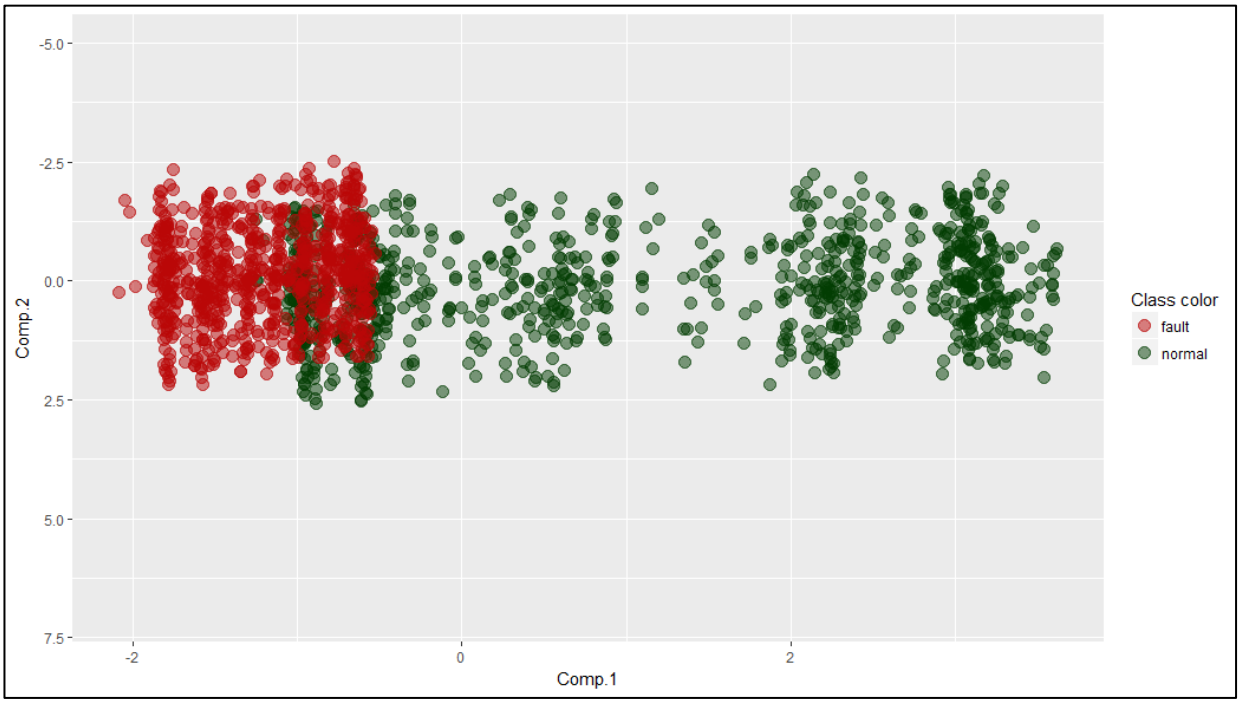


Figure 4.8 2D visualization for the actuator dataset

2401 observations labelled by the class variable (*F7*) are generated. The normal class portion is 53% and the fault one is 47%. Then a training and a testing datasets are randomly sampled. The training (respectively, testing) dataset contains 70% (respectively, 30%) of the observations. Table 4.4 summarizes the number of observations in each sample.

Table 4.4 Number of observations of the training and testing datasets

	Training dataset	**Testing dataset**	**Total**
Normal-class (+)	891	382	1273
Fault-class (-)	790	338	1128
Total	1681	720	2401

In what follows, the four stages of the proposed ILTA methodology are illustrated using the actuator training dataset.

4.5.2 ILTA model building

Stage 1-Knowledge Discovery in the actuator dataset

Based on the training dataset, the LAD-WEKA software discovers 11 (respectively, 29) patterns that discriminate the normal (respectively, fault) class (see Appendix 1). Since LAD-WEKA uses a heuristic technique to select the patterns, the minimum pattern coverage parameter is set to 2%, which means only the patterns with coverage higher or equal to that threshold will be selected.

Stage 2- Obtaining the feasible solutions.

At this stage, the Burn-and-build algorithm finds two feasible solutions for the normal class and only one for the fault class (Table 4.5). Note that each obtained feasible solution is fully covered its class without any overlapping between its constituent patterns.

Table 4.5 Feasible solutions for the normal and fault classes

Feasible solutions for Normal-class (+)	
S_1^+	$P_1 \cup P_2$
S_2^+	P_6

Feasible solutions for Fault-class (-)	
S_1^-	P_{12}

For the normal class, S_1^+ includes two patterns P_1 and P_2, P_1 consists of only one condition ($F < 0.228$). P_2 includes two conditions: $(F > 0.228)\ AND\ (T1 > 0.219)$. Figure 4.9 illustrates the solution S_1^+. Figure 4.9-A plots the variables *T1* and *F* values versus the data index sorted by the class label (green for the normal class and red for the fault one). *T1* has a constant value equal to 0.2 in the normal class and jumps to 1 in the fault class. *F* has a wavy behavior in the normal class and become steady to 1 in the fault class. Figure 4.9-B (respectively, Figure 4.9-C) plots the coverage of P_1 (respectively, P_2). Figure 4.9-D shows the coverage of the combined patterns that form the solution S_1^+. By the same way, S_2^+ consists of one pattern P_6 that includes a single condition ($T1 \leq 0.607$). Figure 4.10 plots the variable behaviour (Figure 4.10-A) and the coverage of S_2^+ (Figure 4.10-B).

For the fault class, Figure 4.11 illustrates the only one feasible solution S_1^- with respect to its single condition ($T1 > 0.607$).

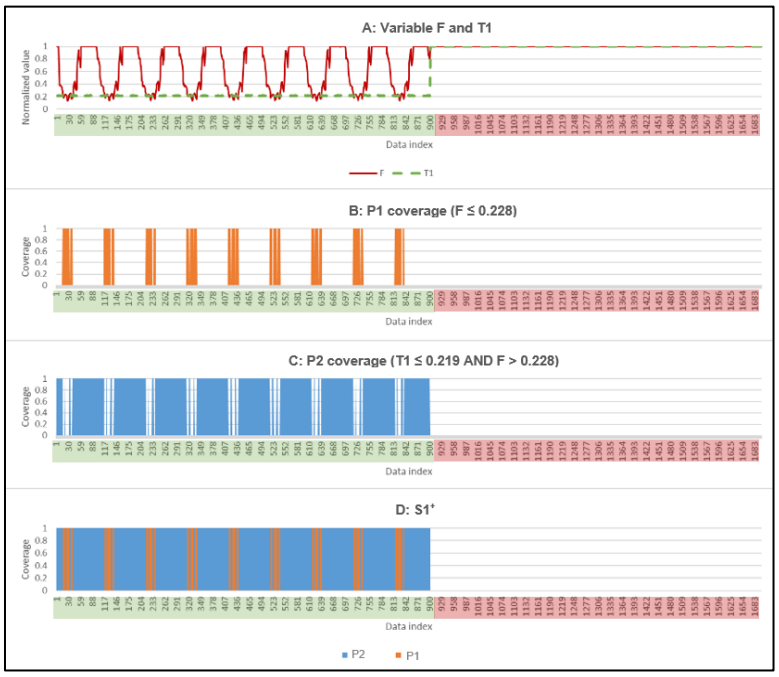


Figure 4.9 Illustration of the feasible solution S_1^+

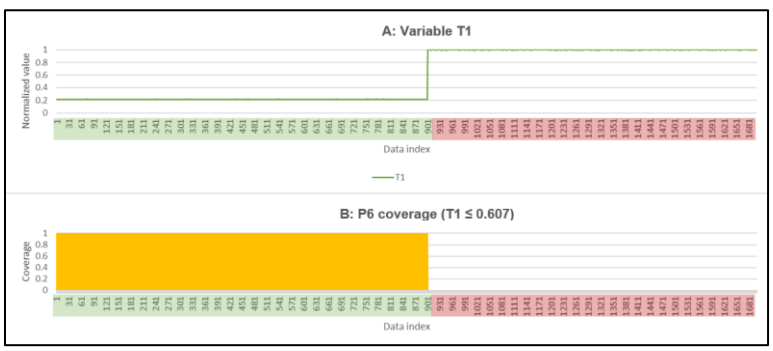


Figure 4.10 Illustration of the solution S_2^+

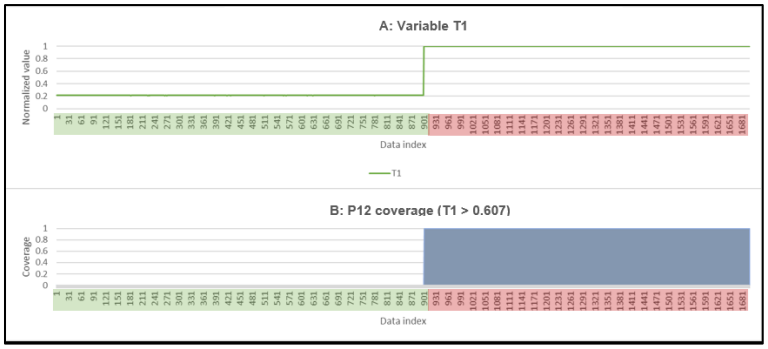


Figure 4.11 Illustration of the solution S_1^-

Stage 3-ILTA model construction

Figure 4.12 presents the constructed ILTA model. It visualizes the system causality that characterizes the normal and fault classes in a three-layer logic tree. At the solution layer, S_1^+ and S_2^+ (respectively, S_1^-) are connected to the normal class (respectively, the fault class) using the OR gate. Similarly, at the pattern layer, P_1 and P_2 are connected to S_1^+ using an OR gate; and P_6 is directly connected to S_2^+. However, at the condition layer, the two conditions $(T1 \leq 0.219)$ and $(F > 0.228)$ are connected to P_2 using the AND gate; and P_{12} is connected to the fault class without any logic gate because it is alone to explain that class.

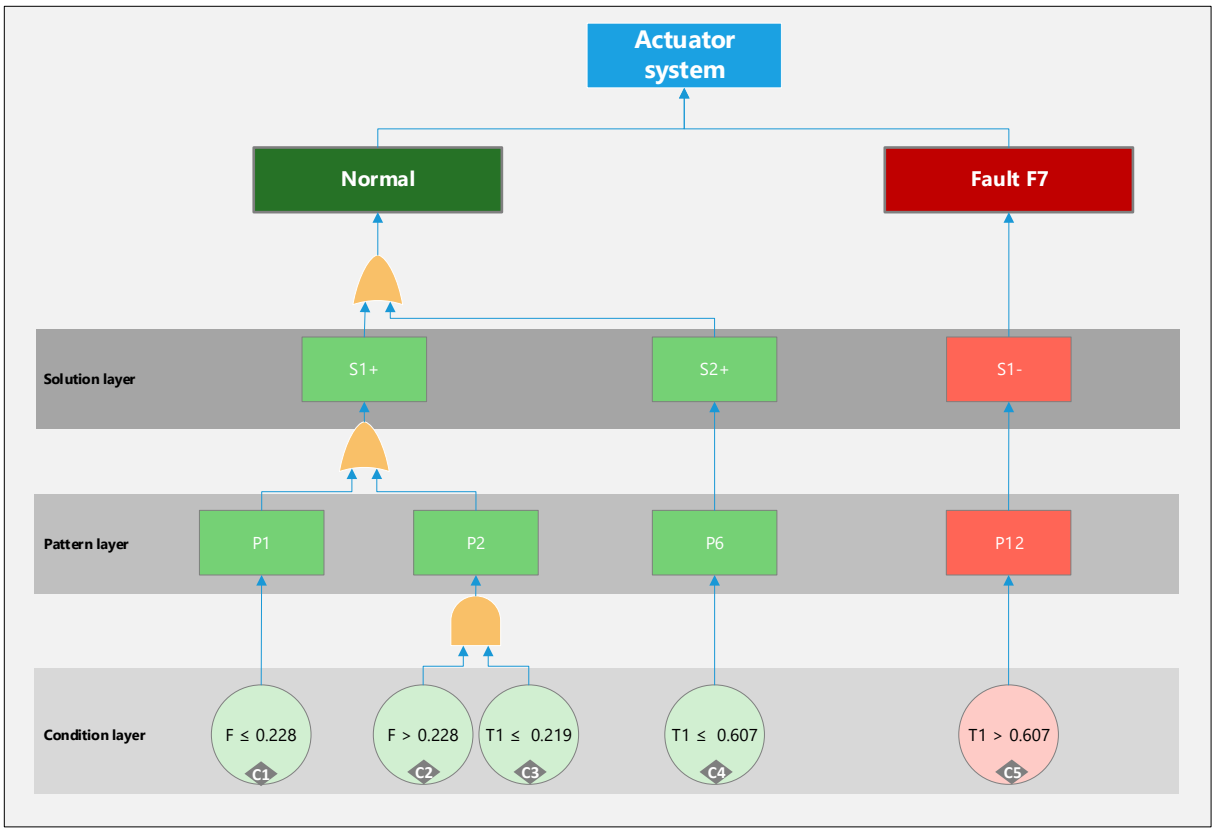


Figure 4.12 The one-level ILTA model of the actuator system

*T*he obtained ILTA model has the ability to visualize the system causality in a one-level logic tree in which the conditions, patterns and solutions layers are simultaneously represented. It explains how the conditions, patterns and solutions are combined to interpret the normal and fault classes using only one model.

Stage 4- Probability calculations

Table 4.6 summarizes the results of the probability calculations of the ILTA model. The equations (1), (3), (4) and (5) are used to estimate the probability of each condition, pattern, feasible solution and class event, respectively.

Table 4.6 Probability results at each layer of the one-level ILTA model

Condition layer	*Condition#*	$C_1: F \leq 0.228$	$C_2: F > 0.228$	$C_3: T1 \leq 0.219$	$C_4: T1 \leq 0.607$	$C_5: T1 > 0.607$
	$\mathcal{P}(C_\#)$	*0.09*	*0.91*	*0.53*	*0.53*	*0.47*
Pattern layer	*Pattern#*	$P_1: C_1$	$P_2: C_2 \cap C_3$ *or* $P_2: C_3 \cap C_2$		$P_6: C_4$	$P_{12}: C_5$
	$\mathcal{P}(P_\#)$	*0.09*	*0.44*		*0.53*	*0.47*
Solution layer	*Solution#*	$S_1^+: P_1 \cup P_2$			$S_2^+: P_6$	$S_1^-: P_{12}$
	$\mathcal{P}(S_\#)$	*0.53*			*0.53*	*0.47*
	Class#	*Normal:* $S_1^+ \cup S_2^+$				*Fault:* S_1^-
	$P(CL_\#)$	*0.53*				*0.47*

At the pattern layer, note that the pattern P_2 consists of the conjunction of $C_2: F > 0.228$ and $C_3: T1 \leq 0.219$. The probability $\mathcal{P}(P_2) = 0.44$ can be estimated using $\mathcal{P}(C_2|C_3)\mathcal{P}(C_3) = 0.836\mathcal{P}(C_3)$ or $\mathcal{P}(C_3|C_2)\mathcal{P}(C_2) = 0.483\mathcal{P}(C_2)$.

4.5.3 ILTA model for system control

This section discusses the usefulness of the proposed ILTA model from the perspective of system control. The ILTA model allows controlling the probability of the system event by tuning the probabilities of its root-causes. Two categories of control rules (full and partial) can be distinguished. The full control rule uses the solutions in which all their conditions are pure. The partial control rule uses solutions with a mixture of pure and fuzzy conditions or solutions with only fuzzy conditions. Figure 4.13 presents the system control rules that characterize the two classes. For examples, it is easy to note from Table 4.6 that the probability of the normal-class is fully controlled by the probability of its pure condition $C_4: T1 \leq 0.607$ when the value of $T1$ is tuned from 0 to 1, according to Rule 1: $\mathcal{P}(CL_{normal}) = \mathcal{P}(S_2^+) = \mathcal{P}(P_6) = \mathcal{P}(C_4)$. In the same manner, the probability of the fault-class is controlled by the probability of the pure condition C_5 according to Rule 4. However, because C_2 and C_3 are two fuzzy conditions, they can control in part the probability of the normal-class according to the rules 2 and 3.

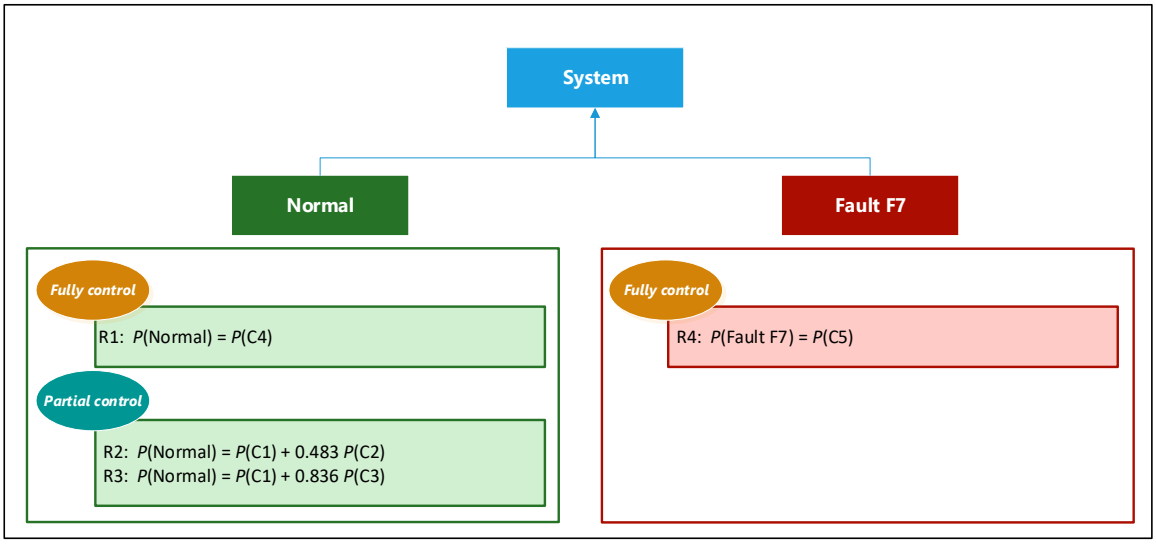


Figure 4.13 The one-level ILTA control rules

4.5.4 Validation of the ILTA model

In this section, the obtained ILTA model is validated through the prior knowledge about the occurrence of the fault *F7* in the data generation tool and the mean error of the predicted probability of a given class based on the testing dataset.

According to the fault simulation equation given in (DAMADICS, 2002), the main cause of *F7* is related to the fluid temperature $T1_f$. In the normal operation, the fault strength value f_s is equal to 0 and the fluid temperature $T1_f$ is normal, equal to $T1_0$. The fault *F7* occurs when the $T1_f$ exceeds $T1_0$. This happens when the fault strength f_s is greater than 0.

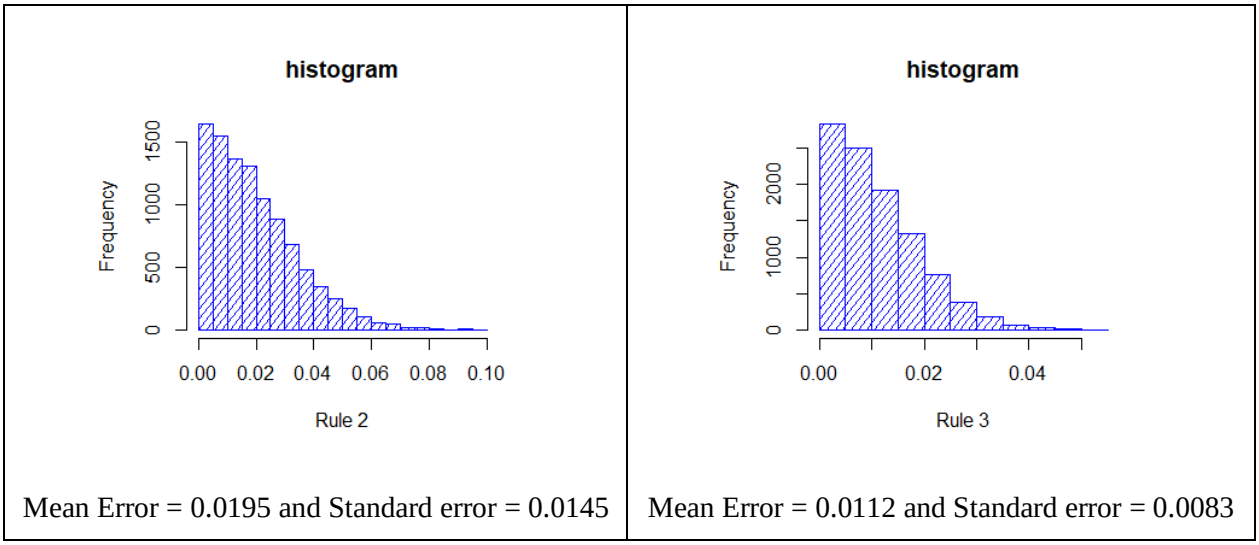
$$Fault\ (F7)\ simulation: \begin{cases} T1_f = T1_0 & if\ \ f_s = 0 \\ T1_f = T1_0 + 200 + 100 * f_s & if\ \ f_s > 0 \end{cases}$$

$$\begin{aligned} &(T1_f) fluid\ temperature \\ &(T1_0)\ nominal\ fluid\ temperature \\ &(f_s)\ fault\ strength \end{aligned} \qquad (6)$$

From the ILTA model, the two pure conditions C_4 and C_5 detects the changing within the data based on the fluid temperature $T1$. According to S_2^+, if $T1 \leq 0.607$ then the actuator operates normally. However, according to S_1^-, if $T1 > 0.607$ then the fault *F7* is detected. Thus, the ILTA

methodology detects the fault *F7* in the effective way. Nevertheless, the ILTA model proposes another feasible solution (S_1^+), which cannot be verified using the equation (6). S_1^+ involves $T1$ and the disturbed medium flow (*F*) to characterize the normal operatin state. Therefore, the human expert is free to keep or remove S_1^+ since this solution may represent useful or redundant causality knowledge about the normal state.

From another side, we use 1000 random data samples generated from the testing dataset of 720 observations (see Table 4.4) to estimate the mean error between the predicted probability of a given class based on its control rules and the actual probability of the same class. Each random data sample has a fixed size of 314 observations with 95% of confidence level. The mean and the standard errors of the two fully control rules 1 and 4 are equal to zero, which indicate that the solutions S_2^+ and S_1^- fit well the data. Because each one of the above solutions depends on only pure condition. Figure 4.14 provides only the error distribution for the rules 2 and 3. The mean and standard errors are less than 2% and 1%, respectively.



Mean Error = 0.0195 and Standard error = 0.0145 | Mean Error = 0.0112 and Standard error = 0.0083

Figure 4.14 The performance of the control equations

4.5.5 Discussion

After the validation of the ILTA model, the discovered root-causes can be ranked according to their importance in controlling the probability of the corresponding class (Table 4.5). The ranking of the root causes (conditions) is based on the associated conditional probability value that describes the occurrence of the involved pattern in the control rule. For instance, the conditions $C_1: F \leq 0.228$; $C_4: T1 \leq 0.607$ and $C_5: \mathrm{T1} > 0.607$ have the highest importance (equal to 1) due to their

pureness. However, $C_3: T1 \leq 0.219$ and $C_2: F > 0.228$ have the lowest scores 0.836 and 0.483. The ILTA root-causes ranking is similar to Birnbaum importance measurement (Miziula & Navarro, 2019), which ranks the basic event based on their contribution on modifying the total risk level.

Table 4.7 The root-causes ranking for normal and fault classes

Normal-class			
Solution	Root-cause	Rule	Ranking
S_2^+	C_4: T1 $\leq$ 0.607	1	1
S_1^+	C_1: F $\leq$ 0.228	2 or 3	1
	C_3: T1 $\leq$ 0.219	3	0.836
	C_2: F $>$ 0.228	2	0.483

Fault-class			
Solution	Root-cause	Rule	Ranking
S_1^-	C_5: T1 $>$ 0.607	4	1

Hence, the proposed ILTA model is effective in the actuator case because the generated database represents a simple system in which the root-causes have a direct influence on the system states. Thus, the ILTA model can easily represent the actuator causality in one-level logic tree. Accordingly, if the responsible variables for increasing the fluid temperature $T1_f$ are included in the database, then the ILTA model will easily select the fault indicator (T1) of the actuator regarding *F7*.

However, in case of complex systems, such as nuclear power plan (Vesely, Goldberg, Roberts, & Haasl, 1981) and aerospace systems (Stamatelatos et al., 2002; Stamatelatos. & Dezfuli., 2011), where several sub-systems and characteristic variables may have interdependency relations between each other, the fault diagnosis using one-level ILTA model is insufficient. To overcome this challenge, a multi-level ILTA model is necessary to address complex causality. Therefore, the one-level ILTA approach proposed in this paper will be employed as a cornerstone module in building the multi-level tree for complex causality analysis. Furthermore, after addressing the diagnosis problem, the big challenge remains the prognosis of the future health conditions of the system over the time. Consequently, the ILTA methodology needs to be improved by including the time as an inherent global variable to the system aging. Also, this future development aims to discover and represent the temporal dependencies between the system events.

4.6 Conclusion

This paper developed an effective causality analysis methodology, named interpretable logic tree analysis (ILTA). It addresses the diagnosis of an undesirable event, fault or failure based on the probabilities of the discovered root-causes. The ILTA methodology builds a logic tree directly by exploiting the database, without the involvement of human expertise in the construction stage. Intuitively, it visualizes the extracted knowledge in the form of an interpretable logic tree. This way, the analysts can view hidden system causalities, interpret them, explore their impacts on system behaviour and control the system states.

The ILTA model is an un-bias interpretable logic tree against the expert knowledge. Unlike the FTA, CAFT and FTKD models, the ILTA model does not involve the expert to construct the FT. The expert has to analyze the obtained FT. It offers more flexibility to address different facets of the system. It can diagnose any event that characterizes the system's behaviour. Moreover, the ILTA methodology can generate multiple feasible solutions and system control rules to help the analyst tune the system's behaviour. It offers several alternatives for system modelling, in addition to many explanations of how the knowledge discovered in the dataset is fully exploited to describe the system states.

However, the ILTA model needs a representative data sample of the system states, similar to any data-driven model, to provide relevant and accurate results. If the data sample does not contains sufficient information about the system states, then the ILTA methodology will construct a partial interpretable logic tree that reflects only the causality within that sample. In such case, the experts should double their efforts to complete the obtained ILTA model by adding the unobserved scenarios due to the limitation of the sample representability.

CHAPTER 5 ARTICLE 2: MULTI-LEVEL INTERPRETABLE LOGIC TREE ANALYSIS: A DATA-DRIVEN APPROACH FOR HIERARCHICAL CAUSALITY ANALYSIS

Kerelous Waghen & Mohamed-Salah Ouali*

Mathematics and Industrial Engineering Department, Polytechnique Montréal

2500 Chemin de Polytechnique, Montréal, QC, H3T 1J4, Canada

kerelous.waghen@polymtl.ca; mohamed-salah.ouali@polymtl.ca

*Member of CIRRELT (https://www.cirrelt.ca) and IVADO (https://ivado.ca)

Submitted at Expert systems with Application journal

(Manuscript Number : ESWA-D-19-04282)

Abstract:

This paper presents a data-driven approach for a hierarchical causality analysis of faults in a complex system, named a multi-level interpretable logic tree (MILTA). From a representative faults dataset, this approach constructs dependent trees that explain the relation structure between the root-causes, intermediate causes and faults with the minimum expert involvement. The MILTA model combines the discovered knowledge in dataset (KDD) in the form of feasible solutions and the fault tree analysis (FTA), level after level, as long as the root-causes are not completely uncovered. A burn-and-build algorithm is developed to maximize the representability of the feasible solutions with a minimum number of patterns. Using Bayes' theorem, the hierarchical causality between the root-causes and the fault is captured through different causality rules that quantify the effects of the root-causes on the fault occurrence. An actuator system dataset that consists of complex fault and normal operation states is used as an illustrative example. The MILTA model finds the same documented root-cause and uncovers other root-causes with higher accuracy.

Keywords: Fault Diagnosis, Causality Analysis; Knowledge Discovery in Dataset; Fault Tree; Complex System.

5.1 Introduction

The inherent complexity of industrial systems makes a fault diagnosis (FD) problem very difficult to apprehend due to the variability of usage conditions, the noisy environment and the interdependency relation between the system failure modes (Duan, Hu, & Lin, 2017). Peng, Ma, and Zhang (2017) highlight the importance of FD in such complex systems to ensure their sustainability and to avoid expensive maintenance costs. The FD consists of three main tasks: fault detection, fault isolation and fault identification. The causality analysis (CA) aims to accomplish these three tasks by identifying the fault occurrence by finding its root-causes. The CA is commonly accomplished using elicitation, event-based or data-driven methods. The elicitation methods take reporting forms to record relevant information about the causes of complex incidents (Johnson, CW, 2002). Event-based methods use graphical constructed tools to describe the occurrence of a top incident based on their intermediate events and root-causes (Leimeister & Kolios, 2018). On the other hand, the data-driven methods extract the fault causality knowledge by mapping the relations between the system input and output records (Li, P. c., Zhang, Dai, & Li, 2017). These methods involve human experts from different engineering domains to model and construct the CA models and to interpret the obtained fault causality structure knowledge within a complex system. They may lead to partial CA in cases in which there is a lack of expertise, depth of analysis or the non-representability of the historical data.

Fault tree analysis (FTA) is common event-based method that helps engineers intuitively identify the faults of a complex system, find their root-causes and quantify their effects on system behaviour in terms of risks (Srivastava & Sinha, 2012). They are used first at the system design stage and then revised periodically when new faults are reported during usage (Jin-San, Jin-Sung, Jae-Goo, & Feel-soon, 2017). Building a Fault Tree (FT) model for a complex system is a difficult and time-consuming task because a fault may be related to the dependent root-causes and unstructured combination of intermediate events that could explain it. Furthermore, expert knowledge can become limited and biased when eliciting the causality structure of a complex fault, even if its root-causes are known.

Knowledge Discovery in Dataset (KDD) offers an innovative modeling perspective of fault diagnosis in a complex system from historical data. Ragab, Ahmed et al. (2018) provide a new methodology for automatic enrichment and updating existing FT in order to achieve accurate fault

detection and isolation in industrial processes with minimal human involvement. The authors discuss the main challenges in combining qualitative FT, descriptive machine learning techniques and expert knowledge to deep understanding abnormal events in real situations. The KDD offers to experts some counterintuitive solutions that enrich their prior root-causes analysis. Waghen, and Ouali (2019) develop a data-driven fault diagnosis methodology, called Interpretable Logic Tree Analysis (ILTA), for automatic FT construction and root-cause probability assessments based only on historical data, as well as a supervised machine learning technique, named logical analysis of data (LAD) (Boros, E. et al., 2011). In a one-level tree, the ILTA model depicts the factual combinations of root-causes, in the form of feasible solutions, which explain the fault event. Bayesian probability rules are derived from the ILTA model to control the occurrence of the fault event based on the probabilities of its root-causes. However, the one-level ILTA model may hide the hierarchical structure of intermediate events (causes) that may occur between the fault event and its root-causes in a complex system.

This paper proposes a multilevel ILTA model (MILTA) that would be able to discover the hierarchical structure of causes within a complex system from a set of representative historical data. The MILTA model uses the ILTA methodology (Waghen & Ouali, 2019) as a construction cornerstone of the multilevel model with minimal involvement of a human expert. The MILTA model searches for all feasible combinations of intermediate causes that explain the fault event in cascade, level after level until the root-causes discovery. Thus, the discovery of these sequences of causes will define the causality structure between the fault event and its root-causes. The MILTA model will construct non-redundant knowledge within the logic tree so that each identified cause at one level will be explained at the subsequent level, and so on. This decomposition will be stopped when there is no more available information in the dataset that could explain the final discovered causes, or when the coverage of the feasible combinations of causes at that level is less than a minimum acceptable threshold. Finally, a set of causality rules is deduced from the final MILTA model in order to describe the structure of the causal sequences between the fault event and its root-causes.

The remainder of this paper is organized into six sections. Section 2 reviews the available methods of CA used in the fault diagnosis of a complex system; the main challenges are underlined and discussed. Section 3 presents the MILTA methodology to construct a logic fault tree and discover the causality sequences between the fault event and its root-causes. Section 4 presents a two-state

illustrative example of an actuator system dataset and discusses the obtained MILTA model. Section 5 focuses on the validation of the MILTA model from the point of view of human expertise and model accuracy. Section 6 concludes the paper and discusses the pros and cons of the proposed MILTA model.

5.2 CA methods for fault diagnosis

A fault is a physical defect that occurs within any component of the system due to natural ageing or abnormal load and usage conditions, which leads to the deviation of system behavior (error) and may cause its failure later (Choi, Edwards, Ko, & Kim, 2016). Lower productivity and quality or total service outage may occur when faults are not diagnosed and removed. The fault diagnosis procedure first detects the abnormal situations (faults) that happen during normal operations, then isolates each fault by identifying its location and causes, and finally quantifies the effects of causes on the occurrence of the fault (Willersrud, 2015). CA methods aim to discover the cause-effect sequences that rely on the causes to the selected fault that serve those procedures. Three categories of CA methods can be distinguished in the literature: the elicitation, event-based and data-driven methods. In what follows, these methods are discussed and their strengths and limitations are summarized help clarify the research gap.

The elicitation method is a high-level framework of thinking about the number of possible causal factors that explain a selected event. It takes the form of an analysis report that collects and describes the evidence and its causes to better understand an event's occurrence. Johnson, Chris (2003) discusses a barrier analysis method to understand the event occurrence, elicit the appropriate prevention and assign a flow path between the event and its consequences. Ramzali, Lavasani, and Ghodousi (2015) analyze the possible barriers in terms of success or failure to prevent the failure of oil and gas drilling systems in offshore oil rig activities. Noh, and Shortle (2018) propose an automatic approach for identifying the best candidates' barriers to include in a risk model related to aviation safety. Wu, G. G., Yang, Song, and Li (2016) develop a quick causality analysis tool for complex faults in engine systems based on the change analysis elicitation method. Wu, G. G., Yang, Li, and Song (2017) propose a 3CA change analysis method that integrates other elicitation methods such as first principle-best practices, barrier failure analysis and prioritization rating code (PRC) matrix to perform control analysis.

The event-based method focuses on representing the interaction between causes that lead to the occurrence of the undesired event. Therefore, it enables a human expert to identify the reasons that explain what happens using graphical tools. The event tree analysis (ETA) and fault tree analysis (FTA) are the two most common event-based methods for fault diagnosis. The ETA is an inductive causality analysis diagram that tracks the evolution of low-level events towards the main event based on the conditional probabilities (Riyadi, 2014). Picoco et al. (2017) develop dynamic event trees (DETs) for fault diagnosis in a nuclear power plant, based on the simulation of generated scenarios, rather than rely directly to experts. Hidayat, and Hermansyah (2018) perform an ETA in a complex gas pipeline distribution network. The proposed tree tracks the pivotal causes by constructing a multilevel tree to identify the possible failure scenarios. The FTA is a top-down deductive causality analysis chart that uses Boolean logic gates to decompose the main event to its possible root-causes. Cao et al. (2019) propose a methodology for extracting belief rules base (BRB) from the constructed FTA model. The BRB rules summarize the FTA in a set of interpretable IF-Then rules and use them in a complex fault diagnosis. Kumar and Ghosh (2017) combine the FTA and ETA models to better identify the dependent fault causes in mining systems. Piadeh et al. (2018) develop a framework that integrates both an FTA and ETA model to identify the fault and recovery actions in wastewater treatment systems, assess the reliability of fault detection and prioritize the recovery actions.

The data-driven causality analysis methods explore historical data in order to discover the physical relations and influences between system components. They aim to describe the statistical dependencies between variables and the system performances. Direct graph (DG), decision tree (DT), Granger causality (GC) and Bayesian network (BN) are the most common data-driven methods that graphically diagnose the fault in complex systems. Ma, and Li (2017) combine a signed directed graph (SDG) and neighbourhood rough set (NRS) technique to improve the efficiency of the fault diagnosis in complex systems. The SDG nodes are fuzzified to describe the variables states and overcome low diagnosis accuracy. The SDG paths are transcribed in the form of a decision table and introduced into the NRS rule extraction module. Meckel, and Obermaisser (2018) propose a component-based framework, called the diagnostic directed acyclic graph (DDAG), which allows for a fault diagnosis of dependent components using different adaptive diagnostic modules of the system inputs.

The DT is a descriptive machine learning method that is able to perform interpretable fault diagnosis thanks to the simplicity of its graphical representation (Alkinani, Al-Hameedi, Dunn-Norman, Alsaba, & Amer, 2019). Ashok, and Yadav (2018) develop a DT diagnosis model of electrical transmission lines using the discrete cosine transforms of the current and voltage signals. Yu et al. (2018) introduce a hybrid approach that combines expert knowledge and DT to diagnose the faults in a variable refrigerant flow system. The DT first extracts the causality structure within the system, then the human expert enriches it to make it more representative. The GC is a common causality analysis method that uses an interpretable graphical causality tree using time series inputs (Amblard & Michel, 2013). Chen, H.-S., Yan, Zhang, Liu, and Yao (2018) propose a multivariate GC analysis methodology, which measures the degree of dependency between the process variables in order to construct a causal map between them and extract the maximum spanning tree to easily identify the system root-causes. Alizadeh, Esmaeil, El Koujok, Mohamed, Ragab, Ahmed, and Amazouz, Mouloud (2018) develop a multivariate time series causality analysis tool for fault diagnosis based on time series stationarity tests and GC. The tool provides a pairwise causality relationship graph validated by the Tennessee Eastman Process benchmark (Bathelt, Ricker, & Jelali, 2015). The BN is a probabilistic directed graph that is able to model the influence and dependency between the system variables (Yazdi, 2019). Gao, J., and Zhao (2018) model the fault propagation pathways using a multilevel BN. The model first extracts the strong causality relationship between the nonstationary variables that later linked the global causality model (multi-level BN) for connecting the stationary variables and nonstationary variables. Chen, X., Wang, and Zhou (2018) develop a BN prediction model of system alarms and apply it to the Tennessee Eastman Process benchmark.

With regards to the three categories of causality analysis methods presented above, the following limitations might arise when dealing with fault diagnosis in a complex system. The elicitation methods involve an important effort from human experts to provide specific details about the causality structure of causes. The experts use a nonstandard form of elicitations with different levels of detail, which involve advanced artificial intelligence techniques to extract valuable decision knowledge. In such cases, the causality structure may be superficial or incomplete. The event-based models improve the previous methods because they involve standard forms of elicitation. However, the methods remain very dependent on prior human knowledge and their domains of expertise. They may lead to partial CA models in a case where faults are omitted. The

data-driven methods are less dependent on human expert knowledge to build the causality models compared to the elicitation and event-driven methods. The obtained causality graphs or charts mainly describe the information flow and dependency given by the system variables rather than the fault's hierarchical decomposition into its causality structure. Therefore, human experts should recover this problem, derive the system root-causes and interpret the causality structure from those data-driven models.

To overcome the above limitations, a data-driven fault tree methodology, called the interpretable logic tree analysis (ILTA), is proposed by Waghen and Ouali (2019) as the first work in addressing the causality analysis in complex systems. The novelty behind the ILTA model is to use the FT as a graphical representation and interpretable analysis technique, which assists an expert in grasping the fault causality from KDD. The discovered root-causes are interconnected using logic gates to explain a selected fault in the system. This model deals with a direct or one level cause-effect relation between the root-causes and the fault event. To discover the hierarchical structure of causes in a complex system, a multilevel ILTA model is proposed in the following section.

5.3 MILTA Methodology

Figure 5.1 depicts the three-phase methodology used to build the architecture of the MILTA model and quantify its elements from root-causes to the fault's events. Phase 1 iteratively uses the ILTA methodology to build and quantify the one-level logic tree based on a given labeled dataset. Phase 2 constructs as many levels as there are unexplained causes in the logic tree. Phase 3 derives the causality rules that characterize the causality structure between the root-causes, intermediate causes and faults.

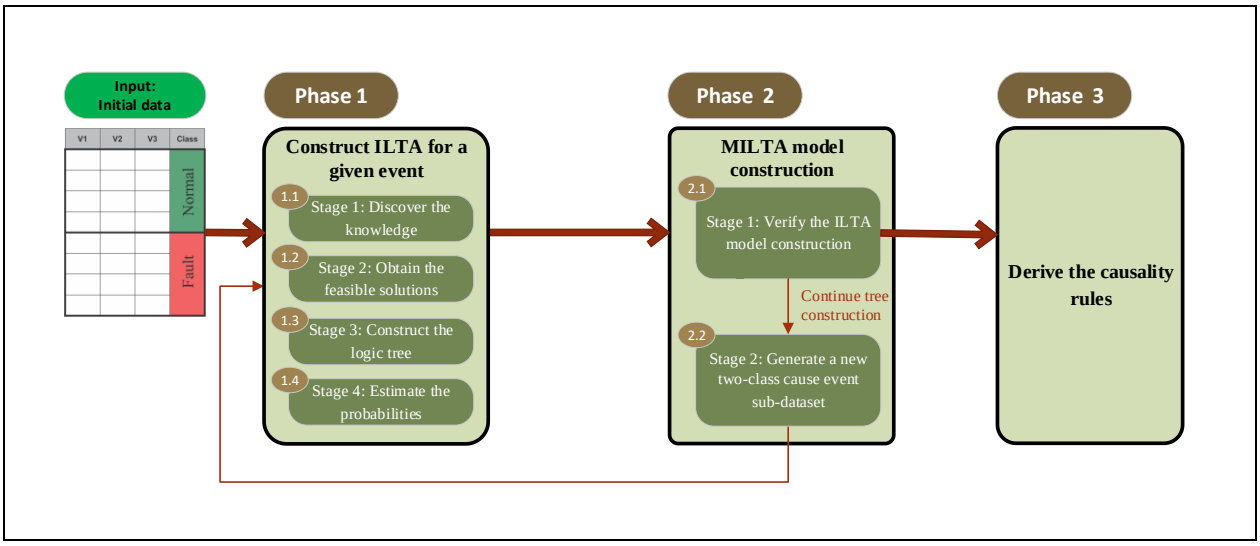


Figure 5.1 The three-phase MILTA methodology

5.3.1 Phase 1-One level ILTA model construction

Step 1 considers a labeled dataset in which each label represents a specific class event (CL) and the variables depict the indicators associated with an observation. Each indicator may reflect the effect of a cause or a root-cause that may explain a given fault class. Four stages are required to build a one level ILTA model: discover knowledge, obtain feasible solutions, construct the logic tree and estimate the probabilities (Waghen, and Ouali (2019)).

- ***Stage 1: Discover knowledge.*** The knowledge is extracted in the form of patterns within the dataset using the two-class LAD as a pattern generation and selection technique (Alexe & Hammer, 2006). Each pattern (P) is a conjunction of some conditions that discriminate one class (CL) of observations from the other classes. Each condition (C) compares the value of the variable (indicator) with the cut point that represents the transition between two different classes of observations (condition: variable ≤ (or >) cut point). Thus, each pattern can totally or partially cover certain observations of the same class. The percentage of coverage characterizes the knowledge represented by that pattern. In turn, the observations belonging to the same class can be covered by more than one pattern, inducing an overlap between the coverage of these patterns. The percentage of the overlap corresponds to the number of observations that are covered simultaneously by at least two patterns over the total number of observations of the class. This percentage represents the redundant knowledge within that class.

- ***Stage 2: Obtain feasible solutions.*** The solution is defined as a combination of certain patterns that cover the observations of the same class. Each solution could be characterized by its coverage (Cov) and overlap (OL) percentages. The Cov denotes the percentage of observations that are covered by any pattern in the solution formed, whereas the OL denotes the percentage of observations that are covered by more than one pattern in the solution. In other words, the Cov corresponds to the percentage of observations covered by the union of the solution's patterns, while the OL is the percentage of observations covered by the intersection of the solution's patterns. The feasible solutions are a group of solutions that deploy a minimum number of patterns while maximizing the coverage percentage and minimizing the overlap percentage at the same time. Therefore, these feasible solutions will maximize the representability of the discovered knowledge while reducing their redundancy in the ILTA model using the minimum knowledge. The search for these feasible solutions involves an improved burn-and-build algorithm. Note that the new burn-and-build algorithm is slightly different from the original one presented in (Waghen and Ouali, 2019). It searches for the feasible solutions that comprise a minimum number of patterns. This improvement aims to provide sequential one level logic trees that may explain the hierarchical causality in the MILTA model. The sub-step 1.3 of the improved algorithm selects the optimal combination for each starting pattern. The step 4 compares the selected combinations with respect to different starting patterns.

Improved burn-and-build algorithm: ***Searching for feasible solutions comprised of a minimum number of patterns***
Input. *i) Set of generated LAD's patterns: $P_{gen} = \{P_1, P_2, \ldots, P_l\}$; where (l) is the number of the discovered patterns* *ii) Overlap Threshold* ***Step 1.*** *Select a start pattern (P_i) and calculate its coverage* *1.1. Remove the overlapped patterns with P_i based on preset overlap threshold* *i. At number of combination (n) = 2* *ii. Select the combination with P_i that has the maximum coverage* *1.2. Repeat (the sub-steps 1.1-i and ii) until the number of combinations (n) = number of the discovered patterns*

1.3. Compare the selected combinations at each n and select the combination that maximizes the coverage with a minimal number of patterns

Step 2. *Select another pattern (P_i) as a start point and repeat 1.1, 1.2 and 1.3.*

Step 3. *Repeat 2 until considering each pattern as a start point.*

Step 4. *Compare the selected combinations over the start patterns P_i and select the combination that maximizes the coverage with a minimal number of patterns*

Output.

Set of feasible solutions: Sol = {S_1, S_2,...., S_k }; where (k) is the number of the formed solutions.

- ***Stage 3: Construct the logic tree.*** The feasible solutions found at Stage 2 are visualized in a logic tree consisting of three distinguished layers related to the solution, pattern and condition using the OR and AND gates. At the condition layer, all of the conditions that form a pattern are connected to that pattern using an AND gate. At the pattern layer, all of the combinations of patterns that constitute a feasible solution are connected to that feasible solution using an OR gate. Similarly, all of the feasible solutions that explain a given class are connected to that class using an OR gate.

- ***Stage 4: Estimate the probabilities.*** Based on the constructed logic tree, the occurrence probability of each class event $\mathcal{P}(CL)$ is estimated using the occurrence of its constituent feasible solutions $\mathcal{P}(S_q)$ $q = 1,2, \dots Q$, patterns $\mathcal{P}(P_j)$ $j = 1..J$ and conditions $\mathcal{P}(C_i)$ $i = 1..I$ as follows. The probability of a given C_i $i = 1..I$ condition is estimated using the ratio of the number of observations N_i covered by C_i and the total number of observations N_T in the dataset sample, $\mathcal{P}(C_i) = \frac{N_i}{N_T}$ (Eq. 1). The probability of a pattern P_j $j = 1..J$ corresponds to the probability of the intersections of its conditions C_i $i = 1..I$, $\mathcal{P}(P_j) = \prod_{i=1}^{n_j-1} \mathcal{P}(C_i|C_{i+1}).\mathcal{P}(C_{i+1})$ (Eq. 2) where $\mathcal{P}(C_i|C_{i+1})$ represents the degree of dependency between the two conditions C_i and C_{i+1} with respect to the occurrence of C_{i+1}. The probability of a solution S_q $q = 1,2, \dots Q$ corresponds to the probability of the union of its patterns P_j $j = 1..J$, $\mathcal{P}(S_q) = \mathcal{P}\left[\cup_{j=1}^{J} P_j\right]$ (Eq. 3). Thus, the probability of a class event is equal to the probability of the union of its feasible solutions S_q $q = 1,2, \dots Q$, $\mathcal{P}(CL) = \mathcal{P}\left[\cup_{q=1}^{Q} S_q\right]$ (Eq. 4).

Figure 5.2 depicts a typical example of the one level ILTA model architecture based on the discovered knowledge from a two-class dataset. The model represents the causality structure of the two classes $CL(+)$ and $CL(-)$ in three layers. One feasible solution S_1 explains the causality of class $CL(+)$ with its two patterns P_1 and P_2 and three conditions C_1 to C_3. While, two feasible solutions S_2 and S_3 characterize $CL(-)$. S_2 consists of one pattern P_3 with one condition C_4. However, S_4 combines two patterns P_4 and P_5. P_4 consists of one condition C_5 and P_5 conjuncts C_6 and C_7.

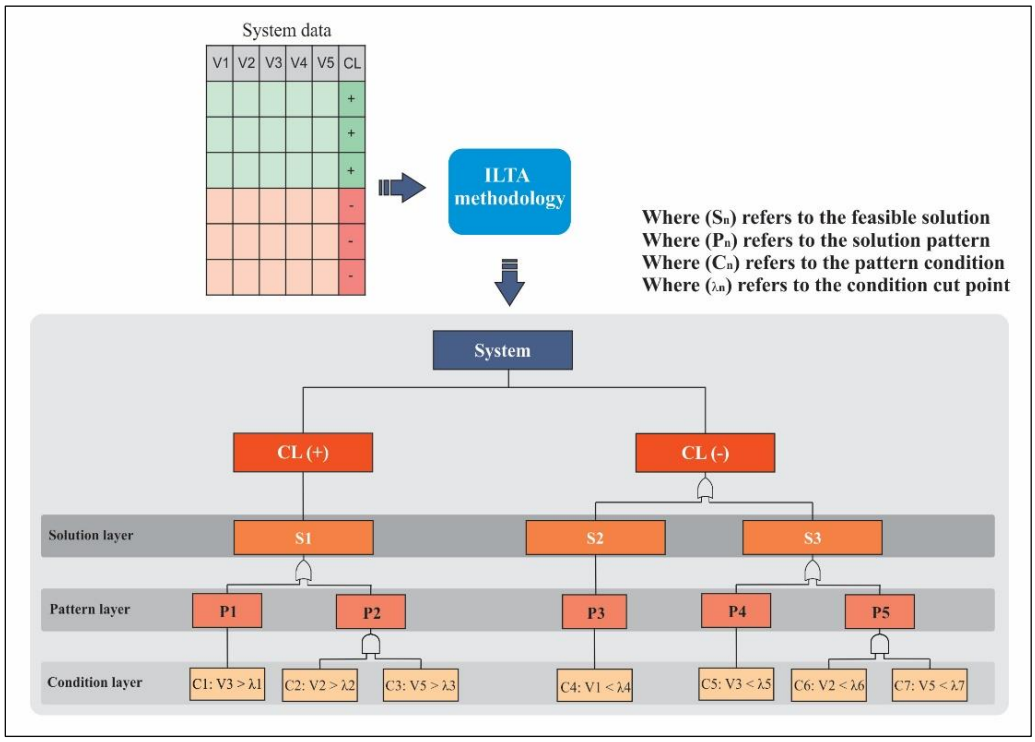


Figure 5.2 A typical example of the ILTA model

In a case in which the dataset represents a one cause-effect relation between the variables and the fault event, the conditions denote the root-causes and only one level ILTA model is required to diagnose the fault event. However, when the dataset hides different cause-effect sequences, the extracted patterns at a given level may involve intermediate causes rather than the root-causes. In that case, the one level ILTA model is not able to capture the causality structure. Each variable that appears at the condition level of the logic tree needs to be further explained until finding the root-causes. To address this issue, we need to consider each variable shown at the condition level as a new two-class sub-dataset and try to find the feasible solutions that explain the occurrence of each

variable class. Using the improved burn-and-build algorithm of Phase 1 – Stage 2, some selected feasible solutions that may characterize different causality relations may be found and represented in the MILTA model. Other unselected solutions that do not satisfy the coverage and the overlap criteria may hide significant amounts of causality knowledge. Consequently, this ignored knowledge may be rediscovered in the lower decomposition levels of the logic tree with stronger reformed feasible solutions.

5.3.2 Phase 2-The MILTA model construction

The MILTA model assembles, in a sequential up-bottom structure, several ILTA models by each unexplained cause at a given level as a new two-class cause event. Phase 2 consists of two stages: verify the ILTA model construction and generate a new two-class cause event sub-dataset when further variables are available to explore the cause event at the condition layer.

- ***Stage 1: Verify the ILTA model construction.*** This stage verifies the quality of the knowledge representability at the solution layer in the last ILTA constructed model and decides whether further decomposition of its conditions are needed or not. This quality of the represented knowledge is characterized by the coverage extended by each feasible solution that may explain a given cause event in the ILTA model. It is important to avoid the decomposition of any low coverage feasible solution, which may lead to weak information branches to explain a given cause event in the MILTA model. It may not carry useful knowledge that will explain the cause event. Therefore, removing those low-quality feasible solutions from the start helps trim and prune the MILTA model in addition to the impact on its general depicted knowledge of cause events. Thus, the coverage of the feasible solution is utilized to measure the quality of the knowledge representability of the ILTA model at each decomposition level. Accordingly, stopping or exploring a cause in a further decomposition level is made based on its solution's coverage percentage compared to a given investigation threshold. For a given cause event, if any of its feasible solutions' coverage is less than this threshold, then the ILTA model associated with that cause is removed from the MILTA model. Therefore, only the feasible solutions with coverage that exceeds this threshold are considered in the subsequent decomposition level of the cause event. If no feasible solution respects the coverage threshold, the associated conditions with those feasible solutions are considered the final root-causes of

the fault event. Also, there is no further root-cause exploration if all of the variables are used in the previous decomposition levels of the fault event.

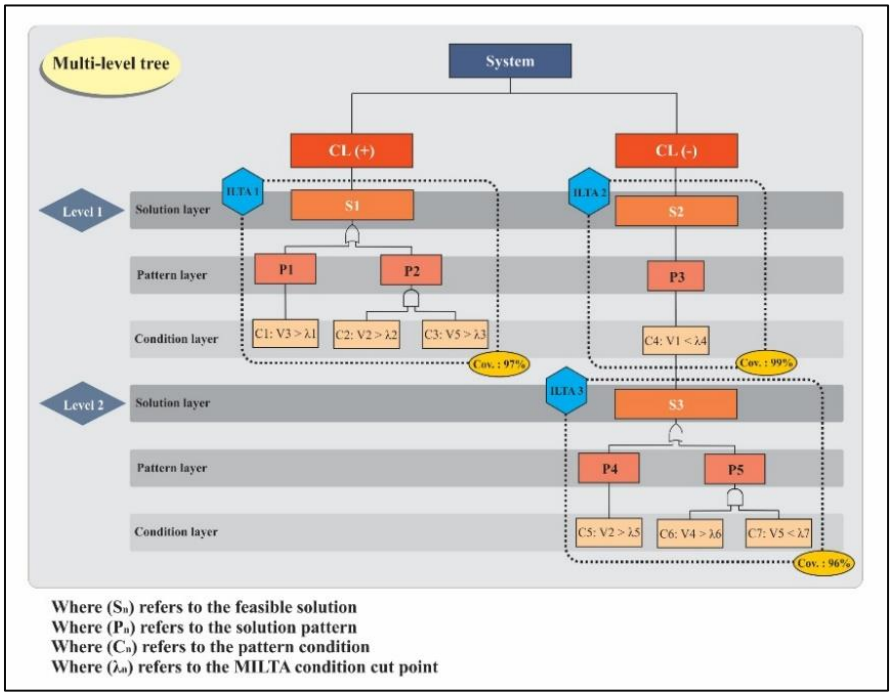


Figure 5.3 MILTA objective for a fault causality analysis

- ***Stage 2: Generate a new two-class cause event sub-dataset.*** In a case where the ILTA model contains intermediate causes at the condition layer, a new sub-dataset is generated for each involved condition and labeled in a two-class condition event based its variable cut point value. Figure 5.4 presents a simple example of a two-level ILTA model constructed from an initial dataset D_1. D_1 includes all of the variables that may explain a given fault event. Figure 5.4-A (iteration 1) builds the first level ILTA model. The model has only one feasible solution S_2, which consists of only one pattern P_3 that involves one condition C_4: $V_1 \leq \lambda_4$ to explain $CL(-)$. If the coverage of S_2 exceeds the expected investigation threshold, then the condition C_4: $V_1 \leq \lambda_4$ will be considered as a new cause event at the 2nd level of the MILTA model. Therefore, a new sub-dataset D_2 is generated for C_4. D_2 is labeled into (+) and (-) classes according to $V_1 \leq \lambda_4$ and $V_1 > \lambda_4$, respectively. D_2 has the same format as D_1 where the fault event is replaced by the new condition event. Then, D_2 is introduced to phase 1 of the methodology for a second iteration (Figure 5.4-B (iteration 2). The probability of each condition, pattern and feasible solution at each decomposition level is estimated using Stage 4 of Phase 1. It is important to

note that, if the feasible solution has more than one condition to explain, the generated sub-dataset for that condition excludes the variables involved with the other conditions.

5.3.3 Phase 3-Derive the causality rules

This phase extracts the causality rules (CR) in the form of conditional probability equations that characterize the cause-effect sequences from the root-causes to the fault event. At each decomposition level of the MILTA model, the probabilities of the conditions, patterns and feasible solutions are calculated using the equations defined in Stage 4 of Phase 1. Accordingly, using the logic gates of the final MILTA model, it is possible to derive the probability of a given fault event based on the probabilities of the conditions involved at the final decomposition levels. Such an equation expresses the causality rule between the fault, the intermediate causes and the root-causes. For example, from the final MILTA model of figure 5.4-B, only one causality rule is derived for $CL\ (-)$. CR1: $\mathcal{P}(CL\ (-)) = \mathcal{P}(C_5) + \mathcal{P}(C_7|C_6).\mathcal{P}(C_6)$.

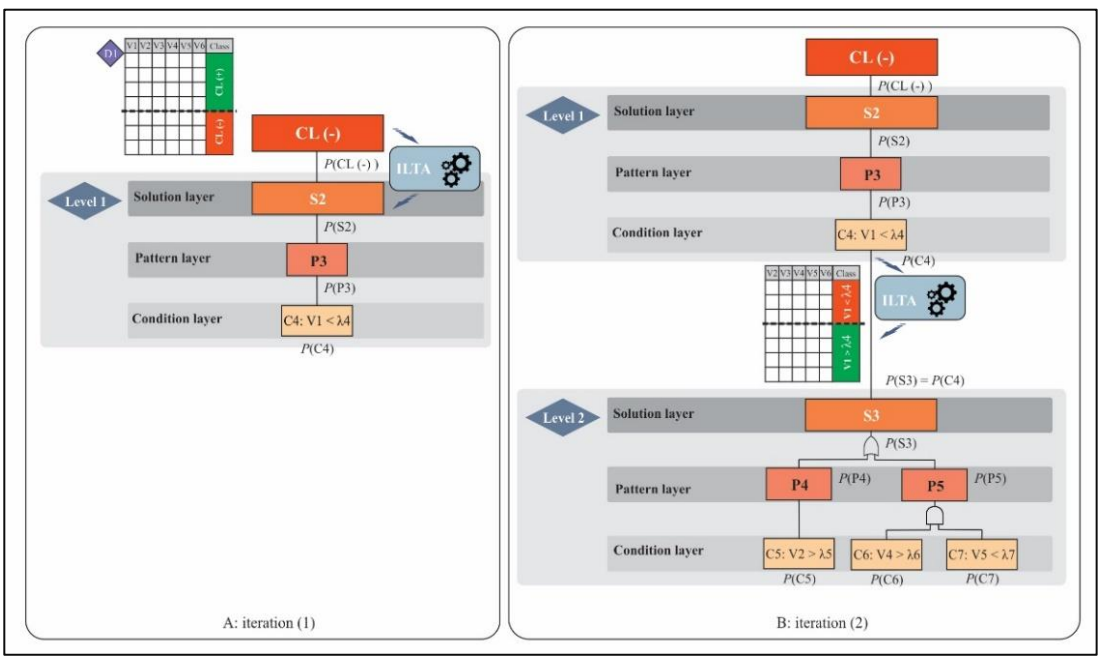


Figure 5.4 Form of the final MILTA model

5.4 Case study

5.4.1 Dataset description

This case study deals with a common complex fault of actuators used to control valves, servomotors and positioners in industrial processes. These actuators work in harsh environments and their defects may have serious effects on the system performance. The dataset is extracted from the actuator 3, which is employed to control the water inflow into the fourth boiler drum in a sugar production process (Wasiewicz, 2001). It consists of 10 numeric variables, as depicted in Table 5.1 (Syfert, 2002).

Table 5.1 Description of the actuator 3 variables

P1	Water pressure (valve inlet)	**X**	Servomotor rod displacement
P2	Water pressure (valve outlet)	**PV**	Process value (water level in steam boiler)
T	Water temperature (valve outlet)	**SF**	Steam flow (steam boiler outlet)
F	Water flow (steam boiler inlet)	**SP**	Steam pressure (steam boiler outlet)
CV	Control value (controller output)	**ST**	Steam temperature (steam boiler outlet)

The dataset is labeled in normal and fault F16 (the *positioner supply pressure drops on Actuator)* classes. It counts 582 observations, which are randomly split into the training and testing datasets. The training dataset represents 65% of the observations for building the MILTA model, while the testing dataset is used to validate the final model. Figure 5.5 visualizes the observations of the training dataset using a parallel coordinates plot (PCP) (Inselberg, 2009) and random forest classification (Breiman, 2001) to rank the variables according to their importance in separating the normal and the fault F16 classes. Accordingly, the variables ST and SF have the full ability to distinguish the two classes compared with the other variables. However, as described in (Bartyś & Syfert, 2001), the drop in the positioner supply pressure, which controls the flow through the valve, is the main indicator of F16. The servomotor rod displacement (variable X) represents the root-cause variable that is able to quantify the pressure drop by measuring the related positioner rod displacement. Therefore, the challenge is to build a MILTA model for F16 and search for a causality relationship between the variables that explain its occurrence.

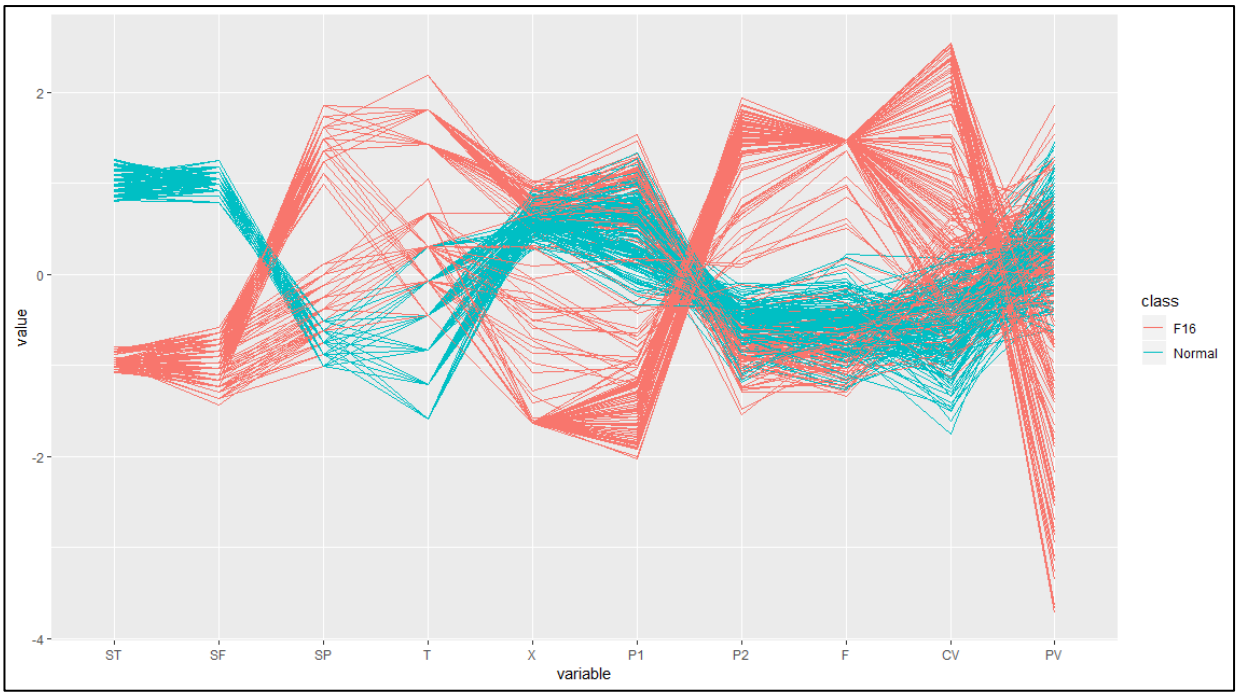


Figure 5.5 PCP multivariate visualization for the dataset variables

5.4.2 MILTA model construction

In what follows, we apply the three-phase MILTA methodology to the actuator training dataset. Phases 1 and 2 are repeated at each level of the MILTA model. The WEKA-LAD software (Gomes & Bonates, 2014a) is employed to extract the patterns that separate the normal class from the F16 class. The minimum class pattern coverage is set to 30%, meaning that only the patterns with a coverage greater or equal to that threshold will be included in the pattern set. Note that it is possible to employ other pattern extraction techniques, such as rough set theory patterns (Thangavel & Pethalakshmi, 2009), patterns extracted from decision tree branches (Berger, Merkl, & Dittenbach, 2006), assembled patterns from the random forest model (Fokkema & Strobl, 2019) or any other technique that adheres to the pattern condition format (variable, inequality sign and cut point).

Figure 5.6 presents the final MILTA model that explains the occurrence of the normal and F16 classes. Four decomposition levels depict the casualty structure between the variables of the actuator. Note that at each level, the minimum solution overlap and coverage percentages are set to 0% and 95%, respectively. At Stage 1 of Phase 1, the overlap threshold controls the knowledge redundancy between the patterns of a given solution, while the coverage threshold is employed as

a decision criterion, at Stage 1 of Phase 2, to avoid further decomposition of a weak, explainable solution.

At the 1st level, two ILTA models are built for the normal and F16 classes. The obtained feasible solutions S_1 and S_2 (respectively, S_3 and S_4) involved in the normal (respectively, the fault) class have 100% coverage and 0% overlap. For both classes, level 1 of the MILTA model includes, at the condition layer, the same variables ST and SF found with the PCP results (Figure 5.5). At the 2nd level, a new two-class dataset is generated for each condition C_1 and C_2 of the normal class, and C_3 and C_4 of F16 class. The solution S_5 that explains both conditions C_1 and C_2 has a coverage of 99% and 0% overlap. S_6 characterizes the conditions C_3 and C_4 has a 96% coverage and 0% overlap. At level 3, five sub-datasets are generated for C_5 to C_9. Then the ILTA model is constructed with each sub-dataset. As no feasible solution that respects the coverage and overlap thresholds explain C_5, C_8 and C_9, these conditions represent the root-causes of their respective class. While for C_6 and C_7, the feasible solutions S_7 and S_8 have a coverage of 99% and 0% overlap. Thus, their respective ILTA models are added to the tree. The same procedure continues at the 4th level. Two sub-datasets are generated for C_{10} and C_{11} leading to two feasible solutions, S_9 and S_{10} with a coverage of 98% and 0% overlap. Then, the corresponding ILTA models are added to the tree. However, no feasible solution satisfies the coverage and overlap thresholds to explain C_{12} and C_{13}. Then, no more levels will be added to the tree. Therefore, C_{12} and C_{13} represent the root-causes of the normal and F16 classes. In addition, C_{12} and C_{13} share the same variable X but with disjunctive intervals. The final MILTA model has 4 levels and finds 5 different root-causes.

Moreover, as described in (Bartyś & Syfert, 2001), the fault F16 occurs when the positioner air supply pressure decreases, which leads to a drop in pressure. The variable X (servomotor rod displacement) is known as the root-cause variable of F16 without any information about its boundary or other hidden root-causes. However, the obtained MILTA model discovers not only the same root-cause and its boundary value ($X \leq 50.7$) at the 4th level, but also two other root-causes at the 2nd level $T > 100.05$ and $SP > 2411.2$ are found.

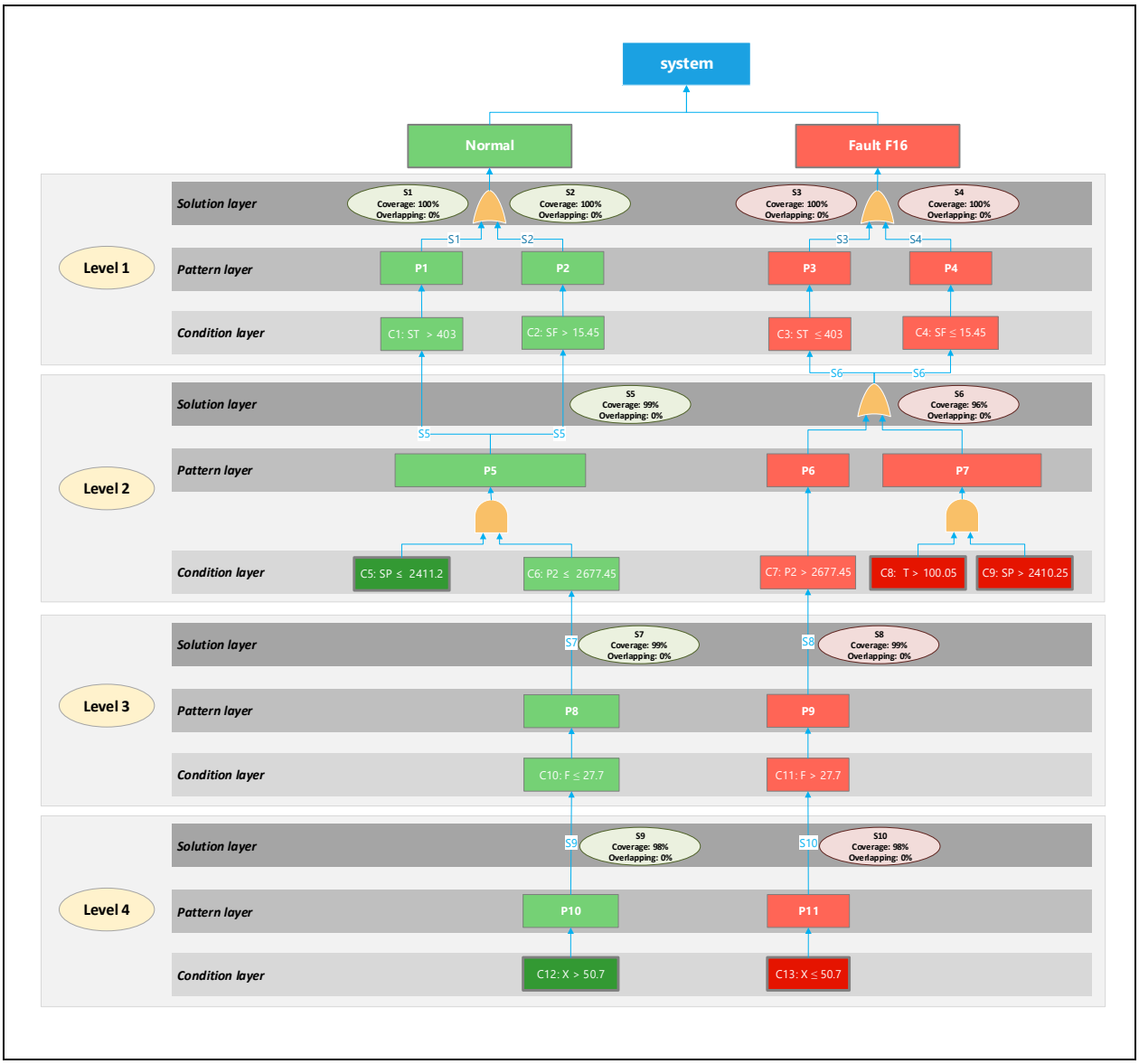


Figure 5.6 The actuator MILTA model

Table 5.2 summarizes the probability calculations associated with the MILTA model. At each level, the probabilities of each condition, pattern, solution and class are estimated using the equations 1 to 4, formulated at Stage 4 of Phase 1, respectively.

Table 5.2 Probability results for each layer in MILTA's four levels

Class#			*Normal*		*Fault (F16)*		
$P(CL_{\#})$			*0.479*		*0.520*		
Leve1 1	*Solution layer*	*S#*	$S_1 : P_1$	$S_2 : P_2$	$S_3 : P_3$	$S_4 : P_4$	
		$\mathcal{P}(S_{\#})$	*0.479*	*0.479*	*0.520*	*0.520*	
	Pattern layer	*P#*	$P_1 : C_1$	$P_2 : C_2$	$P_3 : C_3$	$P_4 : C_4$	
		$\mathcal{P}(P_{\#})$	*0.479*	*0.479*	*0.520*	*0.520*	
	Condition layer	*C#*	$ST > 403$	$SF > 15.4$	$ST \leq 403$	$SF \leq 15.45$	
		$\mathcal{P}(C_{\#})$	*0.479*	*0.479*	*0.520*	*0.520*	
Leve1 2	*Solution layer*	*S#*	$S_5 : P_5$		$S_6 : P_6 \cup P_7$		
		$\mathcal{P}(S_{\#})$	*0.474*		*0.313*	*0.510*	
	Pattern layer	*P#*	$P_5 : C_5 \cap C_6$		$P_6 : C_7$	$P_7 : C_8 \cap C_9$	
		$\mathcal{P}(P_{\#})$	*0.474*		*0.313*	*0.339*	
	Condition layer	*C#*	$SP \leq 2411.2$	$P2 \leq 2677.4$	$P2 > 2677.4$	$T > 100$	$SP > 2410.2$
		$\mathcal{P}(C_{\#})$	*0.557*	*0.686*	*0.313*	*0.353*	*0.482*
Leve1 3	*Solution layer*	*S#*		$S_7 : P_8$	$S_8 : P_9$		
		$\mathcal{P}(S_{\#})$		*0.672*	*0.309*		
	Pattern layer	*P#*		$P_8 : C_{10}$	$P_9 : C_{11}$		
		$\mathcal{P}(P_{\#})$		*0.672*	*0.309*		
	Condition layer	*C#*		$F \leq 27.7$	$F > 27.7$		
		$\mathcal{P}(C_{\#})$		*0.672*	*0.309*		
Leve1 4	*Solution layer*	*S#*		$S_9 : P_{10}$	$S_{10} : P_{11}$		
		$\mathcal{P}(S_{\#})$		*0.658*	*0.300*		
	Pattern layer	*P#*		$P_{10} : C_{12}$	$P_{11} : C_{13}$		
		$\mathcal{P}(P_{\#})$		*0.658*	*0.300*		
	Condition layer	*C#*		$X > 50.7$	$X \leq 50.7$		
		$\mathcal{P}(C_{\#})$		*0.658*	*0.300*		

The causality rules are derived from the main logic tree in the form of mathematic expressions. Each causality rule expresses the cause-effect relations between the root-causes, the intermediate causes and the class event. It helps the expert diagnose each state by grasping the contribution of the discovered root-causes in an interpretable way. Figure 5.7 presents the four causality rules. *R1* and *R2* express the causality rules of the normal class, while *R3* and *R4* concern the fault F16.

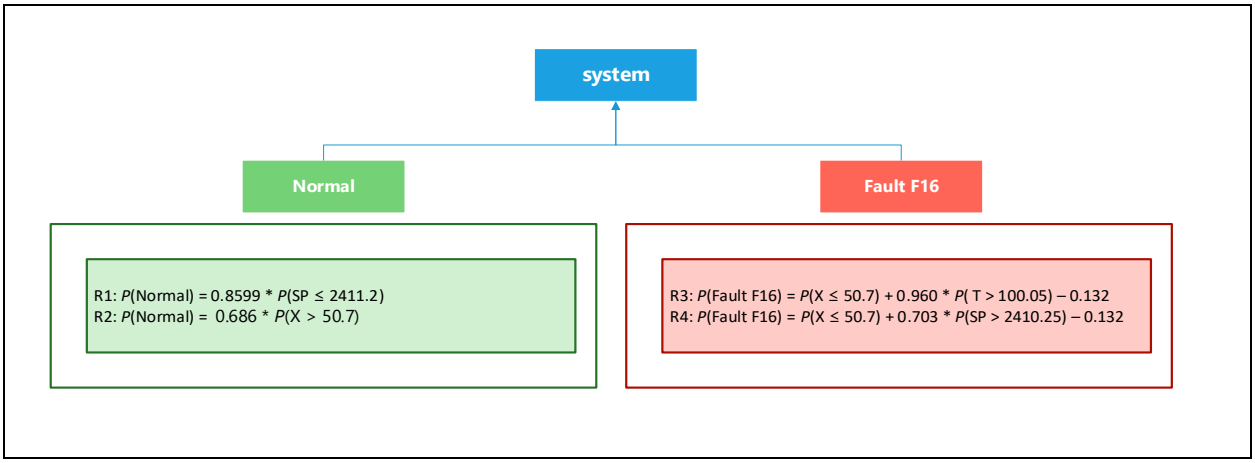


Figure 5.7 The derived causality rules

5.4.3 Validation of the MILTA model

The accuracy of the obtained MILTA model is estimated using a mean error between the predicted probability of a given class and the actual probability. 1000 random samples of 146 observations that give a 95% confidence interval are generated from the test dataset. The mean error distribution of each causality rule is plotted on Figure 5.8 where M_{error} and S_{error} denote the mean and the standard errors, respectively.

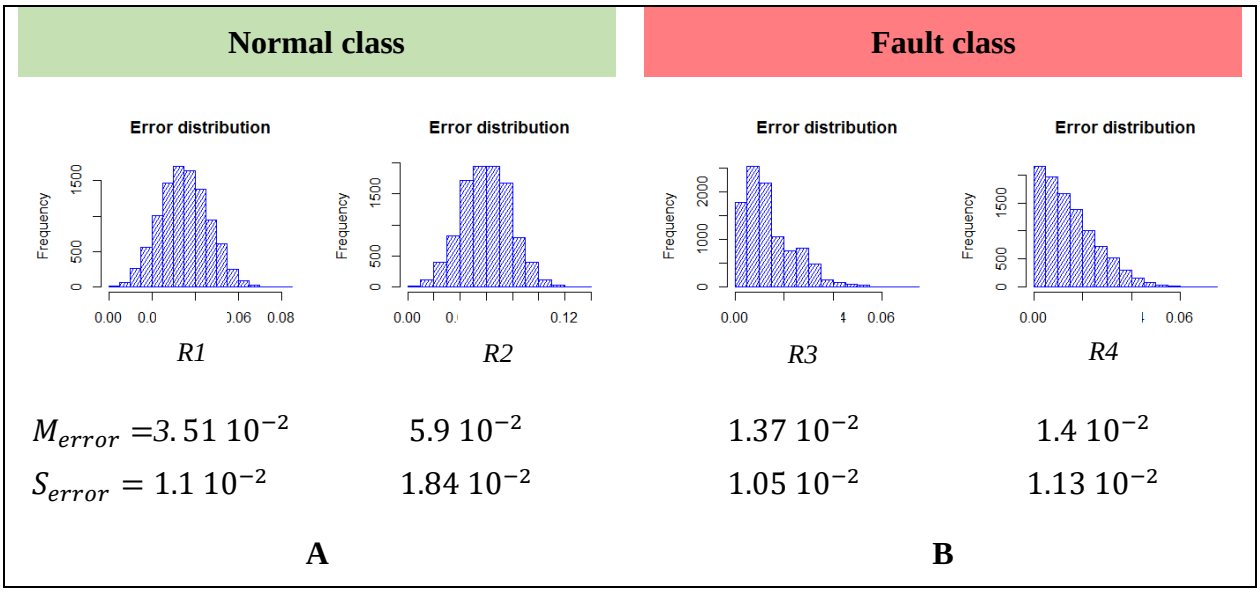


Figure 5.8 Error distribution of each causality rule

According to Figure 5.8-A, the mean errors of *R1* and *R2* follow a normal distribution, whereas the mean errors of *R3* and *R4* have an exponential distribution (Figure 5.8-B). Furthermore, the M_{error} of the fault F16 is lower than the M_{error} of the normal class. In addition, the causality structure between the root-causes, the intermediate causes and the fault F16 are clearly depicted in Figure 5.9. At the 1st level, the coverage of S_3 and S_4 is shown through coverage of their respective conditions C_3 (A) and C_4 (B) which distinguish the fault F16 from the normal class. At the 2nd level, the feasible solution S_6 that explains both C_3 and C_4 has the depicted coverage (C) of the combination of P_6 and P_7 as depicted in D and E. Meanwhile, P_6 has the same coverage (G) of its condition C_7. However, the coverage of P_7 conjuncts the coverage of its two conditions C_8 and C_9 as shown in E and F. At the two final levels (3 and 4), the coverage of the solutions S_8 and S_{10} is depicted through the coverage of C_{11}(H) and C_{13} (I), respectively. Thus, the MILTA model tracks the causality of F16 and isolates the intermediate causes (plots A, B, F and I) iteratively until reaching its the root-causes (plots G, H and J)

Since there is no prior information about the two other root-causes at the 2nd level ($T > 100.05$ and $SP > 2411.2$) in the data documentation, the contribution of T and SP on the occurrence of the fault F16 is verified by calculating their correlation with the main root-cause X individually (Table 5.3).

Table 5.3 Pearson correlation between the fault root-causes

Correlation between the fault root-causes		
R3	X – servomotor rod displacement and T - water temperature (valve outlet)	0.00069
R4	X – servomotor rod displacement and SP - Steam pressure (steam boiler outlet)	-0.011

The correlation values of T with X and SP with X are too low. Therefore, T and SP are more independent than X. The MILTA model does not further explore these two variables, yet they correspond to fault indicators rather than to root-causes. Consequently, some missing variables need to be measured and added into the dataset to depict the role of T and SP in the fault F16 occurrence.

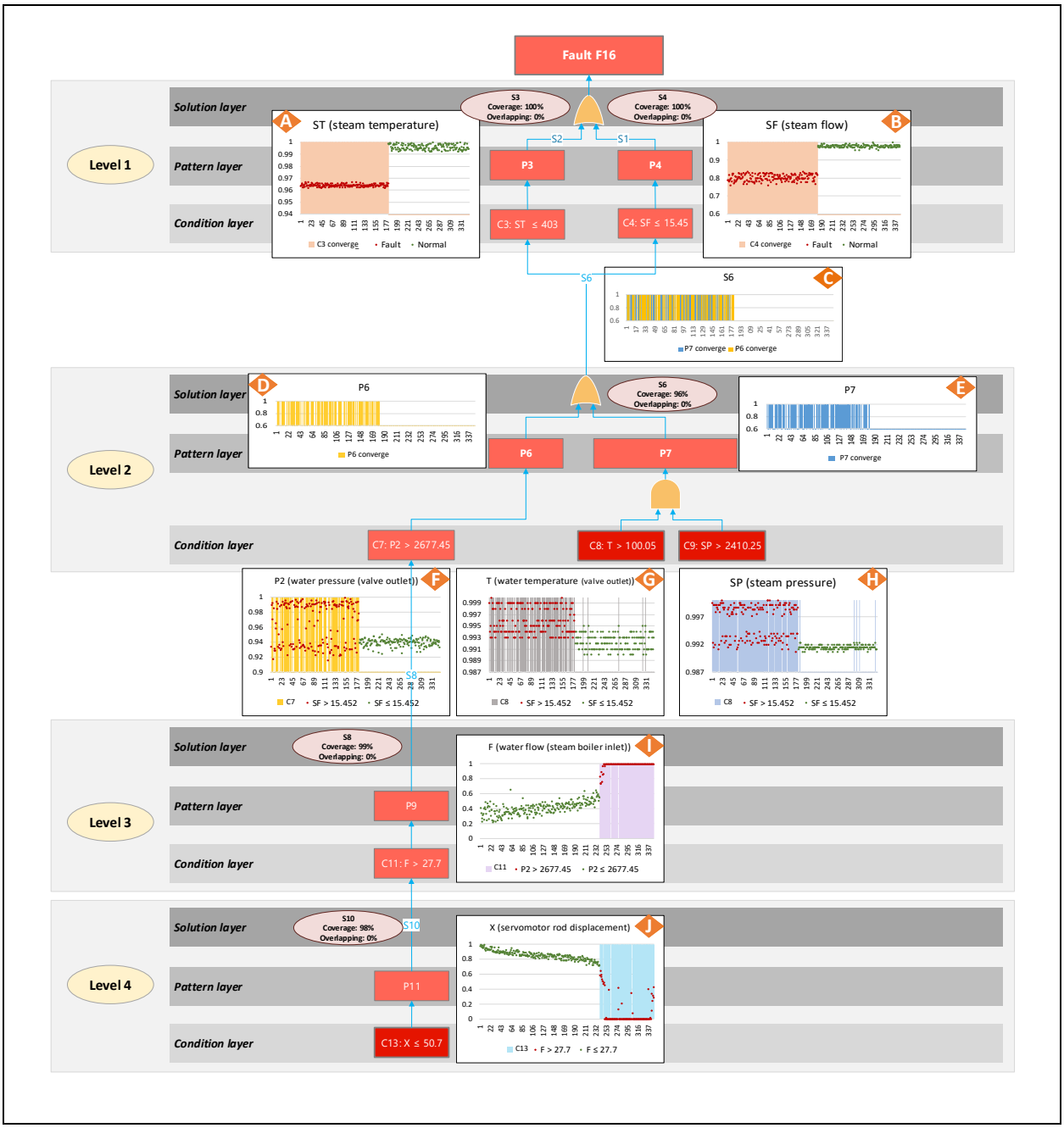


Figure 5.9 MILTA subtree for the fault F16

5.4.4 Root-cause analysis

The causality rules of F16 can be also employed to rank the importance of each root-cause to explain the occurrence of fault F16. The ranking criterion averages three normalized scores:

- The mean coverage of the solutions' path from the fault event to the root-cause. A high score indicates a strong contribution of the root-cause on the fault event occurrence.

- The location (i.e. depth) of the root-cause in the tree. The closer the depth score is to 1, the more the root-cause has a specific power to discriminate the fault event. The score value is normalised against the total number of obtained levels.
- The importance of the root-cause conditional probability in the causality rule. A high value means a strong effect of that probability on the fault event occurrence.

Table 5.4 details the calculation of the ranking values of the five root-causes founded through the four levels. For instance, the root-cause $X \leq 50.7$ has a mean coverage of 0.89, a total depth score of 1 (0.25 for each level depth) and an importance conditional probability of 1 based on the causality rules *R3* and *R4* that characterize the fault F16 event. Thus, the ranking value of $X \leq 50.7$ is equal to the average scores between 0.89, 1 and 1. Based on the raking values, it is clear that the root-cause X has an important effect on the occurrence of the normal class and the fault event. Indeed, the ranking value of $X > 50.7$ has a higher value compared to that of $SP \leq 2411.2$ in the normal class. Also, the ranking value of $X \leq 50.7$ exceeds those of $T > 100.05$ and $SP > 2411.2$ in the fault class.

Table 5.4 MILTA root-causes analysis

	Normal class						Fault class (F16)								
Cond.	(X > 50.7)			(Sp ≤ 2411.2)			(X ≤ 50.7)			(T > 100.05)			(Sp > 2411.2)		
Score	Coverage	Depth	Cond. Prob.	Coverage	Depth	Cond. Prob.	Coverage	Depth	Cond. Prob.	Coverage	Depth	Cond. Prob.	Coverage	Depth	Cond. Prob.
Level 1	1	0.25		0.99	0.25		1	0.25		1	0.25		1	0.25	
Level 2	0.99	0.25		1	0.25		0.6	0.25		0.36	0.25		0.36	0.25	
Level 3	0.99	0.25					0.99	0.25							
Level 4	0.98	0.25					0.98	0.25							
Score value	**0.99**	**1**	**0.69**	**1**	**0.5**	**0.86**	**0.89**	**1**	**1**	**0.68**	**0.5**	**0.96**	**0.68**	**0.5**	**0.7**
Ranking value	**0.89**			**0.79**			**0.96**			**0.71**			**0.63**		

5.5 Conclusion

This paper has proposed a multi-level interpretable logic tree for the hierarchical causality analysis of faults in a complex system. It is a data-driven model that combines the knowledge discovery in a dataset and fault tree analysis. The model uses an iterative burn-and-build algorithm to select the feasible solutions that reflect the causality structure between the fault event and its intermediate causes, level after level, until the root-causes are uncovered. Unlike the event-based causality approaches, the three-phase MILTA methodology does not require any human expert involvement to construct the model. The expert has to fix the decision criteria regarding the coverage and overlap

percentages that are essential to build a knowledge-rich model, and then to provide an appropriate level of interpretability to the MILTA model. Supported by the results of the MILTA model, a human expert continues to play the role of decision-maker with regards to the validation of the root-causes revealed by the model, to remove certain useless branches of the tree or to enrich others.

However, the main limitation of the MILTA model is related to the fact that the causality rules are static in time. They represent the fault event occurrence at a certain time. When the system undergoes a degradation in time, the MILTA model needs to be improved to catch the fault event evolution through that time. Thus, there is a need for a new tree that considers the system's aging effect on the change in the fault causality structure, and is depicted in an interpretable manner, similar to a MILTA model, to perform the fault prognosis task in a dynamic system well.

CHAPTER 6 ARTICLE 3: A DATA-DRIVEN FAULT TREE FOR A TIME CAUSALITY ANALYSIS IN AGING SYSTEMS

Kerelous Waghen & Mohamed-Salah Ouali*

Mathematics and Industrial Engineering Department, Polytechnique Montréal

2500 Chemin de Polytechnique, Montréal, QC, H3T 1J4, Canada

kerelous.waghen@polymtl.ca; mohamed-salah.ouali@polymtl.ca

*Member of CIRRELT (https://www.cirrelt.ca) and IVADO (https://ivado.ca)

Submitted at Reliability Engineering and System Safety

(Manuscript Number : RESS_2020_180)

Abstract:

This paper develops a data-driven fault tree methodology that addresses the problem of fault prognosis of aging systems based on graphical time causality model. It merges the concepts of knowledge discovery in dataset (KDD) and fault tree (FT) to develop an interpretable time causality analysis model (ITCA). The ITCA model overcomes the limitation of the current graphical models by predicting the occurrence of the fault event based on the changing of its causality structure over time. At periodic times, the ITCA model captures the cause-effect relations in the form of interpretable logic trees, then summarizes in the ITCA models to reflect the changing in the causality structure over the time. The obtained root-causes from ITCA model are actualized and depicted the fault development over the time causality structure in bottom up manner. The well-known NASA turbofan engine dataset is used as an illustrative example of the proposed methodology.

Keywords: Knowledge Discovery in Dataset; Fault Tree; Causality Analysis; Aging Systems.

6.1 Introduction

The system aging is a progressive deterioration in its initial performance over time, mainly caused by the occurrence of faults that adversely affect the system's reliability. Causality analysis methods aim to diagnose the fault through identifying, isolating and quantifying the effect of the root-causes on the system performance so that the appropriate maintenance actions can be performed to restore the system to good conditions (Ming, Heng-Chao, Bin, Jun-Hong, & Chee Khiang, 2015). Prior knowledge about future fault behaviour and its drawback on the system's performance are essential in order to optimize the maintenance decision-making (de Jonge, Bram, 2019). The fault prognosis task provides this prior knowledge that depicts the progression of a specific failure mode from its inception until the time of failure (Wang, K. S., 2014). The time causality analysis is an extension from the causality analysis methods, where able to build a prognostic model that demonstrate this prior knowledge through capturing the fault causality structure evolution over the time (Bousdekis, Magoutas, Apostolou, & Mentzas, 2018).

Those prognostics models able to represent the fault development over the time by a mathematical expression that quantified the general fault evolution or a graphical model that graphically demonstrate the changing in the causality structure over the time (Schwabacher, 2005). The expert prior knowledge regarding the future fault development can be modeled to demonstrate the fault causality structure progress. However, this knowledge could be biased and reflect only the expert understanding regarding the fault development (Aggab, Kratz, Avila, & Vrignat, 2018). On the other hand, the fault evolution knowledge can be extract directly from the data, which is unbiased knowledge and reflect the fault causality. However, it is missing the interpretability and the expert representation for the fault hierarchical causality knowledge over the time (Schwabacher & Goebel, 2007).

Waghen, and Ouali (2019) develop a data-driven fault tree method for causality analysis that address the lack in the data-driven model interpretability and overcome the model-based limitation that influenced by the expert prior knowledge. The method visualizes the fault causality architecture in one level that composed three layers. The condition layer identifies the fault root-causes and their coverage ranges within the dataset. The pattern layer arranges the root-causes in the form of interpretable conjunctions. The solution layer combines some selected patterns that depict the fault event. Although the proposed tree is interpretable for the expert, the model hides

the fault hierarchical cause and effect relationships in a complex system that should represented in multilevel tree. Besides, it reflects the fault causality in a static way without taking into account the influence of a system's aging on the change in the fault causality structure over time.

In this paper, an interpretable time causality analysis (ITCA) methodology is developed to address the problem of fault prognosis in aging systems using a data-driven fault tree model. It aims to build a time-dependant multilevel causality model based on the selection of the feasible solutions that characterize the fault occurrence at a certain period of time, from a set of representative time series historical datasets, to address the causality analysis over time in a meaningful way. The ITCA time-dependant multilevel structure based on combination of different one level common fault trees that connected together for depicting the fault time hierarchical structure. At each defined period, an ILTA model is build for identifying and isolating the possible causes of the fault event at that period, compared to the fault state observations. Those constructed ILTA for each period are summarized in a one level common fault tree that graphically summarized the changing in the causality structure over the time and initiate a level in ITCA model. This process is iteratively repeated until the final ITCA time-dependant multilevel structure is constructed. While this process is monitored in order to ensure that redundant knowledge is eliminated within the ITCA model, while maximizing its interpretability over the time. Finally, a set of causality rules are deduced from the ITCA fault tree that characterize the dynamic change effect of the causality structure in the causes of a fault occurrence.

The rest of the paper is organized into four sections. Section 2 reviews the available methods for achieving fault prognosis based on time causality analysis and discusses the main challenges. Section 3 develops the ITCA methodology. It explains the data preparation, the construction of the fault tree models over time and the deduction of the time causality rules for fault prognosis. Section 4 illustrates the ITCA methodology using the NASA turbofan engine degradation dataset. The performance of the ITCA model to predict the fault is demonstrated through the fault trend over time. Section 5 concludes the paper and discusses the contribution of the ITCA methodology in achieving the fault prognosis task.

6.2 Time causality analysis approaches

Time causality analysis is a causal interference over the time, where the temporary dependency over the stochastic process is captured and modelled. It is an analytical tool that provide the expert with the essential knowledge regarding the fault evolution and the changing in the fault causality structure over the time for better achieving the fault prognosis task (Chen, H.-S. et al., 2018). Two categories of time causality analysis approaches can be distinguished in the literature: model-based and data-driven approaches (Schwabacher, 2005). In what follows, these approaches are discussed, and their strengths and limitations for prognosing the fault are summarized help clarify the research gap.

The model-based time causality approach relies heavily on human expertise to describe the system's behaviour over time in degraded conditions (Vania, Pennacchi, & Chatterton, 2013). It able to capture the fault physics developments, using a mathematical expression or graphical model, where they able to underline the fault time causality structure due to the fault development and system degradation (Celaya et al., 2011). Lu et al. (2012) address the drawback of the system downtime due to fault evolution in complex industrial process by enriching the expert knowledge for driving fault prognosis strategy. The proposed methodology estimates the time delay in the process industry. First, time-delayed mutual information (TDMI) is employed to model the fault causality in the form of a time-delayed signed digraph (TD-SDG) mode. Then, a general fault prognosis strategy is used to optimize the system's downtime based on TD-SDG and PCA. Darwish, Almouahed, and de Lamotte (2017) proposed enriched fault tree analysis (FTA) approach for Active Assisted Living Systems. The failure of FTA basic events are ranked based on expert prior knowledge for defining the degree of importance for those events. Those failure distributions are calculated based on fuzzy and possibility theories concepts for obtaining imprecise failure probabilities. While, the expert prior knowledge is employed to rank the importance of the obtained basic event in order to construct the fault failure distribution that reflect efficiently the contribution of those basic events. Ragab, A. et al. (2019) combine the domain knowledge with the extract knowledge from data base for constructing enriched fault tree analysis (FTA). The expert first constructs the FTA model skeleton, which represent the main causality structure for the fault. Then, patterns are extracted from data in order to discover the hidden phenomena about the fault for enriching the FTA model. Yunkai, Bin, Ningyun, and Yang (2015) integrate the Bond graph

modelling technique with the Bayesian network for a fault prognosis of a high-speed train traction system. The Bond graph represents the system structure that mainly constructed based on expert prior knowledge. While, the Bayesian network enriches the expert prior knowledge represented by the bound graph through discovering the hidden causal relationships.

Indeed, the model-based time causality approach able to provide interpretable and relatively accurate models that could build from the first principle of the system's faults. It is mainly applicable on a simple system with well-known causes, for which the human knowledge about the faults, their occurrence and development are clear. Its limited implementation in complex system was overcome by enriching those models based on data-driven techniques, in which the unseen events are discovered and added to the model's prior knowledge. However, forming the model skeleton prior knowledge by the expert in complex system is a challenging task to identify the principal causality structure of the faulty situation in addition combining and positioning the extracted hidden fault knowledge from the data in the constructed model.

Unlike the model-based approach, The data-driven time causality explores the data using machine learning (ML) techniques and does not impose a model to predict the behaviour of a complex system (Jin, S., Zhang, Chakrabarty, & Gu, 2018). The ML data-driven methods build unbaised models and are able to deal with noisy and correlated variables (Niu, 2016). Khumprom, and Yodo (2019) employ deep neural networks (DNN) to estimate the State of Health (SoH) and predict the RUL of lithium-ion batteries based on the NASA dataset benchmark (Saha & Goebel, 2007). Shili, Dong, and Xiaoli (2018) develop a big data-driven approach that increases the efficiency of a prognosis and reduces maintenance costs by determining the relevant features that explain the model. Razavi, Najafabadi, and Mahmoodian (2018) develop an adaptive neuro-fuzzy inference system (ANFIS) algorithm that integrates the artificial neural network (ANN) with fuzzy rule-based systems to predict the RUL in order to optimize the maintenance schedule for aircraft engines.

Although the data-driven models offer an accurate prediction of the RUL that able to generally quantify the fault evolution, they suffer from a lack of interpretability (Doukovska & Vassileva, 2013). This is because they are too shallow to understand the fault causality structure and its changes over time. Therefore, an expert may not be able to deeply understand the cause-effect relations within a complex system. With regards to this challenge, several methods have been

proposed to simplify and unlock the model interpretability. Su et al. (2015) propose a dynamic extraction knowledge method that illustrates the relationship between the environmental stresses and the system failure modes using a fuzzy causality diagram and a Bayesian rough set of multiple decision classes to weigh the extracted knowledge. Kimotho, Sondermann-Woelke, Meyer, and Sextro (2013) Address the challenge of recommending maintenance actions for industrial systems based on remote monitoring and diagnosis. Event-based decision trees were built for graphically identifying problems associated with particular events and conducting to evidence decision due the tree interpretability. Medjaher, Moya, and Zerhouni (2009) implement the Dynamic Bayesian Networks (DBNs) for quantifying the failure prognostic in complex systems. The fault time series data is divided into time slices and a Bayesian network is constructed for each time slice. Then the constructed networks over the time slices are connected through the temporal dependency, which depicts the changing in the fault causality structure over the time and quantify the fault developments.

On the other hand, the achieved data-driven methods attempt to unlock the time-dependent relations between the system variables in an interpretable manner besides, capture the changing in the fault causality over the data time slices. However, building an interpretable data-driven model that is able to directly grasp the influence of the system aging on the fault causality structure and summarize the causality changing in one model still needs to be overcome.

The main motivation is to build an interpretable time causality analysis model that characterizes, first, the hierarchical causality structure between the fault event, intermediate causes and root-causes; and second, the influence of the system aging on that structure over the time. Thus, the proposed ITCA methodology will achieve the fault prognosis task in an efficient way through anticipating the fault event based on the causal relationships discovered over the time. It will be developed in the following section.

6.3 The ITCA Methodology

Figure 6.1 depicts the four-phase ITCA methodology. The main input dataset is an unlabeled timestamp of observations that can represent sequential data. We assume that the system undergoes a certain degradation trend, depicted by the sequential data, from a normal state to a failure state. Phase 1 prepares several labeled subsets from the input data. Each subset is formed by a sub-

sequence of degraded observations and failures. Phase 2 iteratively builds the appropriate logic tree corresponding to each subset of data, and then aggregates them into one common fault tree. Phase 3 constructs the ITCA model by going deeply through each variable in the above common fault tree and seeks its root-causes. Phase 4 deduces the time causality rules that the effects that the system aging has on the evolution of the fault occurrence over time. In what follows, each phase of the proposed methodology is explained in detail.

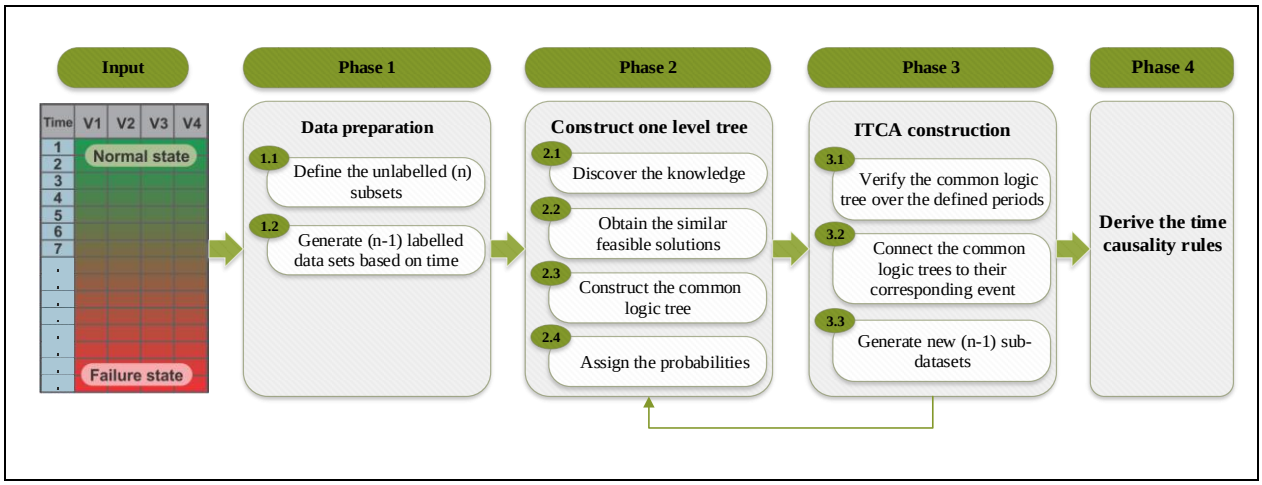


Figure 6.1 The four-phase ITCA methodology

6.3.1 Phase 1-Data preparation

Phase 1 splits the main input data into several subsets according to the expert's prior knowledge about the process degradation trend. Each subset contains the sequential observations that represent the system state at a certain period of time and the observations that characterise the failure state or the worst deterioration condition of the system. The expert should identify the observations that represent the failure before splitting the rest of the data into equal or non equal sizes of subsets, according to his judgment about the amount of system degradation. Equal and non-equal sizes of subsets are suitable for linear and nonlinear degradation processes, respectively. Hence, the original main data is divided into n subsets, where the last one contains the failure observations and the others contain degraded observations. Those n subsets will be concatenated to form (n-1) datasets. Each dataset will contain two classes of observations corresponding to failure and degraded data.

Figure 6.2 depicts the data preparation procedures, in which X_1 and X_2 are two variables. Beginning from the main timestamped dataset, the observations in the last period of time Δ_n belong to the failure state. Then, (n-1) subsets are extracted. Each subset SS_i contains the observations of the

period Δ_i, $i=1,..,(n-1)$. At the end, $(n-1)$ labeled datasets are concatenated. Each dataset D_i contains the observations of the period Δ_i, labeled as class i and the observations of the last period Δ_n, labeled as class n.

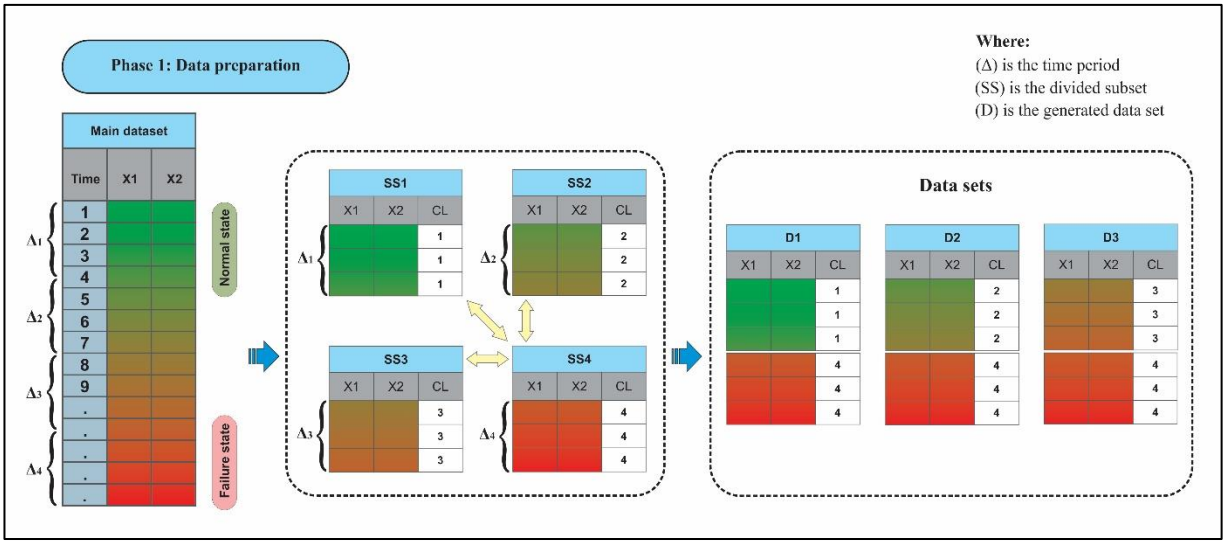


Figure 6.2 Data preparation phase

6.3.2 Phase 2-Build a one level fault tree

Phase 2 iteratively extracts all of the logic trees that differentiate the fault event (class n) from each class i, $i = 1,\ldots,(n-1)$ of the degraded observations individually. Each logic tree highlights the relevant variables that discriminate the observations of the failure state from the degraded ones, from one period to another. Then, the obtained logic trees are merged into one common fault tree, which identifies and isolates the variables that discriminate the failure state from the degraded ones over time. To do so, Waghen, and Ouali (2019) developed a four-stage methodology, named Interpretable Logic Tree Analysis (ILTA), to build a one level fault tree from a two-class dataset (i.e. normal and failure classes). The methodology discovers the knowledge from the dataset (Stage 1); forms feasible solutions (Stage 2); constructs the fault tree (Stage 3); and finally quantifies the fault tree using Bayes' theorem (Stage 4). Although such a methodology can be applied separately with each dataset D_i, $i=1,..,(n-1)$, the merged fault tree may be difficult to interpret due to the dependence of the datasets over time. To overcome this limitation, stage 2 of the ILTA methodology needs to be improved. Nevertheless, for the convenience of the reader, we briefly

recall the four stages of the ILTA methodology and highlight the improvements to stage 2 in the following.

- ***Stage 1-Discover knowledge.*** Discovering knowledge from a two-class dataset can be achieved through different pattern generation and extraction techniques, such as logic analysis of data (LAD) (Hammer, Peter L & Bonates, 2006) and Prediction Rule Ensembles (PRE) (Fokkema, 2017). The pattern is a conjunction of certain conditions that discriminate one class of observations from another class. Each condition includes a variable, an inequality sign and a cut point value. Furthermore, the percentage of observations covered by a given pattern may characterize the knowledge expanse caught by that pattern. However, when the observations of the same class are covered by more than one pattern, an overlap between those patterns may occur, with a certain percentage leading to redundant knowledge.

  - ***Stage 2-Obtain similar feasible solutions.*** A solution is defined as a combination of certain patterns that cover the observations of the same class. Each solution can be characterized by its coverage (Cov) and overlap (OL) percentages. The feasible solution is a solution that respects certain criteria. In the ILTA methodology, only the feasible solution that maximizes the class Cov and minimizes the class OL is selected, which leads to maximizing the interpretability and minimizing the redundancy of the discovered knowledge. However, in the ITCA methodology, we need to search for all of the feasible solutions that respect not only the Cov and OL threshold percentages, but also with minimal number of patterns to capture the fault hierarchical causality. Maximizing the coverage and minimizing the overlap with minimal number of patterns allows to represent the fault most generalized knowledge in the first levels. While, as the ITCA levels are added as the knowledge become more specific to depict certain causality in the tree. Therefore, by forming the feasible solution with minimal number of patterns, ITCA model able to represent the fault hierarchical knowledge through the included ILTA over its levels. As Stage 2 aims to select similar feasible solutions that characterize the knowledge discovery over time, we seek the most frequent patterns over the predefined periods of time. In addition, the frequent pattern involves the same variable and inequality sign in the shared conditions, independent of the cut-point values. Therefore, the initial version of the burn-and-build algorithm proposed in (Waghen and Ouali, 2019) is improved to form a set of feasible solutions instead of only one for each period of time using another decision criterion, called the solution tolerance

selection (STS) threshold. Hence, a time-based searching algorithm is developed in the ITCA methodology to obtain all of the similar feasible solutions over time. It is depicted in the following pseudo code.

Time-based searching algorithm:

Search for similar feasible solutions over time

Input.

iii) (n-1) labelled datasets corresponding to the defined time periods (Δ)

iv) Set of generated patterns: P_{gen} = {P_1,P_2,...., P_l}; where (l) is the number of the discovered patterns

v) Overlap (OL) threshold

vi) Solution tolerance selection (STS) threshold

***For** each (n-1) labelled dataset that represents a defined time period (Δ)*

***Step 1.** Select a start pattern (P_i) and calculate its coverage*

1.4. Remove the overlapped patterns with P_i based on preset overlap threshold

i. At number of combination (n) = 2

ii. Select the combination with P_i that has the maximum coverage

1.5. Repeat (the sub-steps 1.1-i and ii) until the number of combinations (n) = number of the discovered patterns

1.6. Compare the selected combinations at each n and select the combination that maximizes the coverage with a minimal number of patterns

***Step 2.** Select another pattern (P_i) as a start point and repeat 1.1, 1.2 and 1.3.*

***Step 3.** Repeat 2 until considering each pattern as a start point.*

***Step 4.** Compare the selected combinations over the start patterns P_i and select the combination that includes the minimal number of patterns and its coverage value within the STS threshold.*

End

***Step 5.** Compare the selected combinations that represent (n-1) labelled datasets and select the combinations that maximize the similarity over the defined periods (Δ), where each period (Δ) is represented by only one combination.*

Output.

Set of similar feasible solutions: Sol = {S_1, S_2,...., S_k }; where (k) is the number of similar feasible solutions.

Figure 6.3 illustrates the proposed time-based searching algorithm using the above three concatenated datasets D_1, D_2 and D_3 of the toy example (Figure 6.1). Applying Step 1 to Step 4, the algorithm finds a set of five feasible solutions that respect the STS threshold of 90%. To clearly understand this, we assume that each solution consists of only one pattern. From D_1, $S_1: P_1: (X_1 \leq 30)$ and $S_2: P_2: (X_2 > 10)$ are obtained with 98% and 100% of Cov, respectively. From D_2, there is only one formed solution $S_3: P_3: (X_1 \leq 20)$ with a Cov of 90%. From D_3, the obtained solutions $S_4: P_4: (X_1 \leq 10)$ and $S_5: P_5: (X_2 > 20)$ have 95% and 100% of Cov, respectively. Note that the patterns P_1, P_3 and P_4 share the same condition on X_1 except the cut-points. Consequently, at Step 5 the algorithm selects S_1, S_3 and S_4 as the only three similar solutions that characterize the evolution of the same condition through the three periods Δ_1, Δ_2, and Δ_3, respectively. However, the algorithm does not select S_2 and S_5. Because there is a loss of information during the period Δ_2, even though they are similar, by sharing the same condition of X_2 during Δ_1 and Δ_3. Hence, the algorithm evaluates all of the similar feasible solutions and selects the ones that dominate the maximum number of periods.

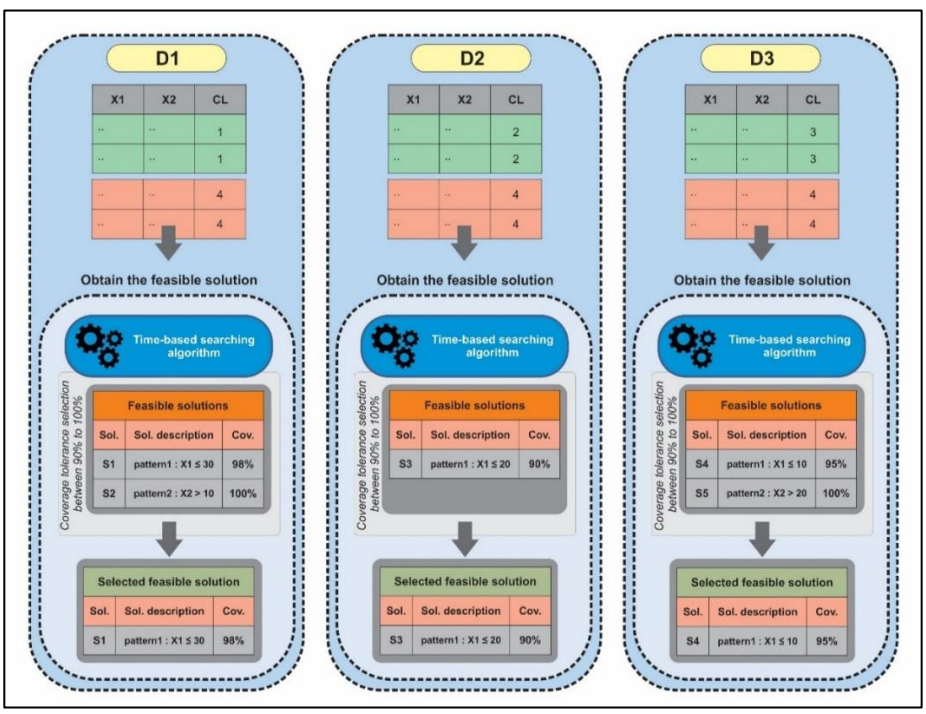


Figure 6.3 Example of selecting similar feasible solutions over the periods Δ_1, Δ_2, and Δ_3

Figure 6.4 depicts the curve of the cut-point values that reflect the evolution of similar feasible solutions obtained over the three periods Δ_1, Δ_2, and Δ_3. Note that these periods are consecutive and the cut-point curve may have a positive, negative or constant trend over time depending on how the cut-point values change over time.

Similar feasible solution

Periods	Δ1	Δ2	Δ3
Sol.	S1: X1 ≤ 30	S3: X1 ≤ 20	S4: X1 ≤ 10

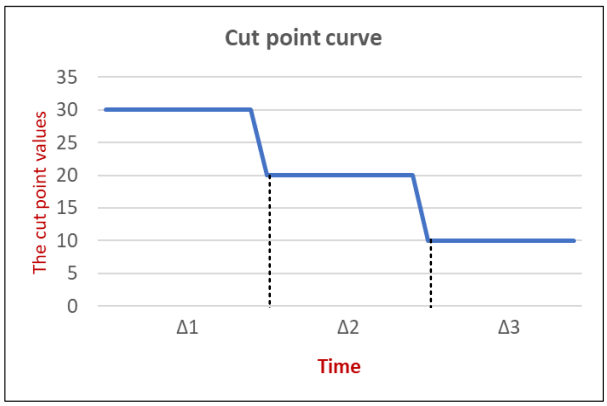


Figure 6.4 Curve of the cut-point values of similar feasible solutions obtained over time

- ***Stage 3-Construct a common logic tree over time.*** The similar feasible solutions obtained are visualized in a one level fault tree through the condition, pattern and solution layers. At the condition layer, all of the involved conditions are connected to their respective patterns using the AND gate. At the pattern layer, all of the patterns of the similar feasible solutions are connected to that solution using the OR gate. Similarly, at the solution layer, all of the selected similar feasible solutions are connected to the fault event using the OR gate.
- ***Stage 4-Assign the probabilities.*** The common logic tree is quantified using the probabilities of the solutions, patterns and conditions involved in similar feasible solutions obtained from the concatenated dataset individually. Let N_k and N_T be the number of observations covered by the condition C_k and the total number of observations in one concatenated dataset, respectively. The equations 1 to 4 calculate the probabilities of the fault class $\mathcal{P}(CL)$ and the involved solutions $\mathcal{P}(S_q)$ $q = 1,2, \dots Q$, patterns $\mathcal{P}(P_j)$ $j = 1..J$ and conditions $\mathcal{P}(C_k)$ $k = 1..K$as follows. Eq.1:

$\mathcal{P}(C_k) = \frac{N_k}{N_T}$. Eq.2: $\mathcal{P}(P_j) = \prod_{k=1}^{n_j-1} \mathcal{P}(C_k|C_{k+1}).\mathcal{P}(C_{k+1})$. Eq.3: $\mathcal{P}(S_q) = \mathcal{P}[\cup_{j=1}^{J} P_j]$. Eq.4: $\mathcal{P}(CL) = \mathcal{P}[\cup_{q=1}^{Q} S_q]$.

For a simple cause-effect relation between the fault event and its root-causes, the common one level logic tree is able to depict the fault causality structure at each period in time, as well as over time through the trend of cut-point curves of similar feasible solutions employed in the tree. For a complex causality structure, the one level logic tree is not sufficient to completely represent a fault occurrence because the variables involved at the condition layer may represent the intermediate causes, and not necessary the root-causes, of the fault event. Therefore, each one of those variables needs a second level of decomposition or more to explore the solution that will explain its causality structure at each period in time. Accordingly, Phase 3 constructs many logic tree levels to address the complex causality structure over time.

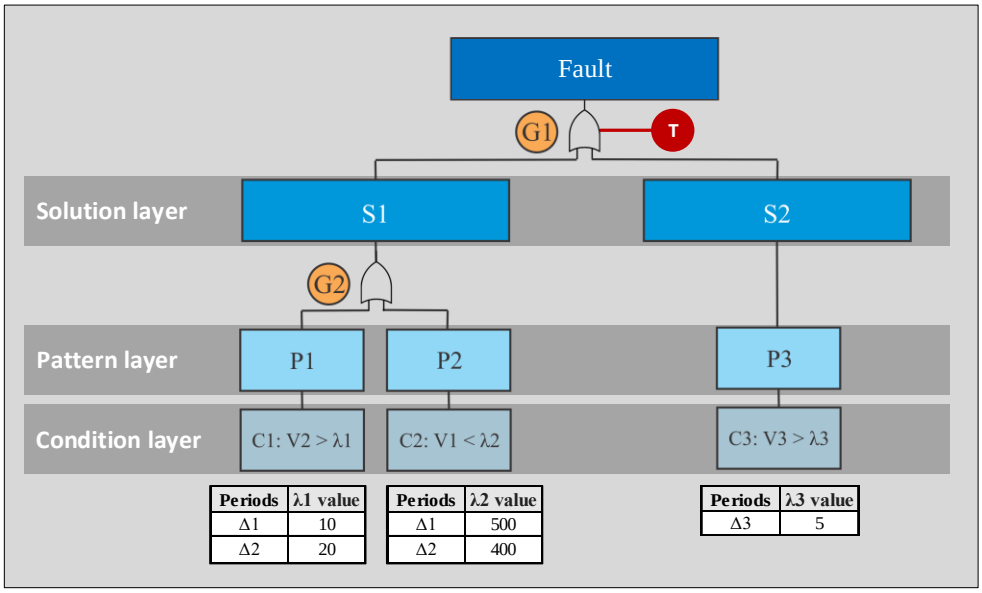
6.3.3 Phase 3: The ITCA model construction

Phase 3 builds, in a sequential up-bottom structure, several logic trees to depict unexplained causes at a given level. It includes three stages: verify the common logic trees' construction, connect those trees to their corresponding causes and generate new labeled sub-datasets that exclude the variables associated with causes already explained from the concatenated datasets.

- ***Stage 1: Verify the common logic trees' construction over the defined periods.*** This stage verifies the knowledge representability of the constructed logic tree for each defined period of time and decides whether further decomposition of its involved conditions is required or not. At each decomposition level, verification of the tree knowledge is characterized by the coverage of the common feasible solution, which assists in avoiding decomposing the weak information branches. Therefore, the model construction is verified to sustain the tree at a non-redundant knowledge level based on the pre-set coverage threshold. Meanwhile, the construction phase can be interpreted if there is no common tree that is able to provide sufficient knowledge representability, or if there are no more variables in the dataset for any further root-cause explorations.
- ***Stage 2: Connect the common logic trees to their corresponding causes.*** The applied relaxation in selecting a common feasible solution over the defined periods is very useful in constructing a common logic tree that easily demonstrates the change in the causality at a given level of

decomposition in the ITCA model. However, it could happen if the time-based searching algorithm fails to form only one common logic tree that dominates all of the defined periods at a certain decomposition level. This case could happen if there is a lack of extracted knowledge or a tight range in the solution tolerance selection (STS). To solve this situation, different common logic trees may be found by the algorithm, but each period of time is dominated by only one common feasible solution. Therefore, if such a situation rises, a time-OR gate is proposed to connect the different common logic trees to represent the change in the event causality knowledge over all of the defined periods at a given decomposition level. The time-OR gate acts as a time switch that shifts between the common logic trees according to their corresponding periods. Hence, an expert could observe the fault behavior over time based on the proposed common similar solution trees at a certain decomposition level of the ITCA model.

Figure 6.5 presents an example of the time-OR gate functionality in a one level ITCA model. Two common feasible solutions, S_1 and S_2, are found by the time-based searching algorithm. S_1 characterizes the fault event at only Δ_1 and Δ_2 using the OR gate (G2) between the patterns P_1 and P_2. While S_2 represents the fault even only at only Δ_3 with a one pattern P_3. The time-OR gate (G1) enables ITCA to fully demonstrate the fault event causality over the three defined periods (Δ_1, Δ_2 and Δ_3). It switches between S_1 and S_2 according to the selected corresponding period that is dominated by the solution. For instance, at the periods Δ_1 and Δ_2, the time-OR gate (G1) enables only S_1 to depict the fault event causality. On the other hand, during the period Δ_3 , the fault causality is explained only by S_2.



Where (λ) is the condition cutpoint values that dominate certain periods

Figure 6.5 Time-OR gate functionality in the ITCA model

- ***Stage 3: Generate new (m-1) sub-datasets.*** In a case in which the added common logic trees are verified at a certain decomposition level of the ITCA model, each one of the involved conditions in the tree is used to generate new labeled sub-datasets based on the condition variable cut-point values. Figure 6.6 takes back the example of Figure 6.3. It presents the generation of the three two-class sub-datasets $D_1^{(2)}$, $D_2^{(2)}$ and $D_3^{(2)}$ at the second decomposition level using the variable X_1 cut-point values 10, 20 and 30, respectively. Note that the generated new sub-datasets contain (m-1) columns each time when a variable is removed from the data.

6.3.4 Phase 4-Derive the time causality rules

Based on the calculation of the probabilities of root-causes, causes and fault events in the final ITCA model, Phase 4 derives the time causality rules that represent the change in occurrence probabilities from one period to another. Each time causality rule summarizes a specific structure of the cause-effect relations over the time between the root-causes, causes and fault events within the ITCA model in the form of an algebraic formula based on the above equations 1 to 4 (Stage 4, Phase 2). The obtained time causality rules allow the fault event occurrence to be controlled based only on its root-causes. Moreover, those rules enable managing the fault occurrence over the

defined time horizon, which makes them more suitable and appropriate for the task of making a prognosis.

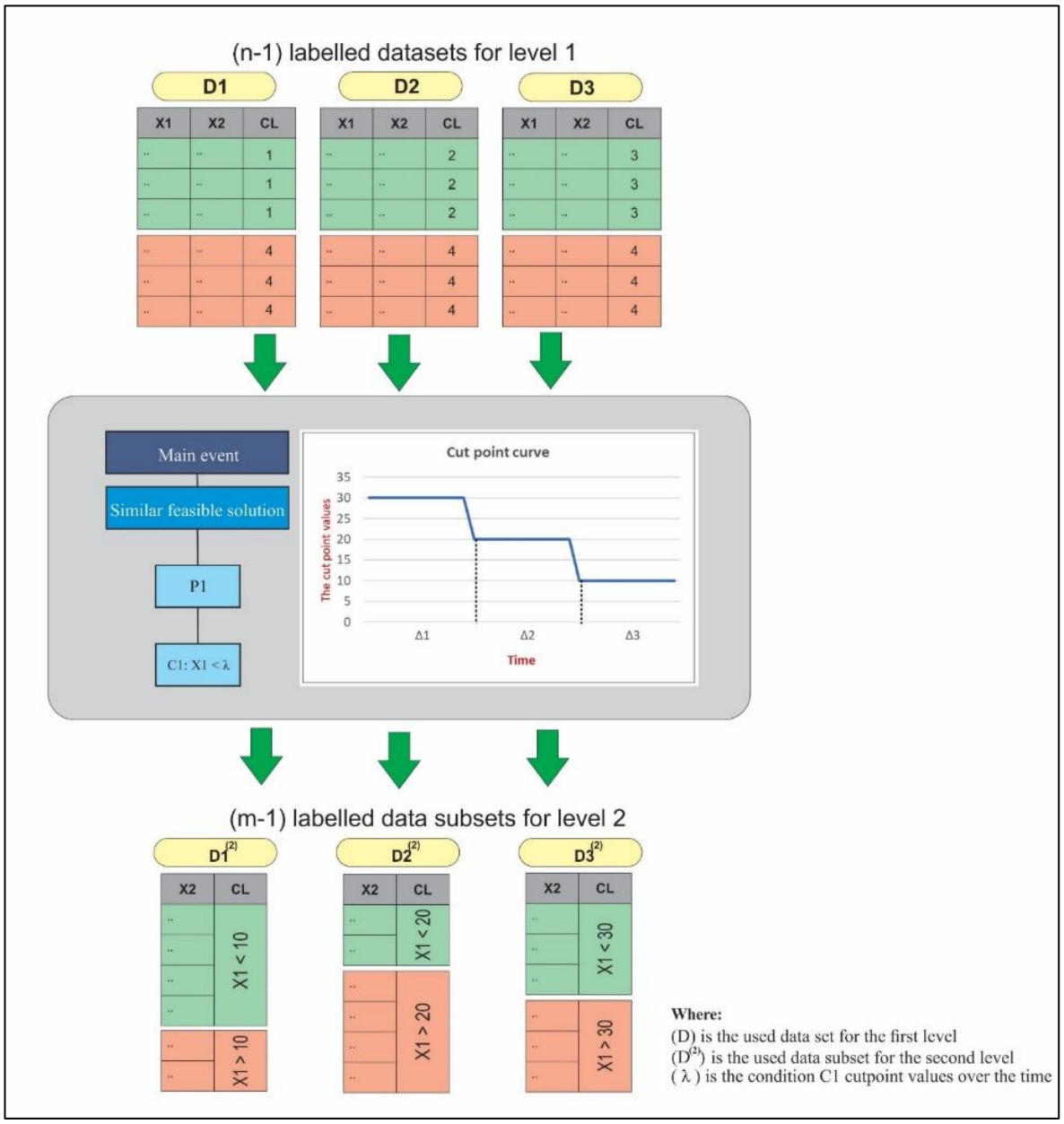


Figure 6.6 Generate new labeled data subsets in the ITCA methodology

6.4 Case study

Most aging systems that include bearings, seals, glands, shafts and couplings are more likely suffer from several degradation processes due to harsh operating constraints such as high temperatures,

vibration and dynamic load, and likely the deficiency of the maintenance plan as well. In this section, the ITCA methodology is deployed on simulated data that reproduce the degradation of a turbofan engine proposed by NASA. It is known as the PHM08 challenge dataset. The dataset is generated by the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) simulator based on MATLAD® and Simulink® (May, Csank, Litt, & Guo, 2010). The simulator uses the combination of three specific operation variables to generate different degradation profiles. The high-pressure compressor (HPC) degradation fault mode is selected as an illustrative example.

Based on the C-MAPSS user guide, the engine consists of several interconnected subsystems (inlet, bypass nozzle, fan, low-pressure compressor (LPC), high-pressure compressor (HPC), combustor, high-pressure turbine (HPT), low-pressure turbine (LPT) and core nozzle). The fuel valve controls the fuel flow into the combustor that turns the HPT. The HPT rotates the HPC, LPT, LPC and the inlet fan. The turbofan engine has two state variables, the fan speed and the core speed (Liu, Yuan, Frederick, DeCastro, Litt, & Chan, 2012) as shown in Figure 6.7.A. Based on the thermodynamic cycle, the air is compressed and combusted by the engine to produce propelling. Figure 6.7.B describes the ambient airflow to the engine. First, the air enters the engine through the inlet and the fan. Then, it is divided by the splitter into two portions. One portion passes through the compressor and then the burner to mix with fuel and produces combustion. The hot exhaust passes through the core and fan turbines to the nozzle, while the other portion is bypassed to the back of the engine. The airflow is controlled by the bypass ratio, which is the ratio of the bypassed mass airflow to the mass airflow that goes through an engine core (National Aeronautics and Space Administration NASA, 2015). The HPC's main functionality drives the airflow to higher pressure and temperature states to prepare it for combustion by using its spinning blades. Therefore, the change in the bypass ratio is the main control element for controlling the HPC outlet air pressure and its temperature for the burning phase.

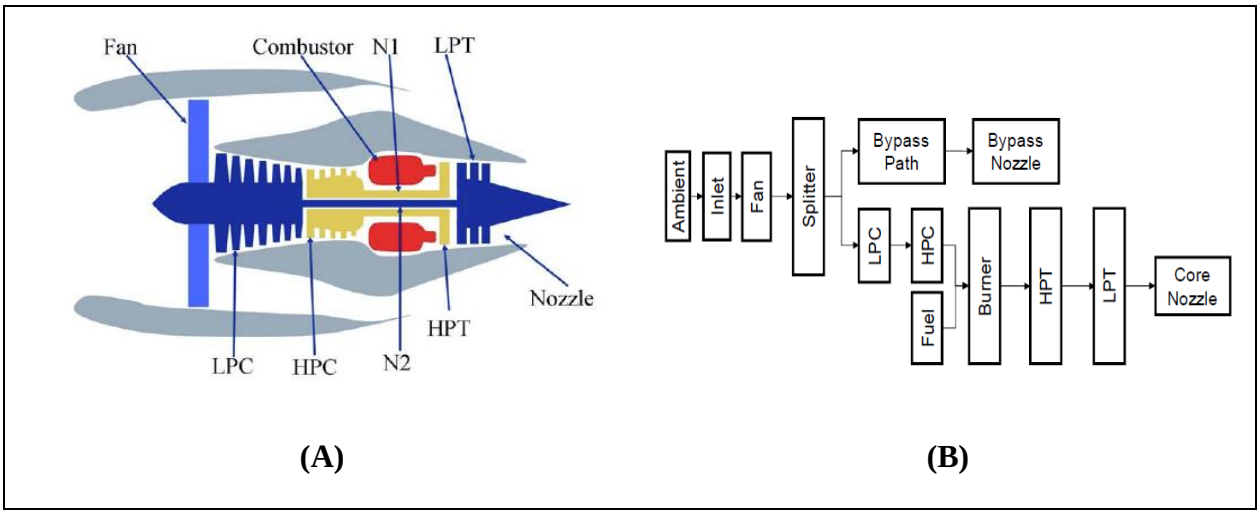


Figure 6.7 The simulated turbofan engine based on C-MAPSS (Frederick, DeCastro, & Litt, 2007)

The challenge addressed by the ITCA methodology is to model the HPC fault causality structure in a dynamic manner, so that the model can demonstrate the effect of the root-cause changes over time on the main HPC degradation curve.

6.4.1 Dataset description

The dataset consists of 21 measurement variables that describe the HPC fault mode (Table 6.1) and 465 timestamp observations. The generated data is divided into training and testing sets with 258 (60%) and 207 observations (40%), respectively. The constant (—), increasing (↑) or decreasing (↓) trend that depicts each variable over time is mentioned in Table 6.1.

Table 6.1 Variable descriptions of the HPC fault mode

Variable	Description (Unit)	Trend (—,↑,↓)	Variable	Description (Unit)	Trend (—,↑,↓)
T2	Total temperature at fan inlet (R)	—	**phi**	Ratio of fuel flow to Ps30 (pps/psi)	↓
T24	Total temperature at LPC outlet (R)	↑	**NRf**	Corrected fan speed (rpm)	↑
T30	Total temperature at HPC outlet (R)	↑	**NRc**	Corrected core speed (rpm)	↓
T50	Total temperature at LPT outlet (R)	↑	**BPR**	Bypass Ratio (rpm)	↑
P2	Pressure at fan inlet (psia)	—	**farB**	Burner fuel-air ratio (without unit)	—
P15	Total pressure in bypass-duct (psia)	—	**htBleed**	Bleed Enthalpy (without unit)	↑
P30	Total pressure at HPC outlet (psia)	↓	**Nf_dmd**	Demanded fan speed (rpm)	—
Nf	Physical fan speed (rpm)	↑	**W31**	HPT coolant bleed (lbm/s)	↓
Nc	Physical core speed (rpm)	↓	**W32**	LPT coolant bleed (lbm/s)	↓
epr	Engine pressure ratio	—	**Ps30**	Static pressure at HPC outlet (psia)	↓
PCNfR_dmd	Demanded corrected fan speed (rpm)	—			

Note that the majority of the variables have an increasing or a decreasing trend over the time, except T2, P2, P15, epr, farB, Nf_dmd and PCNfR_dmd, which are constant no matter the fault mode.

6.4.2 The HPC fault prognosis using the ITCA model

In what follows, the main results of the proposed four-phase ITCA methodology applied on the NASA turbofan engine dataset are presented and discussed in order to perform the HPC fault prognosis task. As per the first phase, the training dataset is ordered according to the timestamp variable and divided into six equal, unlabeled subsets, where each subset SS_i $i = 1..6$ depicts the

period of time Δ_i $i = 1..6$. The subsets are ordered in a timely manner, where SS_1 represents the best normal state of the turbofan while SS_6 depicts its worst or failure state. Consequently, five labeled datasets are concatenated from those 6 subsets as follows D_i: SS_i versus SS_6 $i = 1..5$. Each dataset has 86 labeled observations. Note that the dataset is divided by fixed width for simplicity. However, the expert can assign different width thresholds to produce non-equal data subsets. Meanwhile, the number of subsets is important to capture the evolution of the faults over time. This is a trade-off between time step resolution and ITCA construction time. Phase 2 and Phase 3 are iteratively repeated to construct the ITCA model. The coverage tolerance selection STS threshold used by the time-based searching algorithm (Stage 2 of Phase 2) is set to 10%. In addition, the coverage threshold is set to 90% to control redundant knowledge in the common trees at Stage 1 of Phase 3, when a new level is considered in the ITCA model.

Figure 6.8 depicts the final ITCA model of the HPC fault mode. It includes six levels of decomposition to reproduce the causality structure between the HPC fault and its root-causes over six periods of time. Note that each level of the ITCA model consists of three layers that represent the solutions, patterns and conditions related to the fault event or to one of its causes. The first level includes only one common feasible solution S_1 over the 5 defined time periods (Δ_1 to Δ_5). S_1 has only one pattern P_1 which includes only one condition C_1: $P30 > \lambda_1$. The plot A1 of Figure 6.8 characterizes the degradation of the variable P30 over time. Note that the cut-point curve (blue line) bounds the trend of the variable P30 in time. Additionally, the plot A2 of Figure 6.8 shows the common feasible solution coverage and the overlap percentages over the five time periods. Regarding level 2 of the ITCA model, the same interpretation above can be done for the variable T50. It is clear that the ITCA model captures the trend of the involved variables based on the cut-point curves.

At level 3, two common feasible solutions, S_3 and S_4, are found by the time-based searching algorithm. These solutions respect the construction setting, S_3 explains the cause (C_2: $T50 \leq \lambda_2$) at the time periods Δ_1 and Δ_2. While S_4 dominates the three other periods Δ_3 to Δ_5. S_3 and S_4 each have one only one pattern and condition. The plots C2 and D1 of Figure 6.8 depict the bordering of the cut-point curves that represent the degradation trends of the variables T24 and NF, respectively. Meanwhile, the C1 and D2 plots show the solution coverage and overlap percentages over the corresponding time periods. S_3 and S_4 descript the full-time causality of the cause event

(C_2: $T50 \leq \lambda_2$) through the time-OR gate by toggling between the two feasible solutions. Hence, S_3 explains the event causality at only Δ_1 and Δ_2 while S_4 illustrates the causality of the same event at Δ_3, Δ_4 and Δ_5.

At level 4, two other feasible solutions S_5 and S_6 are found that explain the events C_3: $T24 \leq \lambda_3$) and C_4: $NF > \lambda_4$, respectively. At level 5, only one common feasible solution S_7 is found that explains both events' - C_5: $Ps30 \leq \lambda_5$ and C_6: $Phi \leq \lambda_6$ - causality over the five periods of time. This solution includes one pattern P_7 with only one condition C_7: $NRF \leq \lambda_7$. The same reasoning can be made with the only common feasible solution S_8, which explains the condition C_7 at the last level of the ITCA model using only one pattern P_8 that consists of one root-cause C_8: $BPR \leq \lambda_8$. The cut-point curve of Figure 6.8.H1 bounds the trend of C8.

From the obtained logic tree of Figure 6.8, it is clear that the ITCA model confirms the discussion above about the main root-cause of the HPC fault mode. Effectively, the first level of the ITCA model identifies the variable P30 (total pressure at HPC outlet) as the only fault indicator of the HPC degradation over time. Therefore, P30 can be employed to predict the remaining useful time of the turbofan engine according to the HPC fault mode. At the second level, the variable T50 (total temperature at an LPT outlet) is discovered to explain the effect of the temperature of combustion on the total pressure at the HPC outlet. T50 refines the knowledge discovered about P30. The same reasoning continues until reaching the final level, 6, where the ITCA model discovers the variable BPR (Bypass Ratio), which is identified by the expert as the main control element that affects the occurrence of the HPC fault mode over time. Therefore, the ITCA model provides the expert with more refined knowledge until grasping the effects of the root-causes on the fault trend over the time, which help him to achieve the prognosis task in an efficient way.

Figure 6.8 Obtained ITCA model of the HPC degradation mode

The probabilities associated with the ITCA model are calculated using equations 1 to 4 of Stage 4, Phase 2. They quantify the occurrence of similar feasible solutions, patterns and associated conditions period after period at each level of the ITAC model. Figure 6.9 plots the probabilities of the 8 discovered conditions over five periods of time. Note that the occurrence of each feasible solution is equal to the probability of its associated conditions due to the structure of the obtained logic tree. For example, plot A in Figure 6.9 represents the probability curve of $S_1 : P_1 : C_1$ over the periods Δ_1 to Δ_5. The maximum probability value is equal to 0.1$\underline{6}$ at each time period, since the original data is divided into six equal-size data subsets. Therefore, each subset represents 0.1$\underline{6}$ from the original data size. Note that each common feasible solution tries to maximize its class coverage,

so that the associated condition probability value may not exceed that coverage value over the five periods.

Based on the ITCA model and the calculation of probabilities, only one time causality rule can be derived over 5 investigated periods as follows: $\mathcal{P}\big(HPC(\Delta_i)\big) = \mathcal{P}\big(C_8(\Delta_i)\big)\ i = 1 \ldots 5$ (Eq.5). The time causality rule expresses the contribution of the root-cause on the occurrence of the HPC fault, period after period, according to the C_8 cut-point curve. Since each cut-point value provides the essential knowledge to sustain the turbofan for more or less time in each defined period interval through the maintenance action. For instance, the turbofan can spend more time in Δ_1 by making the C_8 variable (BPR) value under the corresponding cut-point value for a set of time.

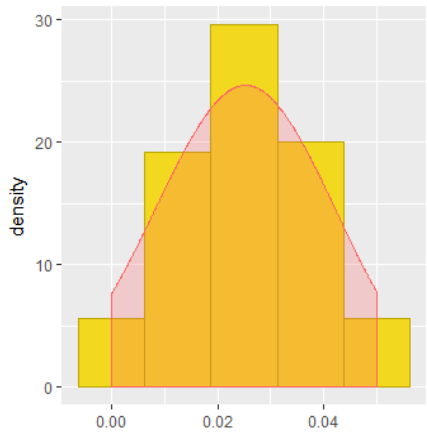
Figure 6.9 Probabilities calculation of the HPC fault mode

6.4.3 Validation of the ITCA model

The accuracy of the obtained ITCA model is quantified using the testing dataset. Five concatenated datasets are formed to represent the five periods of time in the same manner as the data preparation of the training datasets. The mean and the standard error for each period are calculated using the time causality rule and 1000 random data samples; each has a size of 135 observations that provides a 95% confidence level, as shown in Figure 6.10. Based on the error in each period, an average error distribution is generated over the five periods.

From another point of view, the variables T2, P2, P15, epr, farB, Nf_dmd and PCNfR_dmd are not considered in the ITCA model of Figure 6.8 due to the fact that they have a constant trend over time (see Table 6.1). However, the variables NC, NRc, htBleed, W31 and W32 have a changeable

trend, but are not included in the ITCA model. To investigate this situation, the correlation matrix between those omitted variables and those already considered in the ITCA model are measured as depicted in Table 6.2. In each column, the red cell shows the maximum correlation value. The variables NRc, htBleed, and both W31 and W32 are correlated to the variables phi, Nrf and Ps30, respectively, with a correlation value that is higher than 0.6. Except for the variable NC, which measures the physical core speed, which is correlated to P30 with the highest absolute value of 0.17. Accordingly, it seems to be relevant for the HPC degradation. It could be overlooked by the ITCA model.



$M_{error} = 2.51\ 10^{-2}$ and $S_{error} = 1.29\ 10^{-2}$

Figure 6.10 Accuracy of the ITCA model

Table 6.2 Correlation matrix

	NC	NRc	htBleed	W31	W32
T24	-0.159	-0.502	0.595	-0.629	-0.614
T30	-0.211	-0.459	0.534	-0.543	-0.582
T50	-0.153	-0.548	0.644	-0.727	-0.699
P15	-0.001	-0.014	0.065	-0.059	-0.107
P30	**0.175**	0.588	-0.651	0.718	0.739
Nf	-0.167	-0.594	0.707	-0.750	-0.743
Ps30	-0.171	-0.594	0.689	**-0.761**	**-0.742**
phi	0.158	**0.616**	-0.688	0.722	0.721
NRf	-0.169	-0.582	**0.708**	-0.746	-0.715
BPR	-0.184	-0.575	0.605	-0.663	-0.714

6.5 Conclusion

This paper has proposed an interpretable time causality analysis (ITCA) methodology in aging systems. The ITCA model represents the fault hierarchy causality by using the logic of the fault tree graphical and the knowledge discovery in the dataset. The obtained tree models the effect of the system's aging on the changes in the fault causality structure over time for better achieving the fault prognosis task. The illustrated case study demonstrates its usefulness and ability to discover only the relevant root-cause that impacts the fault behavior. Based on the model interpretability, the expert able to understand the time causality structure of the turbofan HPC degradant performance and support his decision through the model's interpretability. Thus, the ITCA model provides the expert with the deep causality knowledge that explain the fault evolution over time. Unlocking the data-driven model's complexity by providing an interpretable graphical model, in addition to summarizing the evolution of the fault over time in one interpretable model, are the two major contributions of the ITCA model over the current time causality data-driven models for fault

prognosis. The ITCA model takes a further step towards reinforcing the link between the experts and the data-driven models. Such a model will help experts elucidate and implement inside the maintenance decision making process.

Our next research work will be to assist the expert by better optimizing the system performance through a set of control actions after understanding the problem causality. We hope to allow our future ITCA model demonstrate the system's reaction regarding a set of proposed control actions based on its causality rules. The expert still needs to observe this system's reaction represented by the new fault causality structure that reflects the system's response for the causality rule control actions that were taken. Therefore, the future ITCA model has to include different scenarios for fault causality structures that reflect the impact of the different combination of control actions based on the derived causality rules.

CHAPTER 7 GENERAL DISCUSSION

In this chapter, the three proposed models are discussed. Strengths and limitations of each tree are highlighted to verify the achievement of the main thesis objective, which is toward the automatic construction of interpretable data-driven fault tree models for fault diagnosis and prognosis in complex systems. The application of the data-driven models to automate risk management in industrial systems proves that this research domain can take the lead in the future. Moreover, the significant prediction accuracy of these models demonstrates that they are adequate solutions for predicting the different system states and recommend decisions in managing faults in complex systems. However, unlike the event-based models, most of the data-driven models are considered to be black box techniques, which misses knowledge interpretability, since the fitted models that predict a fault are complex to interpret by a human expert.

Therefore, extracting interpretable knowledge that characterizes a fault's occurrence and its causality structure contributes to the strength of CBM decision making. The data-driven interpretability limitation is addressed by several techniques, such as the Granger causality analysis, Bayesian network, decision tree and the rules-based model, all of which are able to discover the possible causes of a fault in the form of interpretable patterns. However, those techniques lack representation of a full, understandable graphical representation of a fault causality, which is one of the advantages of event-based models.

The proposed ILTA, MILTA and ITCA models provide a compromise for fault diagnosis and prognosis. The novelty is the linking of data-driven and event-based models to solve the fault diagnosis and prognosis problems. The need for prior information and specialist training in the event-based models is overcome by extracting interpretable knowledge directly from the system's database. The need to unlock a data-driven black box and understand their structures is addressed using the fault tree representation, which drives direct, understandable knowledge to the expert without the need for other, prior analyses.

The proposed ILTA-model (Chapter 4) introduced the main concepts of linking the model-based causality analysis method (FTA) with the knowledge discovery in database (KDD). The extract knowledge from the system database represents causality regarding fault occurrence. Unlike the model-based model, the discovered knowledge quantifies the fault causality more precisely when

compared with the assigned causes by an expert. In addition, the discovered causes based on KDD offer unbiased knowledge about the fault causality, which is a major limitation in the classic FTA, in addition to exploring the hidden fault causality phenomena that might otherwise be overlooked by an expert. The ILTA methodology boosts the advantages of the discovered causality knowledge from the system database by removing the redundant knowledge by introducing the concept of feasible solutions. Therefore, it is a hybrid model that is able to automatically construct the causality structure directly from the database with minimum involvement from human experts. Meanwhile, representing this non-redundant knowledge in a graphical model leads to easy understanding.

However, using the database to perform an analysis through the ILTA-model takes into account only the events that have occurred in the past, which leads to the absence of potential scenarios that might still occur, but have not been observed previously. This is one of the major drawbacks of the data-driven models. To overcome this limitation in the ILTA-model, the experts can enrich the obtained tree by removing some redundant scenarios (i.e. feasible solutions) or adding unobserved scenarios due to the limitation of the sample's representability. In addition, the represented fault causality events and their occurrence probabilities are very useful when the used dataset is known to be representative, complete and accurate. The quality of available data and its representability are the major challenges facing the data-driven model. The ILTA-model is similar to any data-driven model that needs reliable and representative data to provide relevant results. The entire effort of this thesis focuses on constructing relevant and accurate new fault trees for fault diagnosis and prognosis. Therefore, the challenges related to the quality of the available data and the effort made to acquire and prepare them, with adequate quality, could be addressed in our future research.

The proposed MILTA-model (Chapter 5) presented the concept of building a multi-level graphical tree that aggregates different dependant ILTA trees. The MILTA-model overcomes the limitation of a one level tree for fault diagnosis in complex systems. The relation structure between the root-causes, intermediate causes and faults by connecting the dependant ILTA tree depict the causal relationships. The interpretability of the MILTA-model enables the expert to grasp the fault occurrence and its root-causes. Meanwhile, the expert has to validate the constructed tree, rather than to be involved in constructing the event-based causality model from scratch, such as in the FTA.

The contribution of a MILTA-model over the current data-driven models concerns the graphical representation that automatically decomposes the fault into its hierarchical structure. Priority is given to the causes that have a strong, direct relation with the occurrence of the fault. Consequently, the fault indicators, the intermediate causes and the root-causes are discovered level after level. The process of ordering and representing the different fault events provides an expert with an easy and familiar model to understand the fault causality in complex systems.

Despite the advantages of the MILTA-model, the main drawback concerns the cause-and-effect of the fault and the correlation between its causes. This added constraint, selecting the minimal number of patterns that maximize the class coverage to the burn-and-build algorithm, helps capture the fault's hierarchical causality. The algorithm promotes selecting the feasible solutions that have a strong direct relation with the main fault event at the first level. However, it could fall to distinguishing the differences between the cause-and-effect relationship between two causes to assign them to different levels. Another drawback concerns how to assure that the last levels of the tree include the root-causes. The solution's coverage percentage threshold is the main criterion to stop the fault decomposition process. Therefore, it is a trade-off between adding low solution coverage and obtaining root causes, which depend on the data sample's representability. Thus, the expert has to validate the final constructed tree to ensure that the MILTA-model includes the fault's root-causes, Otherwise, he will assign a lower threshold to overcome the lack of representability in the data sample.

The proposed ITCA-model (Chapter 6) addressed the effect of system wear-out on the changes in the fault causality structure for the prognosis task. The ITCA-model introduces a multi-level tree similar to the MILTA structure to determine the fault time causality. Since the fault causality structure changes due to the system's degradation, the causality knowledge represented through different decomposition levels will be updated over time. Therefore, the ITCA-model provides an expert with more sufficient knowledge regarding the changes in the fault causality structure as the system ages.

However, the ITCA-model is based on unlabeled data, unlike the ILTA and MILTA models. This is because there is no state variable that can be used to indicate the differences between the system's normal and faulty states. Paradoxically, the data used is similar to time series data, in which the first observations carry knowledge about the system's normality state. Gradually, the fault and its

effects start to appear progressively in time though the next observations, reaching the last observations that represent the system failure. The ITCA-model captures those transitions over time by splitting the time series observations into several ordered subsets based on the assigned thresholds. Each subset represents a period that summarizes the fault's evolution and its change in the causality structure at that period.

One of the challenges in the ITCA-model construction concerns the sizing of the data subsets, which affect the final tree result. The number of assigned thresholds controls the number of the generated datasets that have a consequence on the tree's resolution in capturing the fault's causality changes. For instance, if the dataset number is too small, large periods are defined and, consequently, some fault development stages can be omitted. On the other hand, if the data subset number is too big, it affects the processing and the tree's construction time. Therefore, the involvement of the expert to choose reasonable thresholds that define the datasets size is required. In addition, the same MILTA challenges are considered in the ITCA-model, such as distinguishing between the cause-and-effect relationship and event correlations, besides assuring the discovery of the fault root-causes through the tuning of a feasible solution coverage threshold.

CHAPTER 8 CONCLUSION AND RECOMMENDATIONS

The three trees take a further step to graphically model the fault causality structure and its time evolution in order to provide the expert with a better understanding of the fault cause and effect relationships. However, the implementation of those trees in order to conduct an automatic control framework that manages the complex fault through maintenance actions still needs to be developed.

Our future research will focus on linking the trees with the system control to automate the fault management. The time causality rules, based on the Bayesian theorem, are very useful to quantify the importance of the discovered root-causes by ranking the root causes according to their contribution to fault occurrence. However, there is a lack of application of time causality rules, since they are not linked with the system performance quantification measurement such as its reliability curve. This link is very crucial, since it helps the expert optimizing the timing of maintenance decision actions. The time causality rules with its associated tree is able to provide the expert with all the needed information regarding the fault causality at a certain time period. While, the system performance quantification measurement will depict the degradation in the system performance at that time period for better grasping the fault consequences with the system performance.

In addition, the proposed ITCA model that dominates the MILTA and the ILTA models for achieving the fault diagnosis and prognosis is able to capture the effect of the system aging on the changes of the fault causality structure. However, it cannot depict the effect of minimal repairs on changing the fault causality structure besides the system aging. Therefore, new data has to be collected to capture those new changes in the system and the faults. The proposed ITCA model have to include different fault causality structure scenarios that link the fault causality rules with the system's responses. Therefore, control can be performed in a dynamic manner, where an expert can understand the system's response and limitations toward assigning more optimal procedures. By achieving this crucial point, the system fault causality becomes fully modeled. In a sense, the fault causality structure is well-depicted, aside from the system's response and the changes in the fault causality.

REFERENCES

Abbas, M., & Shafiee, M. (2020). An overview of maintenance management strategies for corroded steel structures in extreme marine environments. *Marine Structures, 71*. doi:10.1016/j.marstruc.2020.102718

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics, 2*(4), 433-459.

Abid, F. B., Sallem, M., & Braham, A. (2019). *Optimized SWPT and Decision Tree for Incipient Bearing Fault Diagnosis.* Paper presented at the 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, STA 2019, March 24, 2019 - March 26, 2019, Sousse, Tunisia (pp. 231-236).doi:10.1109/STA.2019.8717197

Aggab, T., Kratz, F., Avila, M., & Vrignat, P. (2018). Model-based prognosis applied to a coupled four tank MIMO system. *IFAC - Papers Online, 51*(24), 655-661. doi:10.1016/j.ifacol.2018.09.645

Ahmadi, R. (2012). *Maintenance scheduling of a manufacturing system subject to linear deterioration model.* Paper presented at the Proceedings of the 18th ISSAT international conference on reliability and quality in design (pp. 355-363).

Alauddin, M., Khan, F., Imtiaz, S., & Ahmed, S. (2018). A bibliometric review and analysis of data-driven fault detection and diagnosis methods for process systems. *Industrial & Engineering Chemistry Research, 57*(32), 10719-10735.

Alexe, S., & Hammer, P. L. (2006). Accelerated algorithm for pattern detection in logical analysis of data. *Discrete Applied Mathematics, 154*(7), 1050-1063.

Alizadeh, E., El Koujok, M., Ragab, A., & Amazouz, M. (2018). A Data-Driven Causality Analysis Tool for Fault Diagnosis in Industrial Processes. *IFAC-PapersOnLine, 51*(24), 147-152.

Alizadeh, E., El Koujok, M., Ragab, A., & Amazouz, M. (2018). A Data-Driven Causality Analysis Tool for Fault Diagnosis in Industrial Processes. *IFAC - Papers Online, 51*(24), 147-152. doi:10.1016/j.ifacol.2018.09.548

Alkinani, H. H., Al-Hameedi, A. T. T., Dunn-Norman, S., Alsaba, M. T., & Amer, A. S. (2019). *Review of the applications of decision tree analysis in petroleum engineering with a rigorous analysis.* Paper presented at the SPE Middle East Oil and Gas Show and Conference 2019, MEOS 2019, March 18, 2019 - March 21, 2019, Manama, Bahrain (Vol. 2019-March).

Amblard, P. O., & Michel, O. J. J. (2013). The Relation between Granger Causality and Directed Information Theory: a Review. *Entropy, 15*(1), 113-143. doi:10.3390/e15010113

Amin, A. A., & Hasan, K. M. (2019). A review of Fault Tolerant Control Systems: Advancements and applications. *Measurement: Journal of the International Measurement Confederation, 143* 58-68. doi:10.1016/j.measurement.2019.04.083

Analouei, R., Taheriyoun, M., & Safavi, H. R. (2020). Risk assessment of an industrial wastewater treatment and reclamation plant using the bow-tie method. *Environmental Monitoring and Assessment, 192*(1). doi:10.1007/s10661-019-7995-x

Ashok, V., & Yadav, A. (2018). *A Novel Decision Tree Algorithm for Fault Location Assessment in Dual-Circuit Transmission Line Based on DCT-BDT Approach.* Paper presented at the International Conference on Intelligent Systems Design and Applications (pp. 801-809).

Ayyub, B. M. (2014). *Risk analysis in engineering and economics*: Crc Press.

Badida, P., Balasubramaniam, Y., & Jayaprakash, J. (2019). Risk evaluation of oil and natural gas pipelines due to natural hazards using fuzzy fault tree analysis. *Journal of Natural Gas Science and Engineering* 284-292. doi:10.1016/j.jngse.2019.04.010

Baig, A. A., Ruzli, R., & Buang, A. B. (2013). Reliability analysis using fault tree analysis: a review. *International Journal of Chemical Engineering and Applications, 4*(3), 169.

Bakdi, A., & Kouadri, A. (2017). A new adaptive PCA based thresholding scheme for fault detection in complex systems. *Chemometrics and Intelligent Laboratory Systems, 162* 83-93. doi:https://doi.org/10.1016/j.chemolab.2017.01.013

Balasubramanian, A., & Muthu, R. (2017). Model based fault detection and diagnosis of doubly fed induction generators - a review. *Energy Procedia, 117* 935-942. doi:10.1016/j.egypro.2017.05.213

Bartyś, M., & Syfert, M. (2001). Using damadics actuator benchmark library (dablib). *Final, ver, 1* 01-20.

Bathelt, A., Ricker, N. L., & Jelali, M. (2015). Revision of the Tennessee Eastman Process Model. *IFAC-PapersOnLine, 48*(8), 309-314. doi:https://doi.org/10.1016/j.ifacol.2015.08.199

Bensaci, C., Zennir, Y., Pomorski, D., & Mechhoud, E.-A. (2017). *Complex safety study of intelligent multi-robot navigation in risk's environment.* Paper presented at the 2017 International Carnahan Conference on Security Technology, ICCST 2017, October 23, 2017 - October 26, 2017, Madrid, Spain (Vol. 2017-October, pp. 1-6).doi:10.1109/CCST.2017.8167809

Beresh, R., Ciufo, J., & Anders, G. (2007). *Basic fault tree analysis for use in protection reliability.* Paper presented at the Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, 2007. PSC 2007 (pp. 1-7).

Berger, H., Merkl, D., & Dittenbach, M. (2006). *Exploiting partial decision trees for feature subset selection in e-mail categorization.* Paper presented at the Proceedings of the 2006 ACM symposium on Applied computing (pp. 1105-1109).

Bernardo, J. M., & Smith, A. F. (2001). Bayesian theory: IOP Publishing.

Bezerra Viana, I., Sandoval Goes, L. C., & Conceicao Rocha, G. (2013). *An integrated hybrid methodology of time series forecast and case-based reasoning for fault prognosis.* Paper presented at the 2013 IEEE Conference on Prognostics and Health Management (PHM), 24-27 June 2013, Piscataway, NJ, USA (pp. 9 pp.).doi:10.1109/ICPHM.2013.6621420

Bhagavatula, A., Tao, J., Dunnett, S., & Bell, P. (2016). *A new methodology for automatic fault tree construction based on component and mark libraries.* Paper presented at the Safety and Reliability (Vol. 36, pp. 62-76).

Bianchini, A., Rossi, J., & Antipodi, L. (2018). A procedure for condition-based maintenance and diagnostics of submersible well pumps through vibration monitoring. *International Journal*

of System Assurance Engineering and Management, 9(5), 999-1013. doi:10.1007/s13198-018-0711-3

Bingamil, A., Alsyouf, I., & Cheaitou, A. (2017). *Condition monitoring technologies, parameters and data processing techniques for fault detection of internal combustion engines: A literature review.* Paper presented at the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA) (pp. 1-5).

Biswas, S., Dash, R. N., Choudhury, K., & Sahoo, S. P. (2018). *A review paper on inrush fault isolation methods of a three-phase transformer.* Paper presented at the 2018 Technologies for Smart-City Energy Security and Power (ICSESP) (pp. 1-5).

Board, N. E. (2017). Canada's Pipeline Transportation System 2016. Retrieved from https://www.neb-one.gc.ca/nrg/ntgrtd/trnsprttn/2016/cnds-ppln-trnsprttn-systm-eng.html

Bonates, T. O., Hammer, P. L., & Kogan, A. (2008). Maximum patterns in datasets. *Discrete Applied Mathematics, 156*(6), 846-861.

Boros, E., Crama, Y., Hammer, P. L., Ibaraki, T., Kogan, A., & Makino, K. (2011). Logical analysis of data: classification with justification. *Annals of Operations Research, 188* 33-61. doi:10.1007/s10479-011-0916-1

Boros, E., Hammer, P. L., Ibaraki, T., & Kogan, A. (1997). Logical analysis of numerical data. *Mathematical Programming, 79*(1-3), 163-190.

Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., & Muchnik, I. (2000). An implementation of logical analysis of data. *IEEE Transactions on Knowledge and Data Engineering, 12*(2), 292-306.

Boukra, T. (2015). *Identifying new prognostic features for remaining useful life prediction using particle filtering and Neuro-Fuzzy System predictor.* Paper presented at the 15th IEEE International Conference on Environment and Electrical Engineering, EEEIC 2015, June 10, 2015 - June 13, 2015, Rome, Italy (pp. 1533-1538).doi:10.1109/EEEIC.2015.7165399

Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2018). Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance. *Journal of Intelligent Manufacturing, 29*(6), 1303-1316. doi:10.1007/s10845-015-1179-5

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Burkhalter, M., Martani, C., & Adey, B. T. (2018). Determination of risk-reducing intervention programs for railway lines and the significance of simplifications. *Journal of Infrastructure Systems, 24*(1). doi:10.1061/(ASCE)IS.1943-555X.0000401

Cao, Y., Wei, Y., Hu, G., Zhou, Z., Han, X., & He, W. (2019). *BRB fault diagnosis model based on fault tree analysis.* Paper presented at the 2018 Chinese Automation Congress, CAC 2018, November 30, 2018 - December 2, 2018, Xi'an, China (pp. 783-787).doi:10.1109/CAC.2018.8623670

Celaya, J., Kulkarni, C., Biswas, G., Saha, S., & Goebel, K. (2011). A model-based prognostics methodology for electrolytic capacitors based on electrical overstress accelerated aging.

Chelson, P. O. (1971). Reliability computation using fault tree analysis.

Chemweno, P., Pintelon, L., Muchiri, P. N., & Van Horenbeek, A. (2018). Risk assessment methodologies in maintenance decision making: A review of dependability modelling approaches. *Reliability Engineering & System Safety, 173* 64-77. doi:https://doi.org/10.1016/j.ress.2018.01.011

Chen, B., Matthews, P. C., & Tavner, P. J. (2013). Wind turbine pitch faults prognosis using a-priori knowledge-based ANFIS. *Expert Systems with Applications, 40*(17), 6863-6876. doi:10.1016/j.eswa.2013.06.018

Chen, H.-S., Yan, Z., Zhang, X., Liu, Y., & Yao, Y. (2018). Root Cause Diagnosis of Process Faults Using Conditional Granger Causality Analysis and Maximum Spanning Tree. *IFAC-PapersOnLine, 51*(18), 381-386. doi:https://doi.org/10.1016/j.ifacol.2018.09.330

Chen, X., Wang, J., & Zhou, J. (2018). Process Monitoring Based on Multivariate Causality Analysis and Probability Inference. *IEEE Access, 6* 6360-6369. doi:10.1109/ACCESS.2018.2795535

Chen, Y., Fang, X., Yang, L., Liu, Y., Gong, C., & Di, Y. (2019). *Artificial Neural Networks in the Prediction and Assessment for Water Quality: A Review.* Paper presented at the 2019 4th International Conference on Intelligent Computing and Signal Processing, ICSP 2019, March 29, 2019 - March 31, 2019, Xi'an, China (Vol. 1237).doi:10.1088/1742-6596/1237/4/042051

Cheng Geng, H., Xiang Yu, L., Hong Zhong, H., & Yan Feng, L. (2019). *Fault prognosis of Engineered Systems: A Deep Learning Perspective.* Paper presented at the 2019 Annual Reliability and Maintainability Symposium (RAMS), 28-31 Jan. 2019, Piscataway, NJ, USA (pp. 7 pp.).doi:10.1109/RAMS.2019.8768982

Chidaravalli, S., Jenila Livingston, L. M., & Manjunath, T. K. (2017). *Vibration based expert system: Review.* Paper presented at the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2017, July 6, 2017 - July 7, 2017, Kannur, India (Vol. 2018-January, pp. 486-492).doi:10.1109/ICICICT1.2017.8342611

Chikalov, I., Lozin, V., Lozina, I., Moshkov, M., Nguyen, H. S., Skowron, A., & Zielosko, B. (2013). Logical analysis of data: Theory, methodology and applications. In *Three approaches to data analysis* (pp. 147-192): Springer.

Choi, J.-H., Edwards, P., Ko, K., & Kim, Y.-S. (2016). Definition and classification of fault damage zones: A review and a new methodological approach. *Earth-Science Reviews, 152* 70-87.

Choudhary, A. K., Harding, J. A., & Tiwari, M. K. (2009). Data mining in manufacturing: A review based on the kind of knowledge. *Journal of Intelligent Manufacturing, 20*(5), 501-521. doi:10.1007/s10845-008-0145-x

Cook, R. I. (1998). How complex systems fail. *Cognitive Technologies Laboratory, University of Chicago. Chicago IL.*

Corcoglioniti, F., Rospocher, M., Cattoni, R., Magnini, B., & Serafini, L. (2018). The knowledgestore: A storage framework for interlinking unstructured and structured knowledge. In (Vol. 2, pp. 686-721): IGI Global.

Cui, Y., Wu, Y., Du, Y., Xu, N., Zhang, T., & Wang, L. (2018). *Application of Alarm Information Statistical Analysis Based on Protection and Fault Information System.* Paper presented at the 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), 10-12 Dec. 2018, Piscataway, NJ, USA (pp. 719-722).doi:10.1109/IICSPI.2018.8690492

DAMADICS. (2002). DAMADICS Actuator Benchmark. Retrieved from http://diag.mchtr.pw.edu.pl/damadics/

Darwish, M., Almouahed, S., & de Lamotte, F. (2017). The integration of expert-defined importance factors to enrich Bayesian Fault Tree Analysis. *Reliability Engineering & System Safety, 162* 81-90. doi:10.1016/j.ress.2017.01.007

de Jonge, B. (2019). Discretizing continuous-time continuous-state deterioration processes, with an application to condition-based maintenance optimization. *Reliability Engineering and System Safety, 188* 1-5. doi:10.1016/j.ress.2019.03.006

de Jonge, B., Dijkstra, A. S., & Romeijnders, W. (2015). Cost benefits of postponing time-based maintenance under lifetime distribution uncertainty. *Reliability Engineering & System Safety, 140* 15-21. doi:10.1016/j.ress.2015.03.027

Debnath, S., Reddy, J., Jagadish, & Das, B. (2019). Investigation of thermal performance of SAC variables using fuzzy logic based expert system. *Journal of Mechanical Science and Technology, 33*(8), 4013-4021. doi:10.1007/s12206-019-0543-3

Ding, F., & He, Z. (2010). *Formalization of reliability model for assessment and prognosis using proactive monitoring mechanism.* Paper presented at the 2010 Prognostics and System Health Management Conference, PHM '10, January 12, 2010 - January 14, 2010, Macau, China.doi:10.1109/PHM.2010.5413419

Djeziri, M. A., Ananou, B., & Ouladsine, M. (2013). *Data driven and model based fault prognosis applied to a mechatronic system.* Paper presented at the 2013 4th International Conference on Power Engineering, Energy and Electrical Drives, POWERENG 2013, May 13, 2013 - May 17, 2013, Istanbul, Turkey (pp. 534-539).doi:10.1109/PowerEng.2013.6635665

Doukovska, L., & Vassileva, S. (2013). Knowledge-based Mill Fan System Technical Condition Prognosis. *WSEAS Transactions on Systems, 12*(8), 398-408.

Duan, R., Hu, L., & Lin, Y. (2017). Fault diagnosis for complex systems based on dynamic evidential network and multi-attribute decision making with interval numbers. *Eksploatacja i Niezawodność, 19*(4).

Duvvuri, S. S. S. R. S. (2019). *Model-Based Fault Detection and Diagnosis of Slip-Ring Induction Motors: A Simulation Study.* Paper presented at the 8th IEEE India International Conference on Power Electronics, IICPE 2018, December 13, 2018 - December 15, 2018, Jaipur, India (Vol. 2018-December).doi:10.1109/IICPE.2018.8709558

Fokkema, M. (2017). PRE: an R package for fitting prediction rule ensembles. *arXiv preprint arXiv:1707.07149*.

Fokkema, M., & Strobl, C. (2019). Fitting Prediction Rule Ensembles to Psychological Research Data: An Introduction and Tutorial. *arXiv preprint arXiv:1907.05302*.

Frederick, D. K., DeCastro, J. A., & Litt, J. S. (2007). User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS).

Fussell, J. B. (1973). A formal methodology for fault tree construction. *Nuclear Science and engineering, 52*(4), 421-432.

Gao, J., & Zhao, C. (2018, 30 Nov.-2 Dec. 2018). *A Multi-level Bayesian Network Based on Causality Analysis for Fault Diagnosis of Nonstationary Processes.* Paper presented at the 2018 Chinese Automation Congress (CAC) (pp. 1626-1631).doi:10.1109/CAC.2018.8623076

Gao, J., & Zhao, C. (2019). *A Multi-level Bayesian Network Based on Causality Analysis for Fault Diagnosis of Nonstationary Processes.* Paper presented at the 2018 Chinese Automation Congress, CAC 2018, November 30, 2018 - December 2, 2018, Xi'an, China (pp. 1626-1631).doi:10.1109/CAC.2018.8623076

Garga, A. K., McClintic, K. T., Campbell, R. L., Chih-Chung, Y., Lebold, M. S., Hay, T. A., & Byington, C. S. (2001, 10-17 March 2001). *Hybrid reasoning for prognostic learning in CBM systems.* Paper presented at the 2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542) (Vol. 6, pp. 2957-2969 vol.2956).doi:10.1109/AERO.2001.931316

Gilbert, C., Amalberti, R., Laroche, H., & Paries, J. (2007). Errors and failures: towards a new safety paradigm. *Journal of risk research, 10*(7), 959-975.

Gomes, V. S. D., & Bonates, T. O. (2014a). LAD-WEKA. Retrieved from https://lia.ufc.br/~tiberius/lad/index.htm

Gomes, V. S. D., & Bonates, T. O. (2014b). LAD WEKA. Retrieved from https://lia.ufc.br/~tiberius/lad/index.htm

Grabowski, F., & Strzalka, D. (2008). *Simple, complicated and complex systems—the brief introduction.* Paper presented at the 2008 Conference On Human System Interactions (pp. 570-573).

Gul, S. T., Imran, M., & Khan, A. Q. (2018). *An online incremental support vector machine for fault diagnosis using vibration signature analysis.* Paper presented at the 2018 IEEE International Conference on Industrial Technology (ICIT) (pp. 1467-1472).

Guo, C.-x., He, D., Zhang, J.-j., Lu, M.-m., & Yang, X.-y. (2014). A Proportional Hazard Model for oil-immersed transformer concerning thermal aging. *Journal of Electric Power Science and Technology, 29*(2), 14-20.

Guo, Y., Wang, J., Chen, H., Li, G., Huang, R., Yuan, Y., . . . Sun, S. (2019). An expert rule-based fault diagnosis strategy for variable refrigerant flow air conditioning systems. *Applied Thermal Engineering* 1223-1235. doi:10.1016/j.applthermaleng.2018.12.132

Habibi, H., Howard, I., & Simani, S. (2019). Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review. *Renewable Energy* 877-896. doi:10.1016/j.renene.2018.12.066

Halloul, Y., Chiban, S., & Awad, A. (2019). Adapted fuzzy fault tree analysis for oil storage tank fire. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 41*(8), 948-958. doi:10.1080/15567036.2018.1522393

Hamill, M., & Goseva-Popstojanova, K. (2009). Common trends in software fault and failure data. *IEEE Transactions on Software Engineering, 35*(4), 484-496.

Hammer, P. (1986). *The Logic of Cause-effect Relationships: A lecture.* Paper presented at the International Conference on Multi-Attribute Decision Making via Operations Research-bsaed Expert Systems, Passau, Germany.

Hammer, P. L., & Bonates, T. O. (2006). Logical analysis of data—An overview: From combinatorial optimization to medical applications. *Annals of Operations Research, 148*(1), 203-225.

Hammer, P. L., Kogan, A., Simeone, B., & Szedmák, S. (2004). Pareto-optimal patterns in logical analysis of data. *Discrete Applied Mathematics, 144*(1), 79-102. doi:https://doi.org/10.1016/j.dam.2003.08.013

Han, S., He, Y., Zheng, S., & Wang, F. (2019). Intelligent Fault Inference of Inverters Based on a Three-Layer Bayesian Network. *Mathematical Problems in Engineering, 2019*. doi:10.1155/2019/3653746

He, D., Li, R., & Zhu, J. (2012). Plastic bearing fault diagnosis based on a two-step data mining approach. *IEEE Transactions on Industrial Electronics, 60*(8), 3429-3440.

He, X., Tong, J., & Chen, J. (2007). Maintenance risk management in Daya Bay nuclear power plant: PSA model, tools and applications. *Progress in Nuclear Energy, 49*(1), 103-112. doi:10.1016/j.pnucene.2006.10.004

Hidayat, M. E. N., & Hermansyah, H. (2018). *Risk analysis of gas distribution pipelines in apartments using the event tree analysis method.* Paper presented at the 8th Annual Basic Science International Conference, 6-7 March 2018, USA (Vol. 2021, pp. 020004 (020005 pp.)).doi:10.1063/1.5062721

Hu, J., Zhang, L., & Wang, A. (2016). Quantitative safety early warning method of fault propagation for petrochemical plants. *Huagong Xuebao/CIESC Journal, 67*(7), 3091-3100. doi:10.11949/j.issn.0438-1157.20151774

Huang, X., Gao, J., & Gao, Z. (2013). Precise signed digraph modelling based on causal dependence identification. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, 227*(1), 3-16. doi:10.1177/0954408911435745

Inselberg, A. (2009). Parallel coordinates. In *Encyclopedia of Database Systems* (pp. 2018-2024): Springer.

Isermann, R. (2006). Supervision and fault management of processes — tasks and terminology. In *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance* (pp. 13-30). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing, 20*(7), 1483-1510.

Jardine, A. K., & Tsang, A. H. (2005). *Maintenance, replacement, and reliability: theory and applications*: CRC press.

Javed, W., Chen, D., Farrag, M. E., & Xu, Y. (2019). System configuration, fault detection, location, isolation and restoration: A review on LVDC microgrid protections. *Energies, 12*(6). doi:10.3390/en12061001

Jian, Q., Huifang, W., Dongyang, L., & Benteng, H. (2016). *Nonparametric regression-based failure rate model for electric power equipment using lifecycle data.* Paper presented at the 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), 3-5 May 2016, Piscataway, NJ, USA (pp. 1).doi:10.1109/TDC.2016.7519850

Jiang, B., Wu, Y.-K., Lu, N.-Y., & Mao, Z.-H. (2018). Review of fault diagnosis and prognosis techniques for high-speed railway traction system. *Kongzhi yu Juece/Control and Decision, 33*(5), 841-855. doi:10.13195/j.kzyjc.2017.1447

Jichao, H., Zhenpo, W., & Yongtao, Y. (2019). Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks. *Applied Energy, 251* 296-309. doi:10.1016/j.apenergy.2019.113381

Jie, D., Mengyuan, W., Xiong, Z., Liang, M., & Kaixiang, P. (2017). Joint data-driven fault diagnosis integrating causality graph with statistical process monitoring for complex industrial processes. *IEEE Access, 5* 25217-25225. doi:10.1109/ACCESS.2017.2766235

Jin-San, K., Jin-Sung, C., Jae-Goo, B., & Feel-soon, K. (2017). Fault tree analysis and its application for designing high reliability electrical system in underwater vehicle. *Transactions of the Korean Institute of Electrical Engineers, 66*(1), 33-39. doi:10.5370/KIEE.2017.66.1.33

Jin, S., Zhang, Z., Chakrabarty, K., & Gu, X. (2018). *Failure prediction based on anomaly detection for complex core routers.* Paper presented at the 37th IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2018, November 5, 2018 - November 8, 2018, San Diego, CA, United states (pp. ACM; IEEE).doi:10.1145/3240765.3243476

Jin, W. (2016). *Modeling of Machine Life Using Accelerated Prognostics and Health Management (APHM) and Enhanced Deep Learning Methodology.* (ProQuest LLC).

Jocelyn, S., Chinniah, Y., Ouali, M.-S., & Yacout, S. (2017). Application of logical analysis of data to machinery-related accident prevention based on scarce data. *Reliability Engineering & System Safety, 159* 223-236.

Jocelyn, S., Ouali, M.-S., & Chinniah, Y. (2018). Estimation of probability of harm in safety of machinery using an investigation systemic approach and Logical Analysis of Data. *Safety Science, 105* 32-45. doi:https://doi.org/10.1016/j.ssci.2018.01.018

Johnson, C. (2002). A brief overview of causal analysis techniques for electrical, electronic or programmable, electronic systems. *Glasgow: University of Glasgow*.

Johnson, C. (2003). *The application of causal analysis techniques for computer-related mishaps.* Paper presented at the 22nd International Conference on Computer Safety, Reliability and Security, SAFECOMP 2003, September 23, 2003 - September 26, 2003, Edinburgh, United kingdom (Vol. 2788, pp. 368-381).

Kabir, S. (2017). An overview of fault tree analysis and its application in model based dependability analysis. *Expert Systems with Applications, 77* 114-135. doi:https://doi.org/10.1016/j.eswa.2017.01.058

Kaplan, R. S., & Mikes, A. (2012). Managing risks: a new framework. *Harvard Business Review, 90*(6), 48-60.

Kawatsu, K. (2019). *PHM by Using Multi-Physics System-Level Modeling and Simulation for EMAs of Liquid Rocket Engine.* Paper presented at the 2019 IEEE Aerospace Conference, AERO 2019, March 2, 2019 - March 9, 2019, Big Sky, MT, United states (Vol. 2019-March).doi:10.1109/AERO.2019.8741827

Khoo, L., Tor, S., & Li, J. (2001). A rough set approach to the ordering of basic events in a fault tree for fault diagnosis. *The International Journal of Advanced Manufacturing Technology, 17*(10), 769-774.

Khumprom, P., & Yodo, N. (2019). A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm. *Energies, 12*(4). doi:10.3390/en12040660

Kim, K., & Mylaraswamy, D. (2006). *Fault diagnosis and prognosis of gas turbine engines based on qualitative modeling.* Paper presented at the 2006 ASME 51st Turbo Expo, May 6, 2006 - May 11, 2006, Barcelona, Spain (Vol. 2, pp. 881-889).doi:10.1115/GT2006-91210

Kimotho, J. K., Sondermann-Woelke, C., Meyer, T., & Sextro, W. (2013). Application of Event Based Decision Tree and Ensemble of Data Driven Methods for Maintenance Action Recommendation. *International Journal of Prognostics and Health Management, 4*(suppl2), 1-6.

Kiraly, A., Laiho, A., Abonyi, J., & Gyenesei, A. (2014). Novel techniques and an efficient algorithm for closed pattern mining. *Expert Systems with Applications, 41*(11), 5105-5114. doi:10.1016/j.eswa.2014.02.029

Kordes, A., Wurm, S., Hozhabrpour, H., & Wismuller, R. (2018). *Automatic fault detection using cause and effect rules for in-vehicle networks.* Paper presented at the 4th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2018, March 16, 2018 - March 18, 2018, Funchal, Madeira, Portugal (Vol. 2018-March, pp. 537-544).

Kothamasu, R., Huang, S. H., & VerDuin, W. H. (2006). System health monitoring and prognostics—a review of current paradigms and practices. *The International Journal of Advanced Manufacturing Technology, 28*(9-10), 1012-1024.

Kui, C., Laghrouche, S., & Djerdir, A. (2018). *Proton exchange membrane fuel cell degradation and remaining useful life prediction based on artificial neural network.* Paper presented at the 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA), 14-17 Oct. 2018, Piscataway, NJ, USA (pp. 407-411).doi:10.1109/ICRERA.2018.8567023

Kumar, A., Shankar, R., & Thakur, L. S. (2018). A big data driven sustainable manufacturing framework for condition-based maintenance prediction. *Journal of Computational Science, 27* 428-439. doi:10.1016/j.jocs.2017.06.006

Kumar, D., & Klefsjo, B. (1994). Proportional hazards model: a review. *Reliability Engineering and System Safety, 44*(2), 177-188. doi:10.1016/0951-8320(94)90010-8

Kumar, S., Goyal, D., Dang, R. K., Dhami, S. S., & Pabla, B. S. (2018). Condition based maintenance of bearings and gears for fault detection - a review. *Materials Today: Proceedings, 5*(2), 6128-6137. doi:10.1016/j.matpr.2017.12.219

Kwag, S., Oh, J., & Lee, J.-M. (2018). Application of Bayesian statistics to seismic probabilistic safety assessment for research reactor. *Nuclear Engineering and Design, 328* 166-181. doi:10.1016/j.nucengdes.2018.01.022

Lai, C. T. A., Jiang, W., & Jackson, P. R. (2019). Internet of Things enabling condition-based maintenance in elevators service. doi:10.1108/JQME-06-2018-0049

Lapp, S. A., & Powers, G. J. (1977). Computer-aided synthesis of fault-trees. *IEEE Transactions on Reliability, 26*(1), 2-13.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of machine learning research, 6*(Nov), 1783-1816.

Lee, W.-S., Grosh, D. L., Tillman, F. A., & Lie, C. H. (1985). Fault Tree Analysis, Methods, and Applications ℘ A Review. *IEEE transactions on reliability, 34*(3), 194-203.

Leimeister, M., & Kolios, A. (2018). A review of reliability-based methods for risk analysis and their application in the offshore wind industry. *Renewable and Sustainable Energy Reviews, 91* 1065-1076. doi:10.1016/j.rser.2018.04.004

Levy, S. C. (1997). Safety and reliability considerations for lithium batteries. *Journal of Power Sources, 68*(1 pt 1), [d]75-77. doi:10.1016/S0378-7753(96)02622-5

Li, H.-C., Wu, L., Gao, H.-M., & Zhang, G.-J. (2006). Review and prognosis of state of arts on remote welding teleoperation. *Hanjie Xuebao/Transactions of the China Welding Institution, 27*(6), 108-112.

Li, P. c., Zhang, L., Dai, L. c., & Li, X. f. (2017). Study on operator's SA reliability in digital NPPs. Part 2: Data-driven causality model of SA. *Annals of Nuclear Energy, 109* 185-191. doi:10.1016/j.anucene.2017.05.011

Li, R., Li, J.-h., & Su, L.-j. (2005). Application of rough set to build electric power grid fault diagnosis model based on decision tree. *Relay, 33*(18), 1-5.

Liang, Z., & Lee, Y. (2013). Eigen-analysis of nonlinear PCA with polynomial kernels. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 6*(6), 529-544.

Lin, W.-C., & Ghoneim, Y. A. (2016). *Model-based fault diagnosis and prognosis for Electric Power Steering systems.* Paper presented at the 2016 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp. 1-8).

Lin, W.-C., & Ghoneim, Y. A. (2016). *Model-based fault diagnosis and prognosis for Electric Power Steering systems.* Paper presented at the 2016 IEEE International Conference on Prognostics and Health Management, ICPHM 2016, June 20, 2016 - June 22, 2016, Ottawa, ON, Canada.doi:10.1109/ICPHM.2016.7542840

Liu, B., Ding, Z., Wu, J., & Yao, L. (2019). *Ontology-based fault diagnosis: A decade in review.* Paper presented at the 11th International Conference on Computer Modeling and Simulation, ICCMS 2019, January 16, 2019 - January 19, 2019, Melbourne, VIC, Australia (pp. 112-116).doi:10.1145/3307363.3307381

Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing, 108* 33-47. doi:10.1016/j.ymssp.2018.02.016

Liu, S., Zhu, X., Zhang, Y., & Wu, L. (2016). Research on health status assessment of high-speed railway catenaries based on gray clustering and combination assigning method. *Tiedao Xuebao/Journal of the China Railway Society, 38*(7), 57-63. doi:10.3969/j.issn.1001-8360.2016.07.008

Liu, Y., Frederick, D. K., DeCastro, J. A., Litt, J. S., & Chan, W. W. (2012). User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS): Version 2.

Liu, Y., Xu, Z., Li, G., Xia, Y., & Gao, S. (2019). Review on Applications of Artificial Intelligence Driven Data Analysis Technology in Condition Based Maintenance of Power Transformers. *Gaodianya Jishu/High Voltage Engineering, 45*(2), 337-348. doi:10.13336/j.1003-6520.hve.20190130001

Loukopoulos, P., Zolkiewski, G., Bennett, I., Pilidis, P., Fang, D., & Mba, D. (2017). Dealing with missing data as it pertains of e-maintenance. *Journal of Quality in Maintenance Engineering, 23*(3), 260-278. doi:10.1108/JQME-08-2016-0032

Lu, N., Jiang, B., Wang, L., Lu, J., & Chen, X. (2012). A Fault Prognosis Strategy Based on Time-Delayed Digraph Model and Principal Component Analysis. *Mathematical Problems in Engineering, 2012*.

Lucchese, C., Orlando, S., & Perego, R. (2010). *Mining Top-K patterns from binary datasets in presence of noise.* Paper presented at the 10th SIAM International Conference on Data Mining, SDM 2010, April 29, 2010 - May 1, 2010, Columbus, OH, United states (pp. 165-176).

Luo, M., Wang, D., Pham, M., Low, C., Zhang, J., Zhang, D., & Zhao, Y. (2005). *Model-based fault diagnosis/prognosis for wheeled mobile robots: a review.* Paper presented at the 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005. (pp. 6 pp.).

Ma, X., & Li, D. (2017, 26-27 May 2017). *A hybrid fault diagnosis method based on fuzzy signed directed graph and neighborhood rough set.* Paper presented at the 2017 6th Data Driven Control and Learning Systems (DDCLS) (pp. 253-258).doi:10.1109/DDCLS.2017.8068078

Mahdaoui, R., Mouss, L. H., Haboussi, A., Chouhal, O., Haouassi, H., & Maarouk, T. M. (2019). A Temporal Neuro-Fuzzy System for Estimating Remaining Useful Life in Preheater Cement Cyclones. *International Journal of Reliability, Quality and Safety Engineering, 26*(3). doi:10.1142/S0218539319500128

Manling, D., Hanbo, Z., Yiyi, Z., Kuikui, S., Shuai, Y., Xiaokuo, K., . . . Lei, G. (2019). A Novel Maintenance Decision Making Model of Power Transformers Based on Reliability and Economy Assessment. *IEEE Access, 7* 28778-28790. doi:10.1109/ACCESS.2019.2897606

Martinez, S., Lara, P., Saucedo, M., & Fierro, S. (2018). Industry 4.0 framework for management and operations: a review. *Journal of Ambient Intelligence and Humanized Computing, 9*(3), 789-801. doi:10.1007/s12652-017-0533-1

May, R., Csank, J., Litt, J., & Guo, T.-H. (2010). Commercial modular aero-propulsion system simulation 40k (C-MAPSS40k) user's guide. *NASA TM-216831*.

Mayer, G., & Hennings, C. (2008). Determination of the traffic safety of road tunnels by quantitative risk analyses

Bestimmung der Verkehrssicherheit von Stracentunneln Mittels Quantitativer Risikoanalysen. *VDI Berichte*(2019), 3-23.

Md Nor, N., Che Hassan, C. R., & Hussain, M. A. (2018). A review of data-driven fault detection and diagnosis methods: Applications in chemical process systems. doi:10.1515/revce-2017-0069

Meckel, S., & Obermaisser, R. (2018, 10-14 June 2018). *Component-based combination of online-diagnosis methods using diagnostic directed acyclic graphs.* Paper presented at the 2018 7th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-5).doi:10.1109/MECO.2018.8406089

Medjaher, K., Moya, J. Y., & Zerhouni, N. (2009). Failure prognostic by using Dynamic Bayesian Networks. *IFAC Proceedings Volumes, 42*(5), 257-262. doi:https://doi.org/10.3182/20090610-3-IT-4004.00049

Mehrmolaei, S., & Keyvanpour, M. R. (2016). *Time series forecasting using improved ARIMA.* Paper presented at the 2016 Artificial Intelligence and Robotics (IRANOPEN), 9 April 2016, Piscataway, NJ, USA (pp. 92-97).doi:10.1109/RIOS.2016.7529496

Melani, A. H. A., Murad, C. A., Caminada Netto, A., Souza, G. F. M. d., & Nabeta, S. I. (2018). Criticality-based maintenance of a coal-fired power plant. *Energy, 147* 767-781. doi:10.1016/j.energy.2018.01.048

Mesenzhnik, Y. Z., Prut, L. Y., Gnedin, A. A., & Bugrova, L. D. (1990). Investigating preventive and emergency maintenance of insulation in submersible electrical apparatus for oil production in West Siberia. *Elektrichestvo*(10), 84-88.

Mhenni, F., Nguyen, N., & Choley, J.-Y. (2014). *Automatic fault tree generation from SysML system models.* Paper presented at the Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on (pp. 715-720).

Mi, J., Li, Y.-F., Peng, W., & Huang, H.-Z. (2018). Reliability analysis of complex multi-state system with common cause failure based on evidential networks. *Reliability Engineering & System Safety, 174* 71-81. doi:https://doi.org/10.1016/j.ress.2018.02.021

Miljković, D. (2011). *Fault detection methods: A literature survey.* Paper presented at the 2011 Proceedings of the 34th international convention MIPRO (pp. 750-755).

Ming, L., Heng-Chao, Y., Bin, H., Jun-Hong, Z., & Chee Khiang, P. (2015). A data-driven two-stage maintenance framework for degradation prediction in semiconductor manufacturing industries. *Computers & Industrial Engineering, 85* 414-422. doi:10.1016/j.cie.2015.04.008

Miziula, P., & Navarro, J. (2019). Birnbaum Importance Measure for Reliability Systems With Dependent Components. doi:10.1109/TR.2019.2895400

Mortada, M.-A., Yacout, S., & Lakis, A. (2011). Diagnosis of rotor bearings using logical analysis of data. *Journal of Quality in Maintenance Engineering, 17*(4), 371-397.

Mortada, M.-A., Yacout, S., & Lakis, A. (2014). Fault diagnosis in power transformers using multi-class logical analysis of data. *Journal of Intelligent Manufacturing, 25*(6), 1429-1439.

Mortada, M. A., Yacout, S., & Lakis, A. (2014). Fault diagnosis in power transformers using multi-class logical analysis of data. *Journal of Intelligent Manufacturing, 25*(6), 1429-1439. doi:10.1007/s10845-013-0750-1

Mourtzis, D., Papakostas, N., & Makris, S. (2019). Complexity in industry 4.0 systems and networks. *Complexity, 2019*. doi:10.1155/2019/7817046

Mukherjee, S., & Chakraborty, A. (2007). *Automated fault tree generation: bridging reliability with text mining.* Paper presented at the 2007 Annual Reliability and Maintainability Symposium, 22-25 Jan. 2007, Piscataway, NJ, USA (pp. 6 pp.).

Mutlu, M., Arnold, Z., Franchek, M. A., & Meraz, J. (2017). *Qualitative fault tree analysis of blowout preventer control system for real time availability monitoring.* Paper presented at the Offshore Technology Conference, OTC 2017, May 1, 2017 - May 4, 2017, Houston, TX, United states (Vol. 2, pp. 1187-1197).

National Aeronautics and Space Administration NASA. (2015). Turbofan Engine. Retrieved from https://www.grc.nasa.gov/www/k-12/airplane/Animation/turbtyp/etfh.html

Niu, G. (2016). *Data-driven technology for engineering systems health management: Design approach, feature construction, fault diagnosis, prognosis, fusion and decision*: Springer Singapore.

Noh, S., & Shortle, J. (2018). Barrier Analysis of an Aviation Safety Assessment Model. *INCOSE International Symposium, 28*(1), 616-627. doi:10.1002/j.2334-5837.2018.00504.x

Ntalampiras, S. (2018). Fault Diagnosis for Smart Grids in Pragmatic Conditions. *IEEE Transactions on Smart Grid, 9*(3), 1964-1971. doi:10.1109/TSG.2016.2604120

Omidi, A., & Liu, S. (2018). Overview of the Importance of Intelligent Approaches on Machinery Faults Diagnosis and Prediction Based on Prognostic and Health Management/Condition-Based Maintenance. *Journal of Donghua University (English Edition), 35*(3), 270-273.

Papadopoulos, Y., Walker, M., Parker, D., Rude, E., Hamann, R., Uhlig, A., . . . Lien, R. (2011). Engineering failure analysis and design optimisation with HiP-HOPS. *Engineering Failure Analysis, 18*(2), 590-608. doi:10.1016/j.engfailanal.2010.09.025

Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences, 11*(5), 341-356.

Peeters, J. F. W., Basten, R. J. I., & Tinga, T. (2018). Improving failure analysis efficiency by combining FTA and FMEA in a recursive manner. *Reliability Engineering & System Safety, 172* 36-44. doi:https://doi.org/10.1016/j.ress.2017.11.024

Peng, K.-X., Ma, L., & Zhang, K. (2017). Review of Quality-related Fault Detection and Diagnosis Techniques for Complex Industrial Processes. *Zidonghua Xuebao/Acta Automatica Sinica, 43*(3), 349-365. doi:10.16383/j.aas.2017.c160427

Picoco, C., Aldemir, T., Rychkov, V., Alfonsi, A., Mandelli, D., & Rabiti, C. (2017). *Coupling of RAVEN and MAAP5 for the dynamic event tree analysis of nuclear power plants.* Paper presented at the 27th European Safety and Reliability Conference, ESREL 2017, June 18, 2017 - June 22, 2017, Portorož, Slovenia (pp. 2803-2808).doi:10.1201/9781315210469-356

Polotski, V., Kenne, J.-P., & Gharbi, A. (2019). Optimal production and corrective maintenance in a failure-prone manufacturing system under variable demand. doi:10.1007/s10696-019-09337-8

Purdy, G. (2010). ISO 31000:2009—Setting a New Standard for Risk Management. *Risk Analysis, 30*(6), 881-886. doi:10.1111/j.1539-6924.2010.01442.x

Qin, S. J. (2009). Data-driven fault detection and diagnosis for complex industrial processes. *IFAC Proceedings Volumes, 42*(8), 1115-1125.

Ragab, A., El-Koujok, M., Poulin, B., Amazouz, M., & Yacout, S. (2018). Fault diagnosis in industrial chemical processes using interpretable patterns based on Logical Analysis of Data. *Expert Systems with Applications, 95* 368-383.

Ragab, A., El Koujok, M., Ghezzaz, H., Amazouz, M., Ouali, M. S., & Yacout, S. (2019). Deep understanding in industrial processes by complementing human expertise with interpretable patterns of machine learning. *Expert Systems with Applications, 122* 388-405. doi:10.1016/j.eswa.2019.01.011

Ramzali, N., Lavasani, M. R. M., & Ghodousi, J. (2015). Safety barriers analysis of offshore drilling system by employing Fuzzy event tree analysis. *Safety Science, 78* 49-59. doi:10.1016/j.ssci.2015.04.004

Razavi, S. A., Najafabadi, T. A., & Mahmoodian, A. (2018). *Remaining Useful Life Estimation Using ANFIS Algorithm: A Data-Driven Approcah for Prognostics.* Paper presented at the 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018, October 26, 2018 - October 28, 2018, Chongqing, China (pp. 522-526).doi:10.1109/PHM-Chongqing.2018.00095

Rengasamy, D., Morvan, H. P., & Figueredo, G. P. (2018). *Deep Learning Approaches to Aircraft Maintenance, Repair and Overhaul: A Review.* Paper presented at the 21st IEEE International Conference on Intelligent Transportation Systems, ITSC 2018, November 4, 2018 - November 7, 2018, Maui, HI, United states (Vol. 2018-November, pp. 150-156).doi:10.1109/ITSC.2018.8569502

Rigatos, G., Zervos, N., Serpanos, D., Siadimas, V., Siano, P., & Abbaszadeh, M. (2019). Fault diagnosis of gas-turbine power units with the derivative-free nonlinear Kalman Filter. *Electric Power Systems Research, 174* 105810.

Riyadi, E. H. (2014). *A review for identification of initiating events in event tree development process on nuclear power plants.* Paper presented at the 4th International Conference on Advances in Nuclear Science and Engineering (ICANSE 2013), 16-19 Sept. 2013, USA (Vol. 1615, pp. 152-155).doi:10.1063/1.4895877

Rui, L., & Lie, X. (2018). Review of DC fault protection for HVDC grids. *Wiley Interdisciplinary Reviews: Energy and Environment, 7*(2), e278 (213 pp.). doi:10.1002/wene.278

Ryoo, H. S., & Jang, I.-Y. (2009). Milp approach to pattern generation in logical analysis of data. *Discrete Applied Mathematics, 157*(4), 749-761.

Saha, B., & Goebel, K. (2007). Battery data set. *NASA AMES prognostics data repository*.

Saponara, S., Ciarpi, G., & Fanucci, L. (2018). *Failure analysis of plastic packages for low-power ICs.* Paper presented at the Applications in Electronics Pervading Industry, Environment

and Society, APPLEPIES 2016, 15-16 Sept. 2016, Cham, Switzerland (pp. 160-166).doi:10.1007/978-3-319-55071-8_21

Schwabacher, M. (2005). A survey of data-driven prognostics. In *Infotech@ Aerospace* (pp. 7002).

Schwabacher, M., & Goebel, K. (2007). *A survey of artificial intelligence for prognostics.* Paper presented at the Artificial Intelligence for Prognostics - Papers from the AAAI Fall Symposium, November 9, 2007 - November 11, 2007, Arlington, VA, United states (Vol. FS-07-02, pp. 107-114).

Shili, X., Dong, H., & Xiaoli, L. (2018). *A Generalized Predictive Framework for Data Driven Prognostics and Diagnostics using Machine Logs.* Paper presented at the TENCON 2018 - 2018 IEEE Region 10 Conference, 28-31 Oct. 2018, Piscataway, NJ, USA (pp. 0695-0700).doi:10.1109/TENCON.2018.8650152

Sihombing, F., & Torbol, M. (2018). Parallel fault tree analysis for accurate reliability of complex systems. *Structural Safety, 72* 41-53.

Sikorska, J., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical systems and signal processing, 25*(5), 1803-1836.

Socolar, J. E. S. (2006). Nonlinear Dynamical Systems. In T. S. Deisboeck & J. Y. Kresh (Eds.), *Complex Systems Science in Biomedicine* (pp. 115-140). Boston, MA: Springer US.

Srivastava, P., & Sinha, S. (2012). *Fault Tree Analysis: A Review.* Paper presented at the World Congress on Engineering (WCE 2012), 4-6 July 2012, Hong Kong, China (Vol. vol.2, pp. 810-815).

Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., & Railsback, J. (2002). Fault tree handbook with aerospace applications.

Stamatelatos., M., & Dezfuli., H. (2011). Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners Retrieved from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120001369.pdf

Su, Y., Jing, B., Huang, Y.-F., Tang, W., Wei, F., & Qiang, X.-q. (2015). Correlation analysis method between environment and failure based on fuzzy causality diagram and rough set of multiple decision classes. *Instrument Technique and Sensor*(8), 100-103.

Suhir, E. (2015). Analytical bathtub curve with application to electron device reliability. *Journal of Materials Science: Materials in Electronics, 26*(9), 6633-6638. doi:10.1007/s10854-015-3263-1

Sule, I., Khan, F., Butt, S., & Yang, M. (2018). Kick control reliability analysis of managed pressure drilling operation. *Journal of Loss Prevention in the Process Industries, 52* 7-20. doi:10.1016/j.jlp.2018.01.007

Sun, B., Wang, J., He, Z., Qin, Y., Wang, D., & Zhou, H. (2019). Fault Detection for Closed-Loop Control Systems Based on Parity Space Transformation. *IEEE Access*.

Syfert, M. B. a. M. (2002). DAMADICS RTN Information Web site. Retrieved from http://diag.mchtr.pw.edu.pl/damadics/

Taguchi, G., & Chowdhury, S. (2004). The Second Industrial Revolution and Information Technology. In *Taguchi's Quality Engineering Handbook* (pp. 1-24).

Tahan, M., Tsoutsanis, E., Muhammad, M., & Abdul Karim, Z. A. (2017). Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: a review. *Applied Energy, 198* 122-144. doi:10.1016/j.apenergy.2017.04.048

Tao, Z., & Alves, T. M. (2019). Impacts of data sampling on the interpretation of normal fault propagation and segment linkage. *Tectonophysics, 762* 79-96. doi:10.1016/j.tecto.2019.03.013

Telukdarie, A., Ndlovu, L., & Medoh, C. (2018). *A review on effective maintenance strategies and management for optimizing equipment systems.* Paper presented at the 39th International Annual Conference of the American Society for Engineering Management: Bridging the Gap Between Engineering and Business, ASEM 2018, October 17, 2018 - October 20, 2018, Coeur d&#65533;Alene, ID, United states (pp. 272-281).

Thangavel, K., & Pethalakshmi, A. (2009). Dimensionality reduction based on rough set theory: A review. *Applied Soft Computing, 9*(1), 1-12.

Thapaliya, A., Jeong, D., & Kwon, G. (2018). *Failure analysis in safety critical systems using failure state machine.* Paper presented at the International Conference on Computer Science and its Applications, CSA 2017, December 18, 2017 - December 20, 2017, Taichung, Taiwan (Vol. 474, pp. 540-545).doi:10.1007/978-981-10-7605-3_89

Torngren, M., & Sellgren, U. (2018). Complexity Challenges in Development of Cyber-Physical Systems. In *Principles of Modeling - Essays Dedicated to Edward A. Lee on the Occasion of His 60th Birthday. Lecture Notes in Computer Science (LNCS 10760)* (pp. 478-503). Cham, Switzerland: Springer International Publishing.

Tu, J. L., Duan, R. X., & Dong, D. C. (2010). *An approach for construction of fault trees using rough set and fuzzy clustering algorithm.* Paper presented at the Applied Mechanics and Materials (Vol. 34, pp. 186-191).

Vania, A., Pennacchi, P., & Chatterton, S. (2013). *Fault diagnosis and prognosis in rotating machines carried out by means of model-based methods: A case study.* Paper presented at the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2013, August 4, 2013 - August 7, 2013, Portland, OR, United states (Vol. 8, pp. Computers and Information in Engineering Division; Design Engineering Division).doi:10.1115/DETC2013-12933

Velmurugan, R., & Dhingra, T. (2015). Maintenance strategy selection and its impact in maintenance function: a conceptual framework. *International Journal of Operations & Production Management, 35*(12), 1622-1661.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis. I. Quantitative model-based methods. *Computers & Chemical Engineering, 27*(3), 293-311. doi:10.1016/S0098-1354(02)00160-6

Vera do Carmo, C., Lopes, L. F. D., & Souza, A. M. (2004). Comparative study of the performance of the CuSum and EWMA control charts. *Computers & Industrial Engineering, 46*(4), 707-724.

Vesely, W. E., Goldberg, F. F., Roberts, N. H., & Haasl, D. F. (1981). *Fault tree handbook*. Nuclear Regulatory Commission Washington DC.

Waghen, K., & Ouali, M.-S. (2019). Interpretable Logic Tree Analysis: A Data-Driven Fault Tree Methodology for Causality Analysis. *Expert Systems with Applications*. doi:https://doi.org/10.1016/j.eswa.2019.06.042

Wang, J., Zhang, W., Shi, Y., Duan, S., & Liu, J. (2018). Industrial Big Data Analytics: Challenges, Methodologies, and Applications. *arXiv preprint arXiv:1807.01016*.

Wang, K. S. (2014). *Key techniques in intelligent predictive maintenance (IPdM) - A framework of intelligent faults diagnosis and prognosis system (IFDaPS).* Paper presented at the 4th International Workshop of Advanced Manufacturing and Automation, IWAMA 2014, October 27, 2014 - October 28, 2014, Shanghai, China (Vol. 1039, pp. 490-505).doi:10.4028/www.scientific.net/AMR.1039.490

Wang, L., Qian, Y., Li, Y., & Liu, Y. (2017). *Research on CBM information system architecture based on multi-dimensional operation and maintenance data.* Paper presented at the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp. 167-172).

Wasiewicz, P. (2001). Specification of sugar production processes connected to the actuators intended to use for benchmark definition. *for internal use of DAMADICS participants, Institute of Automatic Control and Robotics, Chodkiewicza, 8* 02-525.

Wilkins, D. (2002). The Bathtub Curve and Product Failure Behavior Part One-The Bathtub Curve, Infant Mortality and Burn-in. Reliability HotWire, Issue 21.

Willersrud, A. (2015). Model-Based Diagnosis of Drilling Incidents.

Wu, G. G., Yang, Z.-X., Song, L., & Li, G. S. (2016). *Control change cause analysis for engine fault diagnosis.* Paper presented at the 13th IEEE International Conference on Networking, Sensing and Control, ICNSC 2016, April 28, 2016 - April 30, 2016, Mexico City, Mexico (pp. Cinvestav; Consejo Nacional de Ciencia y Tecnologia (CONACYT); IEEE; IEEE Systems, Man, and Cybernetics Society (IEEESMC)).doi:10.1109/ICNSC.2016.7479017

Wu, G. G., Yang, Z. X., Li, G. S., & Song, L. (2017). Control change cause analysis-based fault diagnostic approach. *Journal of Disaster Research, 12*(6), 1182-1191. doi:10.20965/jdr.2017.p1182

Wu, Q., Ding, K., & Huang, B. (2018). Approach for fault prognosis using recurrent neural network. 1-13. doi:10.1007/s10845-018-1428-5

Wurzel, G., & Hasbroucq, S. (2015). *Review of fault signature data collection methods for the validation of Condition-Based Maintenance applications.* Paper presented at the Airworthiness, CBM and HUMS Specialists' Meeting 2015, February 9, 2015 - February 11, 2015, Huntsville, AL, United states (pp. 1-11).

Xiang, J., Chen, X., He, Z., Li, B., & He, Y. (2005). *Quantitative detection of crack in structures based on finite element method of B-spline wavelet on the interval.* Paper presented at the 9th World Multi-Conference on Systemics, Cybernetics and Informatics, WMSCI 2005, July 10, 2005 - July 13, 2005, Orlando, FL, United states (Vol. 6, pp. 103-108).

Xiao-Ping, B., & Wei-Hua, S. (2017). Research on safety risk of dangerous chemicals road transportation based on dynamic fault tree and Bayesian network hybrid method.

International Journal of Engineering, Transactions B: Applications, 30(8), 1144-1151. doi:10.5829/ije.2017.30.08b.05

Xiaochuan, L., Fang, D., Bennett, I., & Mba, D. (2018). Canonical variate analysis, probability approach and support vector regression for fault identification and failure time prediction. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 34*(6), 3771-3783. doi:10.3233/JIFS-169550

Xu, Y., Sun, Y., Wan, J., Liu, X., & Song, Z. (2017). Industrial big data for fault diagnosis: Taxonomy, review, and applications. *IEEE Access, 5* 17368-17380.

Yang, G., & Yu, S. (2019). *Synthesized fault diagnosis method reasoned from rough set-neural network and evidence theory* (Vol. 31).doi:10.1002/cpe.4944

Yazdi, M. (2019). A review paper to examine the validity of Bayesian network to build rational consensus in subjective probabilistic failure analysis. *International Journal of Systems Assurance Engineering and Management, 10*(1). doi:10.1007/s13198-018-00757-7

Yi, Y., Yun, B., Chuan, L., & Yan-Ni, Y. (2018). *Application Research of ARIMA Model in Wind Turbine Gearbox Fault Trend Prediction.* Paper presented at the 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), 15-17 Aug. 2018, Piscataway, NJ, USA (pp. 520-526).doi:10.1109/SDPC.2018.8664793

Yin, S., Ding, S. X., & Zhou, D. (2016). Diagnosis and prognosis for complicated industrial systems—Part I. *IEEE Transactions on Industrial Electronics, 63*(4), 2501-2505.

Yiu, T. W., Cheung, S. O., & Lok, C. L. (2015). A fuzzy fault tree framework of construction dispute negotiation failure. *IEEE Transactions on Engineering Management, 62*(2), 171-183. doi:10.1109/TEM.2015.2407369

Yoon, H., & Youn, B. D. (2019). System reliability analysis of piezoelectric vibration energy harvesting considering multiple safety events under physical uncertainty. *Smart Materials and Structures, 28*(2). doi:10.1088/1361-665X/aaf116

Yu, F., Li, G., Chen, H., Guo, Y., Yuan, Y., & Coulton, B. (2018). A VRF charge fault diagnosis method based on expert modification C5.0 decision tree. *International Journal of Refrigeration, 92* 106-112. doi:https://doi.org/10.1016/j.ijrefrig.2018.05.034

Yuan, Z., Zhang, L., & Duan, L. (2018). A novel fusion diagnosis method for rotor system fault based on deep learning and multi-sourced heterogeneous monitoring data. *Measurement Science and Technology, 29*(11). doi:10.1088/1361-6501/aadfb3

Yuchen, J., Shen, Y., & Kaynak, O. (2018). Data-driven monitoring and safety control of industrial cyber-physical systems: basics and beyond. *IEEE Access, 6* 47374-47384. doi:10.1109/ACCESS.2018.2866403

Yuhensky, S., Munadi, R., & Hafiddudin. (2016). *Forecasting formulation model for amount of fault of the CPE segment on broadband network PT. Telkom using ARIMA method.* Paper presented at the 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 13-15 Sept. 2016, Piscataway, NJ, USA (pp. 185-191).doi:10.1109/ICCEREC.2016.7814964

Yunkai, W., Bin, J., Ningyun, L., & Yang, Z. (2015). Bayesian Network Based Fault Prognosis via Bond Graph Modeling of High-Speed Railway Traction Device. *Mathematical Problems in Engineering* 321872 (321811 pp.). doi:10.1155/2015/321872

Zampino, E. J., & Packard, M. H. (2005). *Ideas and possible approaches for Ultra-Reliability.* Paper presented at the Annual Reliability and Maintainability Symposium, 2005 Proceedings: The International Symposium on Product Quality and Integrity, January 24, 2005 - January 27, 2005, Alexandria, VA, United states (pp. 116-120).

Zangenehmadar, Z., & Moselhi, O. (2016). Assessment of Remaining Useful Life of Pipelines Using Different Artificial Neural Networks Models. *Journal of Performance of Constructed Facilities, 30*(5), 04016032 (04016037 pp.). doi:10.1061/(ASCE)CF.1943-5509.0000886

Zarei, J., Shasadeghi, M., & Ramezani, A. (2014). Fault prognosis in power transformers using adaptive-network-based fuzzy inference system. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 26*(5), 2577-2590. doi:10.3233/IFS-130929

Zeng, Z., Kang, R., Wen, M., & Zio, E. (2018). Uncertainty theory as a basis for belief reliability. *Information Sciences, 429* 26-36. doi:10.1016/j.ins.2017.10.050

Zhang, Y., Ren, Y., Liu, L., & Wang, Z. (2015). *A method of fault tree generation based on go model.* Paper presented at the Reliability Systems Engineering (ICRSE), 2015 First International Conference on (pp. 1-5).

Zhang, Y., Zhou, H., Qin, S. J., & Chai, T. (2010). Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. *IEEE Transactions on Industrial Informatics, 6*(1), 3-10. doi:10.1109/TII.2009.2033181

Zhengxin, Z., Xiaosheng, S., Changhua, H., & Yaguo, L. (2018). Degradation data analysis and remaining useful life estimation: a review on Wiener-process-based methods. *European Journal of Operational Research, 271*(3), 775-796. doi:10.1016/j.ejor.2018.02.033

Zhi-jun, P., & Yan, Z. (2015). Transmission maintenance scheduling strategy considering potential fault risk balance. *International Transactions on Electrical Energy Systems, 25*(12), 3523-3537. doi:10.1002/etep.2049

Zhicai, Z., Dongfeng, L., & Xinfa, S. (2014, 24-27 Aug. 2014). *Research on combination of data-driven and probability-based prognostics techniques for equipments.* Paper presented at the 2014 Prognostics and System Health Management Conference (PHM-2014 Hunan) (pp. 323-326).doi:10.1109/PHM.2014.6988187

Zhong, G., Dong, L., & Ye, O. (2018). *Fault Diagnosis Method for Shearer Equipment of PCA-BP_Adaboost.* Paper presented at the 2018 11th International Symposium on Computational Intelligence and Design (ISCID) (Vol. 2, pp. 128-131).

Zolghadri, A., Henry, D., Cieslak, J., Efimov, D., & Goupil, P. (2014). Review and basic concepts. In (pp. 5-27): Springer International Publishing.

APPENDIX A DISCOVERED PATTERNS

Normal	
Pattern	*Pattern description*
P1	F ≤ 0.228
P2	F > 0.228 AND T1≤ 0.219
P3	CV > 0.334 AND CV ≤ 0.539 AND T1 ≤ 0.607
P4	CV > 0.460 AND T1 ≤ 0.607
P5	CV > 0.386 AND T1 ≤ 0.607
P6	T1 ≤ 0.607
P7	CV > 0.302 AND T1 ≤ 0.607
P8	CV ≤ 0.523 AND T1 ≤ 0.607
P9	CV > 0.492 AND T1 ≤ 0.607
P10	CV > 0.302 AND CV ≤ 0.539 AND T1 ≤ 0.607
P11	CV ≤ 0.539 AND T1 ≤ 0.607

Fault	
Pattern	*Pattern description*
P12	T1 > 0.607
P13	CV ≤ 0.492 AND T1 > 0.607
P14	CV ≤ 0.476 AND CV ≤ 0.554 AND T1 > 0.607
P15	CV ≤ 0.492 AND CV ≤ 0.554 AND T1 > 0.607
P16	CV ≤ 0.492 AND CV ≤ 0.523 AND CV ≤ 0.476 AND CV ≤ 0.507 AND T1 > 0.607
P17	CV ≤ 0.415 AND CV ≤ 0.372 AND CV ≤ 0.554 AND CV ≤ 0.400 AND CV ≤ 0.430
P18	CV > 0.507 AND CV > 0.250 AND CV > 0.346 AND T1 > 0.607
P19	CV ≤ 0.492 AND CV > 0.250 AND T1 > 0.607
P20	CV ≤ 0.460 AND T1 > 0.607
P21	CV ≤ 0.400 AND CV ≤ 0.554 AND T1 > 0.607
P22	CV > 0.539 AND CV > 0.359 AND CV > 0.430 AND T1 > 0.607
P23	CV ≤ 0.492 AND CV ≤ 0.507 AND CV ≤ 0.523 AND CV ≤ 0.539 AND CV ≤ 0.476
P24	CV ≤ 0.445 AND CV ≤ 0.476 AND CV ≤ 0.554 AND CV ≤ 0.523 AND T1 > 0.607
P25	CV ≤ 0.492 AND CV ≤ 0.539 AND CV > 0.250 AND CV ≤ 0.476 AND CV ≤ 0.445
P26	CV ≤ 0.430 AND T1 > 0.607
P27	CV ≤ 0.415 AND T1 > 0.607
P28	CV ≤ 0.460 AND CV ≤ 0.554 AND CV ≤ 0.539 AND T1 > 0.607
P29	CV ≤ 0.492 AND CV ≤ 0.523 AND T1 > 0.607
P30	CV ≤ 0.492 AND CV ≤ 0.476 AND CV > 0.256 AND T1 > 0.607
P31	CV ≤ 0.346 AND T1 > 0.607
P32	CV > 0.507848815 AND T1 > 0.6073358625

P33	CV > 0.5545089005 AND T1 > 0.6073358625
P34	CV ≤ 0.460 AND CV > 0.253 AND CV > 0.250 AND CV > 0.260 AND CV > 0.293885 AND CV ≤ 0.539 AND T1 > 0.607
P35	CV > 0.523 AND T1 > 0.607
P36	CV ≤ 0.492 AND CV ≤ 0.554 AND CV ≤ 0.507 AND CV ≤ 0.539 AND CV > 0.264 AND CV > 0.253 AND CV > 0.250
P37	CV ≤ 0.492 AND CV ≤ 0.507 AND CV ≤ 0.460 AND CV ≤ 0.445 AND CV ≤ 0.554 AND CV ≤ 0.476
P38	CV ≤ 0.400 AND CV ≤ 0.415 AND CV ≤ 0.445 AND T1 > 0.607
P39	CV ≤ 0.415 AND CV ≤ 0.507 AND CV ≤ 0.554 AND T1 > 0.607
P40	CV ≤ 0.492 AND CV > 0.256 AND T1 > 0.607