ABSTRACT

Title of dissertation: Systematic Integration of PHM and PRA (SIPPRA) for Risk and Reliability Analysis of Complex Engineering Systems

Ramin Moradi,
Doctor of Philosophy, 2021

Dissertation directed by: Associate Professor Katrina M. Groth,
Department of Mechanical Engineering

Complex Engineering Systems (CES) such as power plants, process plants, and manufacturing plants have numerous, interrelated, and heterogeneous subsystems with different characteristics and risk and reliability analysis requirements. With the advancements in sensing and computing technology, abundant monitoring data is being collected. This is a rich source of information for more accurate assessment and management of these systems. The current risk and reliability analysis approaches and practices are inadequate in incorporating various sources of information, providing a system-level perspective, and performing a dynamic assessment of the operation condition and operation risk of CES.

In this dissertation, this challenge is addressed by integrating techniques and models from two of the major subfields of reliability engineering: Probabilistic Risk Assessment (PRA) and Prognostics and Health Management (PHM). PRA is very effective at modeling complex hardware systems, and approaches have been designed to incorporate the risks introduced by humans, software, organizational, and other contributors into quantitative risk assessments. However, PRA has largely been used as a static technology mainly used for regulation. On the other hand, PHM has developed powerful new algorithms for understanding and predicting mechanical and electrical device health to support maintenance. Yet, PHM lacks the system-level perspective, relies heavily on operation data, and its outcomes are not risk-informed.

I propose a novel framework at the intersection of PHM and PRA which provides a forward-looking, model- and data-driven analysis paradigm for assessing and predicting the operation risk and condition of CES. I operationalize this framework by developing two mathematical architectures and applying them to real-world systems. The first architecture is focused on enabling online system-level condition monitoring. The second architecture improves upon the first and realizes the objectives of using various sources of information and monitoring operation condition together with operational risk.

Systematic Integration of PHM and PRA (SIPPRA) for Risk and Reliability Analysis of Complex Engineering Systems

by

Ramin Moradi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Dr. Katrina Groth, Chair/Advisor
Dr. Mohammad Modarres
Dr. Enrique Lopez Droguett
Dr. Michelle Bensi
Dr. Shapour Azarm
Dr. Greg Baecher, Dean's Representative

Dedication

To

My beloved parents, Reza and Maryam.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I'd like to thank my advisor, professor Katrina Groth for giving me an invaluable opportunity to work on challenging and interesting projects over the past three and a half years. She taught me how be an independent researcher and develop ideas on my own, how to be organized and plan ahead, and how to professionally present my work. She has always made herself available for help for both academic and non-academic matters. It has been a pleasure to work with and learn from such an extraordinary individual. I was lucky to have Professor Groth as my advisor.

I'd like to thank Professor Enrique Lopez Droguett for collaborating with us in this research, providing real-world datasets, and his valuable feedback on this research. Also, I want to express my sincere gratitude to Professor Mohammad Modarres who has been a great inspiration and a wonderful teacher for me. I would also like to thank my dissertation committee members Professors Azarm, Baecher, and Bensi for reviewing this dissertation and providing valuable feedback to improve this study.

My research colleagues at the Systems Risk and Reliability Analysis (SyRRA) laboratory have enriched my graduate life in many ways and deserve a special mention. My dear friend Andres Ruiz-Tagle with whom I have always enjoyed working and discussing a wide range of topics. Roohollah Heidary, Austin Lewis, and Vincent Paglioni who have always been kind and helpful. Having such colleagues was a great privilege. Also, my friends at the Center for Risk and Reliability (CRR), Sergio Cofre-Martel and Fouad Karimian with whom I have had wonderful collaboration experiences.

I owe my deepest thanks to my family - my parents and my sister who have always supported me and my decisions and guided me to be a better version of myself. Words cannot express the gratitude I owe them.

Last but not least, thanks to my lovely girlfriend with whom my life has improved in so many ways, and whose mere presence has empowered me to endure the hurdles of the past years.

It is impossible to remember all, and I apologize to those I've inadvertently left out.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AUC	Area Under the Curve
BN	Bayesian Network
BNN	Bayesian Neural Network
CES	Complex Engineering Systems
CM	Confusion Matrix
CNN	Convolutional Neural Network
CPT	Conditional Probability Table
DBN	Dynamic Bayesian Network
DL	Deep Learning
DNN	Deep Neural Networks
DOOBN	Dynamic Object Oriented Bayesian Network
DRA	Dynamic Risk Assessment
ET	Event Tree
FORM	First Order Reliability Method
FT	Fault Tree
GRU	Gated Recurrent Unit
HM-GMM	Hidden Markov Gaussian Mixture Model
HMM	Hidden Markov Model
LPRA	Living Probabilistic Risk Assessment
LR	Logistic Regression
LSTM	Long Short Term Memory
NN	Neural Network
NPP	Nuclear Power Plant
OOBN	Object Oriented Bayesian Network
PF	Particle Filter
PHM	Prognostics and Health Management
PoF	Probability of Failure
PRA	Probabilistic Risk Assessment
QRA	Quantitative Risk Assessment
ReLU	Rectifier Linear Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RUL	Remaining Useful Life
SIPPRA	Systematic Integration of PHM and PRA
S-RUL	System-Remaining Useful Life
SVM	Support Vector Machine
TL	Transfer Learning

Chapter 1: Introduction

1.1 Background and Motivation

The core of U.S. economy, security, and quality of life depends on complex engineering systems (CES) that range from power plants, energy systems, and pipelines to aircraft, defense, and transportation systems, shown in Figure 1.1. These systems are dynamic, involve complicated physics, consist of many interconnected and interdependent machine and human elements, and operate at multiple levels and multiple scales. CES failures can cause catastrophic consequences such as loss of human lives, financial losses, or even technology abandonment and a change in the direction of an industry.

Traditionally, CES have been the focus of risk assessment studies (e.g., Quantitative Risk Assessment (QRA) and Probabilistic Risk Assessment (PRA)) [1–3].

Figure 1.1: From left to right, thermal power plant, a refinery, control room of a process plant. All considered complex engineering systems.

While performing QRA or PRA is a legal requirement in several industrial sectors for safety critical systems such nuclear power plants (NPP), they are performed mainly during the design phase. For this reason, they only describe a static risk picture of the system [4].

On the other hand, as modern industry moves towards the vision of a data-driven "Industry 4.0" there are new opportunities for reliability engineers to improve the state-of-the-art in risk and reliability analysis of CES [5]. Advanced digitalization, large number of low-cost sensors, and new high-speed processors offer new promises for data-driven insights. These advances provide an opportunity to better understand the complex system's behavior based on a vast array of data which offers detailed insight into their safety and reliability, improved logistics, reduced costs, and more efficient use of resources. However, harnessing this large, diverse data presents significant challenges, including computational challenges (e.g., data storage, processing), and larger challenges about how to use it to provide meaningful support for decision making about CES.

Fortunately, another sub-field of reliability engineering, Prognostics and Health Management (PHM), has excelled in developing data-based algorithms for systems condition monitoring, diagnostics, and prognostics [6–9]. These algorithms are able to handle large-scale multi-dimensional operation data and generate useful features for detecting and predicting faults in the systems. Although, they lack the comprehensive and system-level perspective and are mainly focused on isolated and relatively simple systems such as ball bearings, pumps, pipelines, valves, and electronic components or devices like LEDs and batteries [10–12].

The importance of CES along with the abundance of operation monitoring data demands for a new generation of risk and reliability analysis algorithms. These algorithms should be dynamic and continuously assess the systems' state, provide multi-level analysis (i.e., system-level and component-level) for faster and more efficient decision-making, enable the analysis of diverse/heterogeneous elements and fusion of the results, be applicable to different systems, and be easily updateable as new data and knowledge becomes available.

Risk assessment (PRA and Quantitative Risk Assessment (QRA)) and PHM seem to have complementary characteristics that if integrated, can provide an answer to this novel challenge in reliability engineering. Thus, in this research, we explore the potential of integrating PRA and PHM algorithms and techniques with focus on generating system-level insight, dynamic assessment or monitoring abilities, and integration of all available sources of information. This research would be the first step towards the development of a comprehensive risk and reliability analysis framework for CES, or as we have named it, systematic integration of PHM and PRA (SIPPRA).

1.2 Research Objectives

Followings are the objectives of this research:

Objective 1 (O1): Develop and establish a unifying conceptual and mathematical framework for connecting PHM data and algorithms to PRA models and decisions in the context of complex engineering systems.

Objective 2 (O2): Develop a condition monitoring mathematical architecture for complicated systems that is capable of performing dynamic condition monitoring and providing multi-level (system-level and subsystem-level) measures of operation condition. This architecture should take the live operation data as input and provide operation condition predictions for subsystems as well as the system as a whole.

Objective 3 (O3): Develop an enhanced mathematical architecture that in addition to the target features of Objective 2, is capable of monitoring the complex engineering systems operation condition and risk simultaneously. This algorithm should be able to integrate all available sources of information, enable diagnosis and prognosis, and model interdependencies among subsystems.

To clarify the scope of this dissertation, it should be mentioned that the terms "complex" and "complicated" are used throughout this manuscript for different purposes. The objectives of this dissertation are defined to contribute to the state of the art in risk and reliability analysis of complex systems. However, performing a case study on a full complex system (i.e., one with all of the the characteristics discussed in Section 2.3) goes beyond the scope of this dissertation. Yet to prove the applicability of the proposed approaches, implementations are needed and are performed on the most complex systems that we have managed to gain access to their operational data - we refer to these as "complicated" systems. Therefore, in Chapters 2 and 3, "complex" or CES is used to summarize the reviewed literature and elaborate on the conceptual framework respectively. On the other hand, in Chapters 4 and 5, "complicated" is used to refer to case study systems that have most, but not all of the characteristics of a complex system. We believe that the approaches therein are suitable for complex systems and acknowledge that demonstrating this is a topic for subsequent research.

1.3 Overview of Dissertation

1.3.1 Chapter 1 Summary

Chapter 1 provides a background for this dissertation and elaborates on the motivation for doing this research. Further, the primary objectives of this dissertation are discussed. These objectives would be the basis for Chapters 3, 4, and 5. Finally, a brief summary on each chapter is provided.

1.3.2 Chapter 2 Summary

In Chapter 2, since this research is at the intersection of Probabilistic Risk Assessment (PRA) and Prognostics and Health Management (PHM), the state-of-the-art in PRA and PHM is reviewed and current research trends are discussed. The reviewed literature in these sections elaborates on existing gaps and research needs in both PHM and PRA and motivates the proposed approach of this dissertation. Also, to obtain a better understanding of the characteristics of Complex Engineering Systems (CES), a comprehensive list of features that would contribute to an engineering system's complexity is discussed.

1.3.3 Chapter 3 Summary

To address the risk and reliability analysis requirements of modern Complex Engineering Systems (CES), as well as data complexity which is an outcome of recent advances in sensing and computing technologies. In this chapter the systematic integration of PRA and PHM is proposed. Prognostics and Health Management (PHM) has developed powerful new algorithms for understanding and predicting the mechanical and electrical devices' health. For complex systems, the techniques of Probabilistic Risk Assessment (PRA), which provide a system-level perspective, have become increasingly dynamic. Both PHM and PRA bring unique advantages and limitations. We first justify this integration idea by overviewing their advantages and disadvantages. Second, unified definitions for both PRA and PHM are provided as there are several variations in the literature. Finally, PHM and PRA are systematically deconstructed and reassembled to introduce a more forward-looking, model- and data- driven framework that enables assessment and prediction of operation risk and condition of CES.

1.3.4 Chapter 4 Summary

As the first step towards realization of SIPPRA, a mathematical architecture is developed for system condition or health monitoring. This framework is able to perform an end-to-end condition monitoring, meaning that the streaming monitoring data is given to the architecture as the input and system-level and subsystem-level health assessment is computed as the output. This architecture integrates Fault

Tree as the system modeling method and deep learning as the subsystems condition monitoring method. The applicability of this architecture is tested by implementing it on a real-world mining stone crusher system. A number of different deep learning models are trained using operation and maintenance data for subsystem-level health assessment. The Fault Tree then fuses the continuous subsystem-level assessment results to provide system-level insight.

1.3.5 Chapter 5 Summary

In this Chapter, to realize the SIPPRA framework and the modeling paradigm introduced in it, we present a novel mathematical architecture for operation condition and risk monitoring of CES. In this architecture, Bayesian Network (BN) is used to model the system, subsystems' relations, scenarios leading to adverse events, and to fuse subsystem-level information. Further, Bayesian Deep Learning (DL) models are trained to provide a probabilistic measure for condition monitoring of subsystems and their outputs are integrated into the root nodes of the designed BN. This integration enables addressing both the data complexity and systems complexity in a single architecture and provides system-level insight. This architecture also has the capacity to incorporate human inputs and qualitative information. We demonstrate the effectiveness of our proposed approach by performing a case study on a real-world Vapor Recovery Unit at an offshore oil and gas production platform.

1.3.6 Chapter 6 Summary

This chapter concludes the dissertation by summarizing the main conclusions discussed in former chapters, discussing the contributions of this research to the risk and reliability engineering field, mentioning the publications, presentations, and awards, and finally a discussion on the future research directions that can potentially improve SIPPRA and the proposed mathematical architectures.

Chapter 2: Literature Review

This research is defined at the intersection of PRA and PHM and concepts and techniques used in these two major reliability engineering subfields are integrated to build a novel framework for condition and risk monitoring of CES. Therefore, in this chapter the state of the art in PRA and PHM is discussed to establish a solid basis for this research and how it is contributing to the field. Also, to further illuminate why the current PHM and PRA practices are individually inadequate for analyzing the CES, the characteristics of CES are discussed in Section 2.3.

2.1 Probabilistic Risk Assessment State of the Art

Probabilistic Risk Assessment (PRA) is a comprehensive and systematic methodology that can be used to assess the risks associated with complex engineering systems. Risk in PRA is characterized by the consequences and the likelihood of occurrence of the possible adverse events. The total risk would then be calculated by multiplication of these two quantities.

To enable a more accurate and robust risk quantification, there are two main research trends in the field of risk assessment, more specifically PRA, which will be discussed in this section.

1. Application of advanced simulation for accidents and adverse scenarios identification and analysis.

2. Incorporation of data and dynamic algorithms to generate time-dependent risk measures. In other words, performing Dynamic Risk Assessment (DRA).

2.1.1 Simulation

Accidents are extreme states of the systems which rarely happen. Accidents often cause severe consequences which makes experimentation infeasible and costly. Therefore, to enable risk assessment, one must first identify and characterize hazardous initiating events and scenarios. Doing so is a challenging task as a large number of variables and conditions need to be considered and tested using simulation. Simulation can serve two main purposes in risk assessment:

- Identify high-risk conditions which can lead to critical consequences: In a simulation-based accident exploration, various initial system configurations, designs, and operating conditions are used as inputs to the simulator. This would result in running a number of different simulations in which the end state of the system is computed and recorded as the output of the simulator. Ultimately, the safety requirements and threshold are compared against the recorded outputs to identify the system configurations that lead to critical consequences or system states (a.k.a "Critical Regions") [13]. The identified critical regions may be already known or be discovered after performing the simulation.

- Estimate the probability of occurrence of rare critical scenarios: For this purpose, Monte Carlo (MC) methods are widely used to perform stochastic discrete event simulation across different industries [14–17].

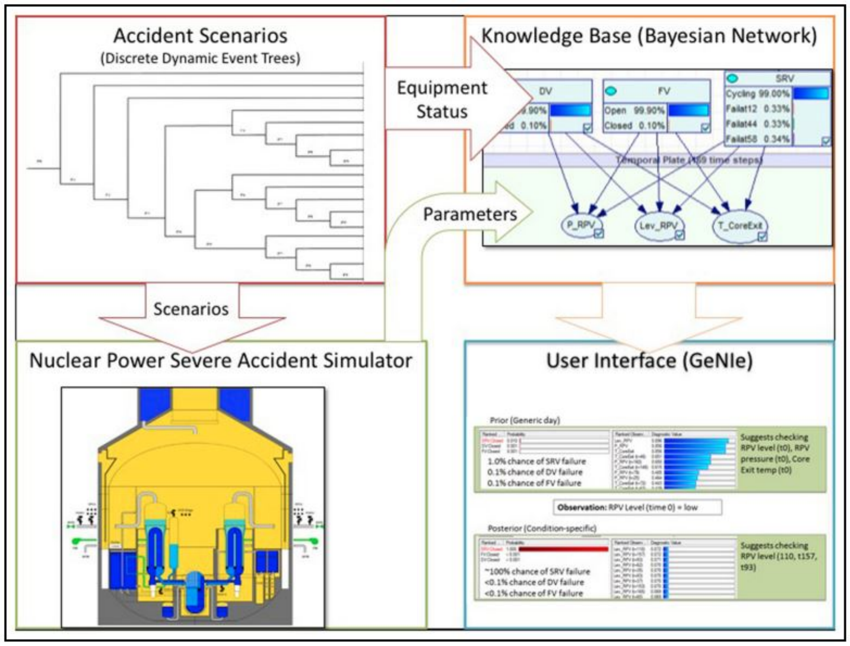

Figure 2.1: An illustration of how simulation helps with risk assessment and knowledge generation for complex systems such as NPPs [18].

Simulation, specially for complex systems, is a challenging task due to the following reasons:

- Computational expense: due to sometimes substantial number of input variables, the exploration space becomes humongous. Also, for some systems, running the simulation even for one configuration can be computationally demanding. Therefore, identification of these configurations becomes harder in the large exploration space of the system [16, 19, 20].

- System's evolution and change in time: as explained in Section 2.3, systems continuously undergo changes for various reasons. Consideration of all the mentioned changes in a simulator has proved to be very difficult [21–24].

- High dimension of inputs and outputs: apart from increasing the computational requirements to find the critical configurations, high dimension of inputs (and outputs) also results in difficult understanding and interpretation of results which demands specialized visualization [25–27].

To address the mentioned challenges, one major current research direction is to develop new algorithms and frameworks for simulation by adaptive sampling. "Adaptive" in this application can be translated into finding ways and measures to direct the progress of the simulation towards the more uncertain scenarios [28], or more hazardous scenarios which are of interest [29–33]. Also, "Adaptive" can be defined with regards to consideration of time dependent variables and consequences [34, 35].

2.1.2 Dynamic Risk Assessment

A proper risk assessment study must be able to account for the time-dependent changes in the system. A system and its components may change due to degradation, change in operating condition, failures, maintenance and replacement, change in components' configuration, etc. Therefore, a risk assessment framework capable of reflecting these changes is needed. This research need has led to the birth of a relatively new research direction which is referred to as Living PRA(LPRA) or more generally, Dynamic Risk Assessment (DRA) [34, 36, 37].

LPRA is a risk assessment architecture designed for a specific system that allows updates whenever a change occurs in the system [37]. These changes can be the result of system's modification, procedure updates, organizational changes, and/or change in the knowledge and understanding of the system as more experience is gained. To enhance the LPRA, the concept of DRA is defined to capture the changes in the components states in a deteriorating system [38, 39].

Bayes theorem and Bayesian networks are widely used in DRA. In earliest studies on DRA Bayes theorem is used to update the accidents probabilities using near miss data [40]. Further, Bayes theorem is used to update the initiating event probability and Event Tree (ET) is used to calculate the consequences probabilities after each update for various systems [41–43].

Bayes theorem is also used in combination with Bow-Tie (BT) to perform DRA. For example, [44] used the Bayes theorem to update the probability of failure in initiating events and safety barriers and used the BT to calculate the updated risk values. A similar approach is used for other applications such as drilling operations [45] and sub-sea pipelines [46].

Bayesian Networks (BN) have become popular in developing DRA frameworks. Specially after Khakzad et al. established how to map a BT into a BN (shown in figure 2.2) [47, 48]. However, a BN offers more flexibility in modeling rather than just being a replacement for the BT. Another way to utilize the BN for DRA is to use its more advanced variation, Dynamic Bayesian Networks (DBN). For example, DBN is used to assess the performance of fire protection systems during a Domino effect [49], a flood control operation for a multi-reservoir system is simulated and the risks are estimated over time using DBN [50], on-demand failure of Nuclear Power Plants (NPP) is assessed using a DBN [51], and the fatigue failure of a subsea wellhead during its operation is modeled by a DBN by [52].

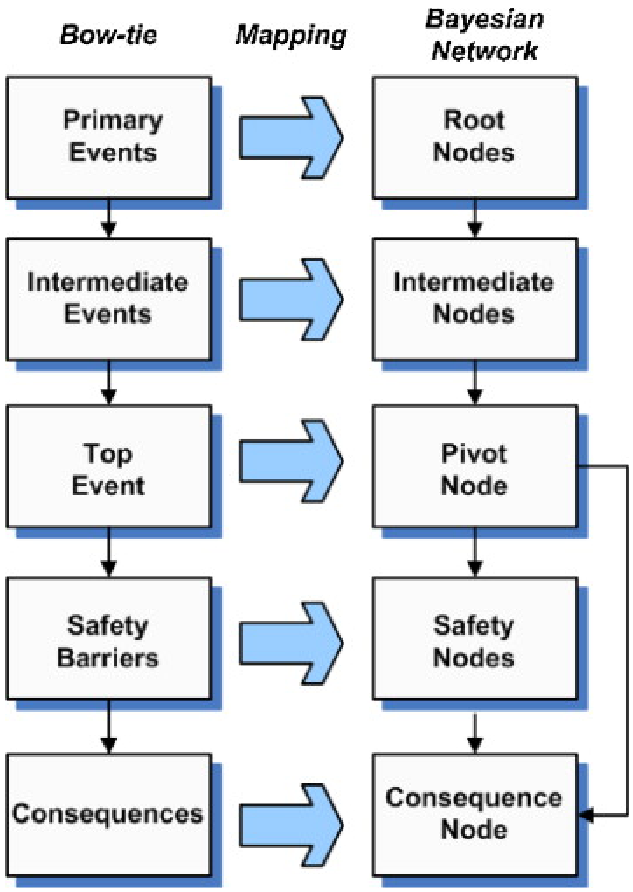

Figure 2.2: The proposed steps to transform a BT into a BN for risk assessment purposes [48].

The mentioned studies and other available DRA studies, use data such as number of accidents and near misses in similar systems, also known as accident precursors. However, this data limits the scope of the DRA analysis to only to the times that an accident is already initiated and is in progress which in many occasions is hazardous. Also, accident precursor data from similar systems may not accurately represent the specific system that is being studied and cause errors in estimations and predictions.

Perhaps the richest source of information is the condition monitoring data collected from the system. Since one can detect the accident precursors by analyzing condition monitoring data. Degradation is a phenomena that ultimately leads to failure and that each failure has potential to be an initiating event for a sequence of adverse events. condition monitoring data provides ample information to continuously asses the reliability of components and safety barriers to provide accurate real-time risk assessment. To take advantage of this opportunity, very few studies have started using condition monitoring data in their DRA. For example, Particle Filter (PF) is used for health estimation of a lithium ion battery using condition monitoring data in [53]. The battery is a safety barrier in an ET. Another example of a health estimation model is presented in [54], in which a Hidden Markov Gaussian Mixture Model (HM-GMM) is trained as a degradation state estimator, and then a Bayesian Network (BN) is used to fuse inspection data with HM-GMM model output. The outcome of this fusion provides updated frequencies for the initiating events or probabilities of safety barriers' success in an ET. The combination of BN and Stochastic Petri Nets (SPN) to capture the dynamic behavior of complex systems was proposed in [55]. This approach is implemented on a pump, a relatively small system. Other algorithms such as self organizing maps [56] and Kalman Filter (KF) [57] are also applied on condition monitoring data to make risk assessments more dynamic.

Application of deep learning in (dynamic) risk assessment of CES is a nuance in this field and very few studies have discussed it. [58] suggests providing all historic data about the system's normal operation, accidents, and expert opinion as features and the instantaneous plant states as labels to a Deep Neural Network (DNN) which calculates a holistic risk index as the output. They demonstrate a new perspective, however there are many challenges such as accessible data (collecting enough data and proper labeling) on the whole system, interpretability of results, and limitations in diagnosis and fault isolation.

2.2 Prognostics and Health Management State of the Art

Prognostics and Health Management (PHM) is a subfield of the reliability engineering that uses the systems operation data to monitor their condition, detect their abnormal operation, diagnose their faults, and prognose their Remaining Useful Life (RUL). To perform the mentioned tasks, three main approaches are used and being advanced that are knowledge-based, data-driven, and hybrid approaches. What follows is an overview of the state-of-the-art in each of these PHM approaches.

2.2.1 Knowledge-Based Approaches

Expert systems are one category of knowledge-based approaches which based on a set of pre-defined rules and conditions, assess the state of the system [59]. These models are less used recently since they require a thorough understanding of the system, yet these models are still widely used in the industry.

Another category is the model-based approaches. In model-based approaches, there is a known mathematical function or physical model available for explaining the behavior of the system/component. This function usually has a number of parameters that need to be tuned based on the failure and condition monitoring data to be applicable to each specific system. Afterwards, the function is used to predict the behavior of the system. Model-based approaches usually perform better than data-driven methods since they can usually be used for relatively longer term RUL predictions. However, developing such models requires a great amount of expert knowledge and/or experimentation, specially for multi-component systems and more complex ones.

The major model-based prognostics approaches in the literature are Paris-Erdogan Law [60] and Forman law [61]. Also, Yu-Harris life equation for bearings' fatigue spall initiation and the Kotzalas-Harris model for failure progression estimation [62] are among the widely used models.

There are numerous studies that are based on the Paris-law equation for different applications. Basically, most of the studies on fatigue RUL prediction are based on this physical law. For example, a model based on the Paris law is proposed to compute the RUL of cracked gears [63]. Cracks in other mechanical components such as bridges [64], gas turbines [65], and aircraft [66] are modeled based on Paris law. It's also the same for other mentioned established physical models which for brevity are not discuss here. Further details can be found in [7].

2.2.2 Data-Driven Approaches

To overcome the limitations of knowledge-based approaches, data-driven models have been increasingly used in PHM applications. Data-driven approaches are mostly Machine Learning (ML) algorithms. To use these algorithms, the condition monitoring data or experimental data is used for training the ML models to perform the target task which can be anomaly detection, fault classification, or RUL prediction.

Based on the type of available training data, the ML problems can be categorized into three groups; supervised, semi-supervised, and unsupervised learning problems. In supervised learning the data is fully understood and different abnormalities and failures are labeled. In contrast, when the data has no labels and the algorithm has to find patterns and distinctions by itself, unsupervised learning algorithms are used. Semi-supervised algorithms are for the cases in which labeled data and the understanding of the system exists, however limited [67]. Data-driven algorithms are explained in this section in two main groups. The first group are the conventional data-driven models which are explained in Section 2.2.2.1, and the second group are the recently developed Deep Learning (DL) models, explained in Section 2.2.2.2.

2.2.2.1 Conventional Data-Driven

In conventional data-driven models, as depicted in Figure 2.3, data should be collected and preprocessed, then feature engineering is required to generate the most

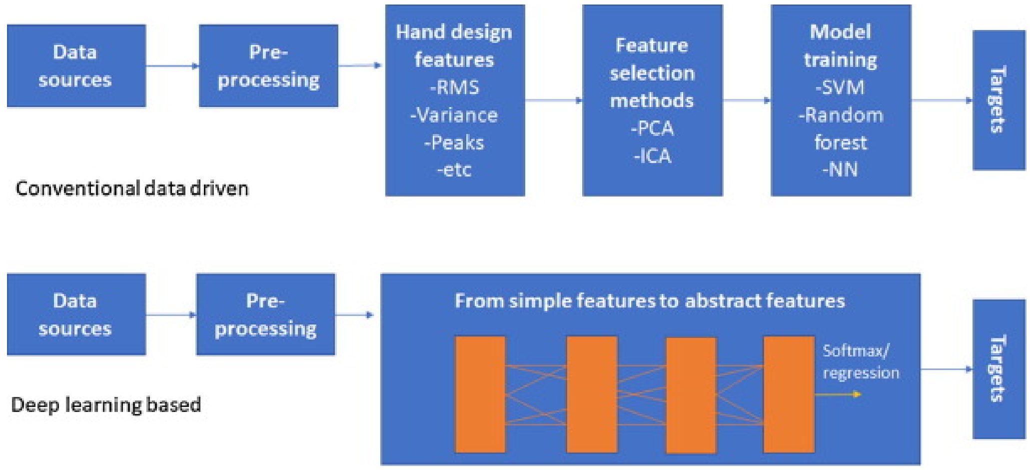

Figure 2.3: Comparison of required steps in conventional data-driven models versus deep learning models [67].

indicative of features for training the models. Feature engineering is perhaps the most challenging task when using these models. Ultimately, the model is trained to perform our desired task which can be classification, regression, or prediction in the context of PHM.

ML algorithms such as Naive Bayes [68], Logistic Regression(LR) [69], Support Vector Machine (SVM) [70], Decision Tree [71], and Random Forest [72] are among the popular supervised classification algorithms that are used for PHM applications.

On the other hand, there are various clustering algorithms [73], self organizing maps [74], and Hidden Markov Models (HMM) [75] that can be used in unsupervised learning problems in PHM. A detailed review of the mentioned methods is provided in [76].

2.2.2.2 Deep Learning (DL)

DL models are basically a re-branding of Neural Networks (NNs), and NNs with more than three hidden layers are called Deep Neural Networks (DNNs) and various architectures of DNNs, result in a large group of models, called DL models. DL models, are widely adopted for PHM applications such as anomaly detection [77], fault diagnostics [78], and prognostics [79]. This is due to their unique ability to handle large amounts of highly non-linear data. Also, DL models process the operation data and automatically generate features for any given task which can be detection, classification, or prediction of patterns in the data. This minimizes the need for domain expertise and extensive feature engineering as demonstrated in Figure 2.3.

To implement any DL algorithm, the three following components are required; (1) data for training and testing the model, (2) an objective function to optimize, and (3) an optimization scheme. Possible variations in these three components would result in generation of numerous distinct DL algorithms and architectures such as Convolutional Neural Networks (CNNs) [80], Recurrent Neural Networks (RNNs) [81], Long Short Term Memory Networks (LSTMs) [82], Generative Adversarial Networks (GANs) [9], Autoencoders [79], etc.

In addition to the three groups of supervised, unsupervised, and semi-supervised learning [83], there is a relatively new type of problem in the DL domain which is Reinforcement Learning (RL) [84]. In RL, the DL model is treated as an agent which tries to perform a task, be evaluated and scored, change its parameters, and perform the task again until it learns how to perform the task satisfactorily based on a set of predefined criteria. RL has been applied for PHM applications in a few studies [85, 86], although it is one of the relatively new research directions in this field and requires further exploration.

Another emerging research direction in DL for PHM applications is Transfer learning (TL). DL models' performance is heavily dependent on the training data and how well that data represents the test data. This issue mandates fine-tuning and even training the models from scratch when there is a slight change in operating conditions or equipment. TL is an approach that can remedy this issue by keeping portions of what is learned from the initial training and transferring them to the new application [87]. Figure 2.4 shows one of the TL techniques used in training DL models. TL has been used for PHM to transfer the knowledge gained from one operating condition to another [88, 89] and to transfer knowledge from simulation and experiment to real-world applications [90]. Similar to RL, TL is at the early adoption stage in PHM and requires further development to become standard practice.

2.2.3 Hybrid Approaches

Other than knowledge-based and Data-driven approaches, the third approach is to use both the knowledge and the data which is called the hybrid approach. These PHM approaches are the most relevant to the topic of this dissertation and therefore are explained in a more detailed manner with focus on system-level PHM.

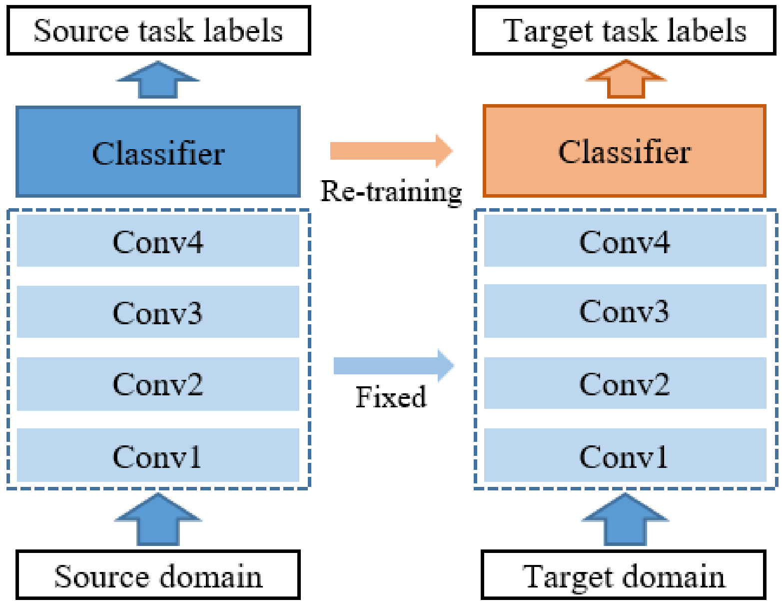

Figure 2.4: The general idea of a TL technique in which the feature extractor layers are frozen and the target domain classifier is re-trained.

There are algorithms that help with the fusion of condition monitoring data and physical models. Kalman Filter (KF) [91] and Particle Filter (PF) [92, 93] are well-known Bayesian algorithms which are widely used for prognostics purposes. These filtering algorithms use a predefined distribution, move this distribution of points forward in time (X axis) according to the physical law, and at the same time update the distribution based on the observations or condition monitoring data on Y axis. This would allow computation of confidence intervals as well as point estimates of their prediction [94].

For systems with multiple components and system-level PHM a limited number of studies exist. The existing ones take advantage of logic-based modeling techniques such as Fault Tree (FT), and Bayesian Networks (BN) along with condition monitoring data. To mention a few, [95] assumed normally distributed RULs for the

9 components of an electrical system and used system failure architecture (FT)) to fuse all the RULs and provide a System-RUL (S-RUL) for maintenance optimization purposes.

[96] used stochastic simulation and inverse first order reliability method (FORM) to compute the S-RUL and concluded that none of these methods are scalable to a large number of components. Further, [97] used manufacturing system operation data in addition to other available contextual information to train a BN representing the relationships among the system variables and produce a system-level measure for failure detection (warning time is considered as an extraction criteria). This BN is then used to extract time bound system-level failure signatures. [98] used a DBN for modeling the system failure logic and dynamically update the probability of system failure according to the observed symptoms. HMM is used for subsystems degradation assessment and DBN is used to achieve a system-level health estimation in[99]. In a similar fashion, [100] selected DBN to perform online system health monitoring and anomaly detection.

In addition to DBN, Object Oriented Bayesian Networks (OOBN) are also used in a number of studies. In OOBNs classes are used to provide a reusable probabilistic model which can be applied to multiple similar objects. For instance, [101] have used OOBNs for real-time fault diagnosis of systems that have many identical subsystems. [102] built system function models based on system structure and engineering knowledge, which are then used along with the causal relations among the subsystems to develop a "system behavior model" using Dynamic Object Oriented Bayesian Network (DOOBN). In an approach similar to BNs, [103] modeled the failure logic of a hydroelectric generator stator using a generic graph theory model. This graph represents root causes, active and inactive physical states and the failure modes of the system. The operation and inspection data is used to map the system physical state to the graph and detect the failure mode that system is in.

2.3 Complex Engineering Systems Characteristics

To better understand why the current PRA and PHM practices cannot meet the requirements of CES risk and reliability analysis, the features and characteristics of CES in contrast with non-complex systems are discussed in this section.

After reviewing the literature [104–111], a consolidated list of characteristics for complex systems is presented in what follows.

2.3.1 Interactivity, Interdependency

Different elements of CES can interact and depend on states of other elements during operation of that system. This interaction can take various forms, as shown in Figure 2.5. There are four recognized types of interaction/dependence in the literature:

1. **Structural:** when components structurally form a part or sub-assembly and have a direct structural connection (or physical connection) or interaction; for example, the gears in a gear box [112].

2. **Functional:** when components or subsystems are connected through control

logic/system, or when the failure or operation of one component causes or impacts the functionality of other components [113].

3. **Economic:** implies that grouping maintenance on components either reduces or raises the overall costs of maintaining the system, compared to individual maintenance [114].

4. **Stochastic:** explains the effect of the deterioration of a component on the lifetime distribution of other components. These dependencies have four main groups [115]: hazard-hazard [116], state-rate [117], degradation-hazard [118], shock-degradation and degradation-shock [119]. For two elements to be stochastically dependent there is often a form connection like via a gas or fluid that is being processed in a system.

These interactions will make it difficult to find root causes of observed errors or find the reason behind anomalous degradation/failures. Emerging behaviors, which are the unforeseen responses and behaviors in the system, can also be a result of the mentioned interactions and often hidden interdependencies.

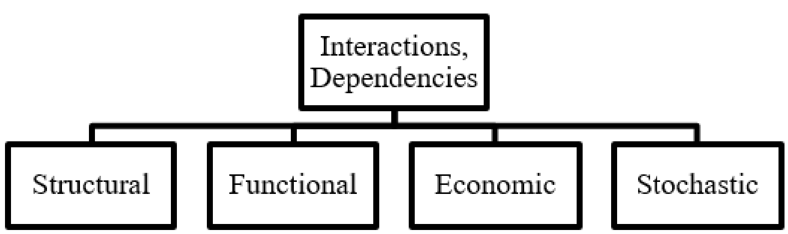

Figure 2.5: Four different types of interactions or dependencies, recognized in the literature.

2.3.2 Diverse/Heterogeneous Elements

There are two main factors that contribute to heterogeneity and diversity in CES, namely, technological difference, and functional differences. Miscellaneous elements work together in CES such as sensors, mechanical components, physical structures, fluids, and materials, software and digital components, sensors, and humans in multiple roles. Technological difference is one contributing factor to heterogeneity (including differences between human-made components and "natural" components). However, even in the case of technologically similar components (e.g., two identical actuators), diversity may exist because of differences in their control logic or relationship to other aspects of the system, both of which would cause functional difference. Note that humans are considered a critical element of CES. Humans play a role in the design, operation, maintenance, and monitoring of these systems and even in automated systems there is a human role. There are three important phases in which humans are playing a role. First, human and machine collaborative decision making and automation, in which the monitoring systems communicates with human operators and the final action will be determined by the operator. Second, in cases involving response to an unusual system condition, a chain of actions needs to be performed. This requires information exchange among decision makers and manipulation of systems. Finally, after a decision is made, correct actions based on the final decision needs to be carried out by humans.

2.3.3 Multi-level and Multi-scale

No matter how complex, CES have a hierarchical structure with multiple levels and they have been successfully modeled in that manner [100, 120, 121]. A basic hierarchical modeling could have the individual components in the bottom of this structure, then the subsystems which are a group of components working together to achieve an objective, larger subsystems, and finally the whole system. Figure 2.6 is a simplified demonstration of the multi-level characteristic of CES (i.e. a cluster of interactive components would form a sub-system, a group of sub-systems forms a larger sub-system which would eventually make the whole CES) Also, to best analyze and capture the behavior of CES, multi-scale (ranging from atomistic to macro scale) design, analysis, simulation, optimization, integration, and data collection is required. Furthermore, various time-scales of the collected data, components life time, prediction horizons, and so on can also make the analysis of CES exponentially harder.

2.3.4 Uncertainty and Ambiguity

Uncertainty and ambiguity can play a vital role in analyzing CES. Uncertainty can exist in our information about the current state of the CES due to inaccurate data collection or imprecise data pre-processing. It can exist because of lack of information about the future states of the system due to variations in operation conditions, environmental conditions, or emerging outcomes of interactions among components of the system. Another important form of uncertainty is the model

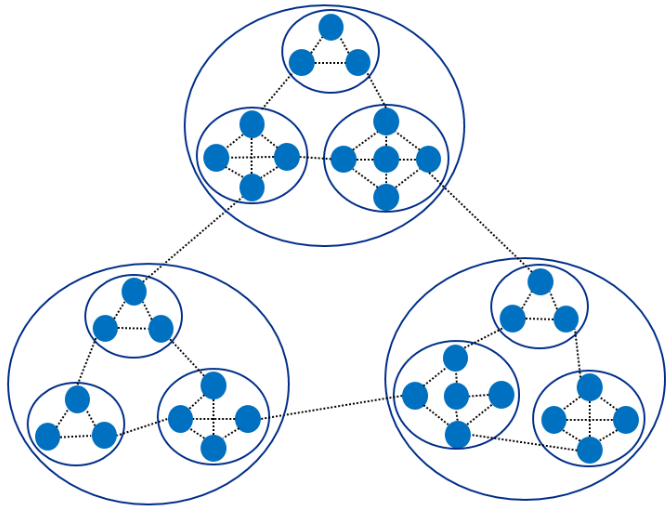

Figure 2.6: Schematic demonstration of how CES operate at various levels. Components are the blue dots and interaction can exist between all subsystems. In some cases the boundaries between elements may overlap or be unclear.

uncertainty. Model uncertainty can present itself in different ways:

- Model form: model discrepancy, missing underlying physics, and model approximations.
- Model parameters: a parameter may be constant but unknown and its uncertainty represented via a probability distribution or it may vary randomly according to some probability distribution.
- Threshold definition: certain thresholds in reliability engineering applications like end of useful life can be fuzzy.
- Model explainability: explainable models make us more confident about the outcome. Thus, being able to intuitively convey the logic behind the results of the model can increase our certainty about the results.

We may have cases in which there is no plausible model, a not-completely-validated single model, competing or complementary models, or composite models that are an integration of sub-models with different degrees of uncertainty [122]. For CES, since each component has a different behavior, a complicated composite model should be used to model the whole system. Therefore, proper uncertainty propagation analysis becomes more challenging. Simulation-based [123] and probabilistic methods [124, 125] can become too expensive to perform and numerous model validations are required. Figure 2.7 summarizes the explained sources of uncertainty.

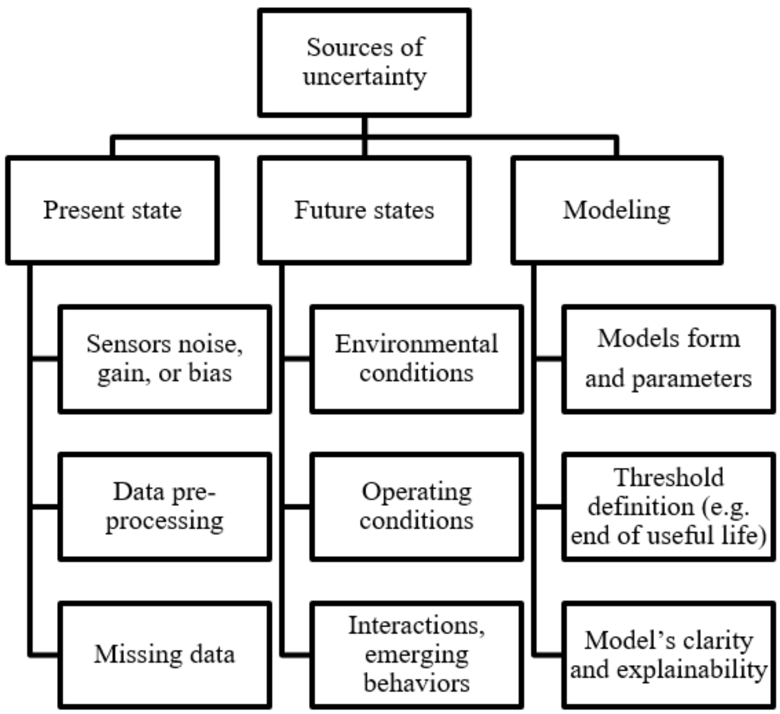

Figure 2.7: Different sources of uncertainty, extended from [126].

2.3.5 Adaptability

The adaptation might occur in the whole or a part of the systems in response to the environmental or operational conditions, or elements may change in response to imposed pressure from the neighboring elements. If humans are considered a part of the system, changes in responsiveness and the type of responses in various conditions can contribute to adaptability of the system as well. Further, CES have various operation modes that they can switch to depending on the required tasks. Furthermore, Internet of Things (IoT) and introduced concepts in industry 4.0, facilitate the application of Machine Learning (ML) and Artificial Intelligence (AI) in controlling the CES. Doing so, would provide us with an intelligent and evolving system which is consequently more difficult to comprehend and analyze.

2.3.6 Unclear systems boundaries

As shown in Figure 2.6, depending on the analysis method and purpose, often it is better to isolate the components and subsystems by a set of designated boundaries which are hard or even impossible to derive due to interactions in the CES. Defining exact boundaries for isolating the whole CES can be a difficult task too. For example, humans and their interactions with a system can increase the complexity of understanding a system considerably since the human behavior is highly variable. Depending on the purpose of an analysis, humans may be viewed as observing systems, independent systems or as system elements. Additionally, the environment may have deep interactions with the system and be the root cause of various errors in the system which would raise the question of whether or not the environment should be considered as a part of the CES.

2.3.7 Non-linearity and Non-equilibrium

Non-linearity is another property of CES that is characterized by approximation, random behavior, and unpredictability. It indicates that the relationship between variables is not static and directly proportional to the input; instead, it is dynamic and variable. Non-linearity can be a result of systems control logic, systems physics, or other complexity features like inter-connectivity and adaptability. On the other hand, non-equilibrium (mostly mentioned in chemical engineering studies [108]), can be inferred as the quality of not being in a constant steady condition (at least for a long time) as the system operates. The main reason is that CES usually possess different operation modes depending on the tasks, environment, load, health state of the system, and so forth.

2.3.8 Emergent Behavior

Another important characteristic that distinguishes the CES from other systems is emergent behaviors. Uncertainty and nonlinearity of CES contributes to having emergent and unforeseen behaviors. Emergent behavior can be defined as the new behavior generated by combination of two or more different subsystems while none of them display that behavior individually [127]. Ross Ashby attributes this characteristic to lack of understanding about the systems and states that if the knowledge is complete, the prediction of the system's behavior can also be complete [128]. However, the complete knowledge becomes more expensive and harder to obtain as the complexity increases.

It worth mentioning that Systems of Systems (SoS) can be confused with CES. SoS is a term that is used to describe a network of heterogeneous systems that interact with one another to provide an overall capability [129, 130]. Although sometimes these two have similar behaviors and their distinction becomes fuzzy, SoS is often associated with operational and managerial Independence of Elements.

Chapter 3: A Systematic Integration of Prognostics and Health Management and Probabilistic Risk Assessment (SIPPRA)

In this chapter, we seek to systematically deconstruct and re-assemble PHM and PRA and develop a new framework which overcomes the limitations of both methods. The new framework will provide a systematic, dynamic approach to risk assessment, condition monitoring and prediction for CES.

3.1 Why Does the Integration of PHM and PRA Make Sense?

In this section, to elaborate on why this integration has the potential to address some of the current most important challenges in reliability engineering, first the advantages and disadvantages of PRA and PHM are discussed, and then the results of a methodical literature search is used to establish the need for better communication and collaboration between the PHM and PRA research communities.

The most recent proposed PHM models are capable of handling high volumes of multi-dimensional data collected by sensors, which nowadays are largely found at the component-level. Recently PHM methodologies have been proposed for relatively more complex systems [131, 132], but there remain gaps for analyzing CES. For example, there is an established need for further development of PHM methodologies that can capture the interactions in a system for an accurate system-level diagnostics and prognostics [7, 133–135]. Furthermore, PHM remains strictly hardware focused, and neglects important elements of complex engineering systems, including human and organizational factors.

On the other hand, traditional PRA is very effective at modeling complex hardware systems, and approaches have been designed to incorporate the risks introduced by human reliability, software failures, and organizational factors that define complex engineering systems. Recent advances in computing have enabled simulation-based dynamic PRA approaches to explore thousands of accident scenarios via multiple approaches [21, 34, 136]. Yet, despite these advances, there remain significant gaps in PRA for analyzing CES. To date, PRA has largely been used as an offline, static technology and mostly focused on retrospective analysis. Only a few studies have attempted to incorporate real-time condition monitoring data within the implementation of dynamic PRA. There remains a need for better analysis of event progressions, the ability to use all available sources of information, and new types of data [36, 53, 55, 137, 138].

Table 3.1 summarizes the main points discussed in the last two paragraphs. Considering what mentioned above, PHM and PRA appear to have complementary characteristics in the context of CES risk and reliability analysis.

Further, to support the need for such an integration, we searched for the keywords presented in Table 3.2 in the titles, abstracts, and keywords of papers in relevant journals (i.e., *Reliability Engineering and Systems Safety*, *Journal of Risk and Reliability*, and *Safety Science*) with emphasis on recent literature (2015-2020).

	PRA	PHM
Uses	Systems engineering and reliability engineering methods	Data preprocessing, feature engineering, machine learning, and deep learning
To	Identify system failure scenarios, estimate the probability of events, and define consequences.	Perform health state assessment, fault diagnostics and prognostics, and Remaining Useful Life (RUL) prediction
Pros	Well-established in high-consequence industries – blends data, models, and expert knowledge & connects with decision makers	Enables online processing of abundant, multi-dimensional operation data & decision support
Cons	Mainly done offline and in the design stage and updated annually (or even less)	Applies to components & isolated systems

Table 3.1: Comparison of PRA and PHM key characteristics.

This search resulted in identification of 267 unique papers. To ensure comprehensiveness beyond these three journals, we also looked at the reference list of the most relevant papers in addition to searching for their titles in Google Scholar to look at similar existing studies. From the resulting set, we kept papers that dealt with relatively complex systems and had one or more of the following elements: a system-level perspective, systems logic modeling, use of different information sources, and dynamic/online/real-time assessment. Finally, twenty papers were determined to be the most relevant studies to the idea of integration of PRA and PHM for CES risk and reliability analysis. The identified 20 papers are discussed in Sections 2.1 and 2.2. They are also used in Table 3.3 to validate the comprehensiveness and novelty of the proposed framework.

No.	Selected search keywords	RESS	Safety Science	JRR
1	Risk AND (prognosis OR prognostics)	8	2	8
2	Risk AND (Real-time OR Dynamic OR Online) AND System	56	44	1
3	(System-level) AND (Reliability Or Risk) AND Complex System	43	25	0
4	("Machine Learning" OR "Deep learning") AND ("logic model"OR "event tree" OR "fault tree" OR "Bayesian network")	43	27	8
5	PHM AND (PRA OR PSA)	2	0	0

Table 3.2: The selected search keywords for finding the most relevant available literature (2015-2020).

3.2 Methodology

To develop the new framework, we first identify and define the main elements of both PRA and PHM. While the main elements of each approach are likely to be well known to the experts within each constituent community, Table 3.2 shows there is a significant lack of overlap between the two fields. This necessitates establishing a consistent set of baseline steps and terminologies. After establishing the basic elements, we interweave the steps of each to create the new framework.

3.2.1 Probabilistic Risk Assessment (PRA)

PRA provides a well-established family of techniques that addresses safety, risk, and reliability of these complex engineering systems. PRA is increasingly embedded in the regulation of nuclear power plants, oil and gas facilities, chemical processes, and infrastructure. Because of the ability to handle uncertainty and complicated trade-offs in complex socio-technical problems, risk-informed approaches are being rapidly introduced within public policy across U.S. federal agencies involved in science, engineering, public health and safety. The use of PRA as a decision support methodology in the nuclear power domain can be traced back to the 1975 WASH-1400 Reactor Safety Study [139].

PRA uses a combination of system and reliability engineering methods, logic-based models, and probabilistic and deterministic methods to deal with complexity and uncertainty. It provides a structured, quantitative method to systematically identify component and system failure sources and scenarios, estimate the probability of events related to failure and mitigation, and determine consequences of undesired events. The results of these steps are then used as a rational, coherent, scientific basis for decision making on risk management measures.

We developed Figure 3.1 to draw together the major elements of PRA, unifying concepts from seminal textbooks in the field [2, 3]. PRA starts with clearly defining the objective and scope of doing the study, and contains six major steps.

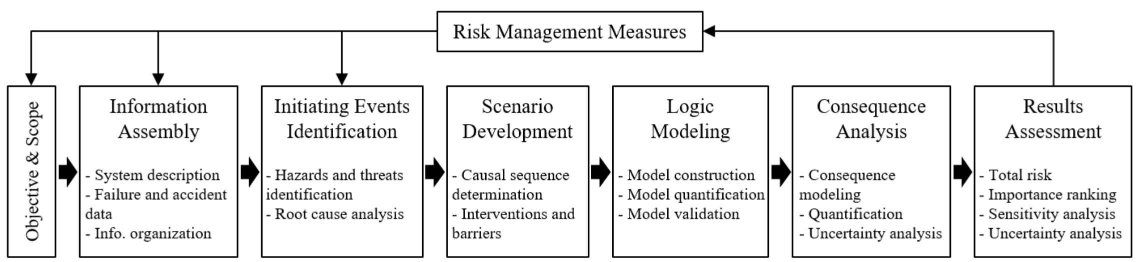

Figure 3.1: A general PRA framework. After defining the objective and scope of the PRA study, there are six major steps that provide risk assessment, which feedback into the risk management of the system. This feedback process continues until the objectives of the analysis are met.

1. Information Assembly: as a first step, information is collected and organized to provide a thorough understanding of the system, its components, and the expected functions and behavior. The typical information includes a variety of qualitative and quantitative information in multiple formats, including detailed system descriptions, layouts, functional descriptions, and available failure and operational data.

2. Initiating Events Identification: serves to identify the potential hazards and threats to the system, that could lead to an undesired event or initiate a sequence of events with an undesired outcome. This includes consideration of the root causes of those events. This step draws on experience, data, and consideration of possible hypothetical situations.

3. Scenario Development: for each initiating event the causal chain leading from initiating event to possible outcomes (consequences) is created. The scenario elements included in this causal sequence are potential success and failure events, hardware, software, controller or human interventions, barriers, and mitigations.

4. Logic Modeling: a set of logic-based models is then constructed for the scenarios. Modeling methods such as Fault Trees (FT), Event Trees (ET), and BNs are used to document known relationships between the scenario elements. These models are quantified with probabilities from the assembled information from component and event databases and operating experience.

5. Consequence Analysis: in this step, the hypothesized consequences (e.g., rupture of a pressure vessel) for each causal scenario are modeled and the potential losses they impose are quantified, typically with a combination of deterministic physical models and probabilistic loss models. An uncertainty analysis is performed on the expected consequences.

6. Results Assessment: the total risk is calculated by integrating the aforemen-

tioned elements. Sensitivity and uncertainty analysis are performed to evaluate the impact of various modeling assumptions on the risk. Depending on the objectives of the study, the results are explored further. Generally, the risk results are compared against a risk tolerability or acceptance thresholds for the system, and the most important scenarios and contributing components are identified, ranked, and major uncertainties are illuminated.

After completing these six steps, there is feedback into the risk management measures for the system, which may change the system, data sources, or other aspects of the analysis. This feedback process continues until the objectives of the analysis are met.

3.2.2 Prognostics and Health Management (PHM)

PHM as a concept has been inspired by medical science practices [135] and has been studied by many researchers from many different engineering fields over the last two decades. PHM is well-established for mechanical components such as gears and bearings, pumps, valves, and for electronic components or devices such as LEDs, batteries, and circuits [140–142]. PHM techniques have been extended to some systems such as engines, fuel cells and wind turbines.

PHM focuses on developing models and algorithms for diagnostics and prognostics (i.e., reliability assessment at the current moment and in the future). PHM systems use real-time and historical operational data to provide decision support for improved performance, reliability, and maintainability.

We developed Figure 3.2 to draw together major concepts from PHM frameworks [7, 12, 133, 143]. In general, PHM studies have a uniform objective: to assess the health state of a component and/or system (the first box in Figure 3.2). PHM typically incorporates four major sequential steps which are data acquisition, diagnostics, prognostics, and health management.

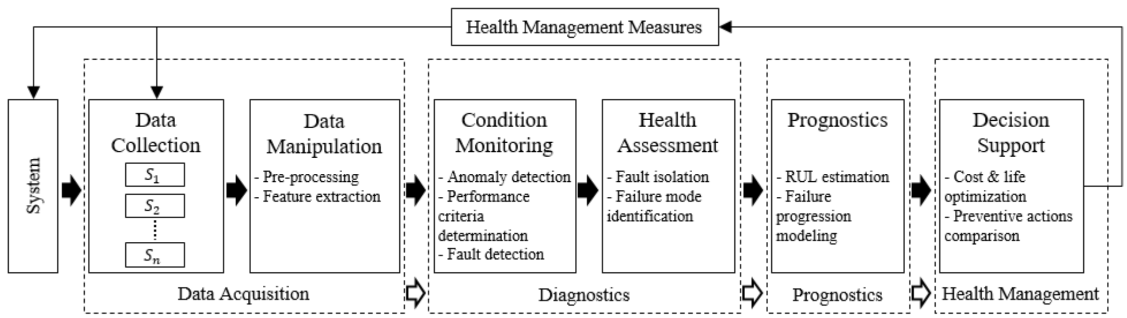

Figure 3.2: A holistic PHM framework. It has four major steps that are indicated with dashed lined: data acquisition, diagnostics, prognostics, and health management). Diagnostics includes sub-steps of condition monitoring and health assessment, and data acquisition incorporates both data collection and manipulation.

1. Data Acquisition: begins with a data collection step, which involves continuously collecting representative information from the system. This typically occurs through a properly designed and maintained network of sensors. Then, the data needs to be manipulated and prepared for use in diagnostics and prognostics. Data pre-processing (i.e., cleaning, normalization, integration) and feature extraction are the necessary tasks of data manipulation. The features in PHM are typically a set of *Health Indexes* (HIs).

2. Diagnostics: includes continuously monitoring the operating condition of the component or system to detect faults or anomalies based on deviation from the expected normal values of performance criteria. Then, if a fault or anomaly

was detected, the health of the system is assessed by identifying the root cause, conducting failure mode and failure mechanism identification, and isolating the faulty or anomalous component(s). In other words, diagnostics is more of a classification problem that prescribes if the health in a particular component or subsystem or the system is degraded.

3. Prognostics: this is the predictive aspect of PHM, a step that predicts the time at which a system or a component will no longer perform as intended and needed. In order to implement prognostics, the most important task is to develop a model for the degradation process of the component or system under study. These models can be physics-based, data-based, or hybrid which is a combination of the two. This model is used to project when the system will meet a specified criteria, such as the time at which it will reach a defined efficiency threshold, failure probability, degradation level, or a minimum tolerable time to failure. The predicted time is called the RUL in the PHM literature. In addition to RUL estimation, computing the probability of failure, estimating relative likelihood of different failure modes in future, and failure progression modeling can be studied in prognostics.

4. Health Management: includes preparing required information to support decision-making. Often in PHM this is intended to provide actionable information on how to optimally extend the RUL of the system while still controlling costs (i.e., to manage the health). This can provide insight for the selection of damage control actions and maintenance policy to extend the RUL of the system

in a way that minimizes maintenance and operational costs while maximizing system availability and capability. Maintenance policies can be either corrective or preventive. Corrective means that the action will be performed after observing a fault or anomaly. In contrast, preventive maintenance is the preferred method (especially for safety-critical systems) since it would intervene early in the degradation process before a failure happens. Preventive maintenance policies can be designed based on reliability, age, and risk of failure in a system. Decision support provides all the mentioned information required for a proper selection of health management measures.

After completing these four steps, there is feedback into the health management measures for the system, which may change the system, data collection, and consequently other aspects of the analysis.

3.3 Proposed Framework

In this section, we pull together the steps of PRA and PHM to construct a comprehensive framework that we call SIPPRA (Systematic Integration of PHM and PRA) throughout this dissertation. We present SIPPRA in two parts. The first part is the construction of the framework, and the second part is the application of the constructed framework. These are shown together in Figure 3.3.

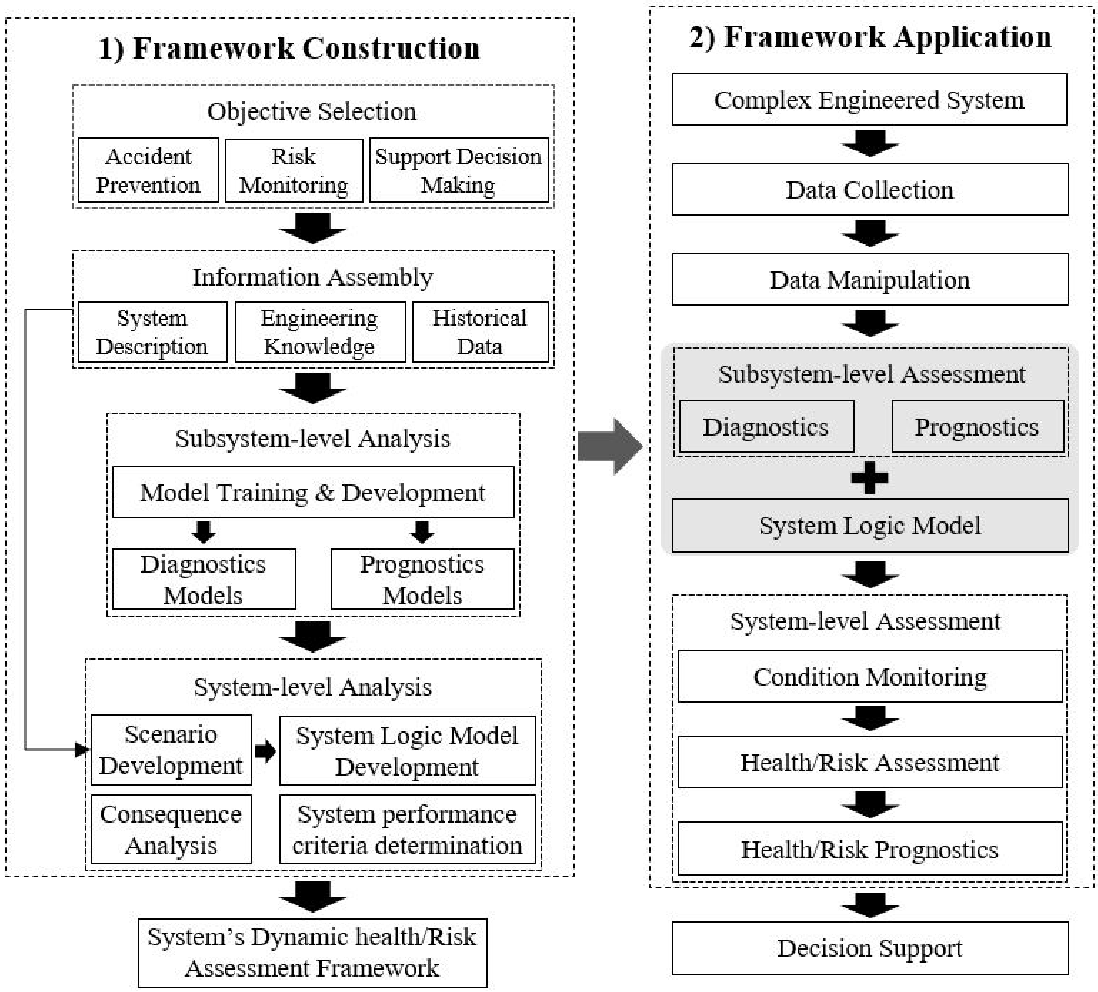

Figure 3.3: The proposed health/risk assessment framework which draws various elements of PHM and PRA. On the left, the steps to construct the framework and on the right, the steps to apply the framework on a complex system are demonstrated. The gray box in framework application conveys that both subsystem-level assessment and system logic model are being used in system-level assessment.

3.3.1 Framework Construction

3.3.1.1 Objective Selection

Similar to PRA, the first step is to determine the general objective of the study. There are a number of possible objectives. Such a framework can be used for system-level health/condition monitoring, determining a specific accident prevention measure, enabling dynamic risk monitoring, and decision making support among others. In addition, owing to inclusion of PHM, the objectives can go beyond PRA or even dynamic PRA; prediction of risk or risk prognostics can be an objective for SIPPRA. Similar to PHM, objectives can also include supporting maintenance decisions and life cycle management.

Considering the objectives of both PRA and PHM, the proposed methodology would be able to answer the following questions:

1. What is the current condition or health state of the system and its subsystems?
2. How much is the operation risk of the system at each time step?
3. How would a fault or failure in one component of the system impact the functionality of other components and the system?
4. When will the system reach a defined risk or health threshold (e.g., end of life)?

3.3.1.2 Information Assembly

The objective of this step is obtaining a thorough understanding of the system, the components, and the relevant data sources similar to the first step of PRA. This step is distinct from current PHM practices which often rely on a large volume of relatively complete data to generate insights. In the context of a CES, obtaining complete, comprehensive, big data would be costly and inefficient, even if techniques existed for collecting and handling that data. Creating a framework that leverages engineering knowledge and multiple sources of partial data, akin to PRA, provides a more tractable approach.

In this step, the desired understanding includes the interdependence among the system, its components and subsystems, the surrounding environment, the relevant data sources, and situational, human, and organization factors. This step requires acquiring information such as system description, functions, data from experiments, material characteristics, performance tests, results from science- and engineering-based simulations at the component and system levels, historic operation data, failure data, inspection, and maintenance reports, and formal expert opinion. These data and information have a variety of modalities, from text-based, to numeric data, to Figures and tables. This step provides a comprehensive set of information which will be used in constructing a system-specific version of the framework.

3.3.1.3 Subsystem-level Analysis

Before describing this step, it is necessary to first discuss some attributes of complex systems which necessitate dividing the analysis into two levels, which we call subsystem-level analysis and system-level analysis. We use the term subsystem to denote the level of decomposition where "typical" PHM data is collected for the system of interest. The subsystem-level analysis is reflective of the model development aspects of current PHM, and the system-level analysis is reflective of the model development aspects of PRA.

First, complex systems can be modeled as a multi-level hierarchical structure with interactive components [100, 120, 121]. We can visualize this as in Figure

2.5, where a cluster of interactive components would form a sub-system, a group of sub-systems forms a larger sub-system which comprise the CES. Second, given the current data collection and monitoring techniques, different types and amounts of data are collected at different levels of system abstraction, and this data collection is happening at different frequencies with different resolutions and qualities. In addition, according to the criticality of the subsystem there may be a need to perform analyses at different levels of fidelity or frequency. a higher quality and more frequent assessment. Thus different analysis techniques are required. Together these characteristics motivate decomposing the CES into subsystems and motivate use of an ensemble of models.

The objective of subsystem-level analysis is to develop (and/or train) the necessary models for each designated subsystem health diagnostics and prognostics. Diagnosis is the step which includes detecting anomalies and faults and identifying the root causes for the relevant subsystem. The subsystem prognostics step includes estimating or predicting the failure progress and its impact on RUL of a subsystem.

As is done in current PHM, these models can be classified as knowledge-based, data-driven, or hybrid models. In knowledge-based models, the degradation process is described by mathematical formulation derived often from the underlying physics of the subsystem degradation. The physical model parameters are determined from experimental data and may be adjusted based on physical characteristics of the subsystem (e.g., material, dimensions, or manufacturing conditions of the subsystem). These parameters determine the behavior of the models and impact the error of the model. The models are employed to diagnose the degradation level of a subsystem and extrapolate the degradation level until it reaches a failure threshold (prognostics).

Data-based models diagnose and prognose directly from monitored data. To achieve this, various classification, regression, and machine learning models are used on proper training data. These models include [144]: 1) Artificial intelligence and machine learning approaches like deep learning [67] and fuzzy logic [145], and 2) Statistical approaches like Gaussian process regression [146], Gamma process [147], Wiener process [148], and Hidden Markov Models [149].

Among data-based models, there are several reasons to choose deep learning models including their ability to handle multi-dimensional data, learn in an unsupervised manner which is a great benefit in cases where the data is not labeled properly [9, 150], and emergence of novel concepts in deep learning such as transfer learning [151] which allows exploring effectiveness of training models on one component and apply them on other similar components with minor adjustments.

Finally, hybrid models integrate merits of different approaches to reduce limitations and achieve more accurate results (for further details please refer to [152, 153]).

This step also involves feature engineering and constructing a set of *Health Indexes* (HIs). Feature engineering is the process of transforming the data into meaningful variables for analysis. Features convey the embedded value or important information in the data. The features are typically a set of HIs (HIs) which can be categorized into two groups: Physical HIs (PHI) and Virtual HIs (VHI) [153]. The selected HIs will be used by the developed models and perform real-time assessment of subsystems health in the framework application stage.

PHIs are related to the physics of failures and generally having a meaningful interpretation, such as the monotonic changes in the time domain features (kurtosis, skewness, crest factor, etc.) which can be extracted from the vibration signals of a degrading bearing [154]. VHIs are HIs that have established predictive value but which do not have clear interpretability (e.g., parameters obtained from machine learning models or those that are a fusion of multiple PHIs or multi-sensor signals).

3.3.1.4 System-level Analysis

The purpose of this step is to provide perspective on how diagnosis and prognosis at the subsystem-level can impact CES functions, and manifest in system-level failures and consequences. This step is informed by four of the main activities from PRA. This step involves system failure scenario development (which includes initiating events identification) and logic modeling for various conditions, wherein the causal chain from initiating events to consequences is established. This includes modeling subsystem success and failure logic as it relates to system-level success and failures. These models by incorporating information such as system architecture. The developed logic model would capture interdependencies among subsystems, and would be quantified using various sources and formats of data. The model would be updatable in the framework application steps. This completed system logic model explains how a fault can propagate throughout the system and can be used for inferring the root causes or the origin of any anomalous behavior.

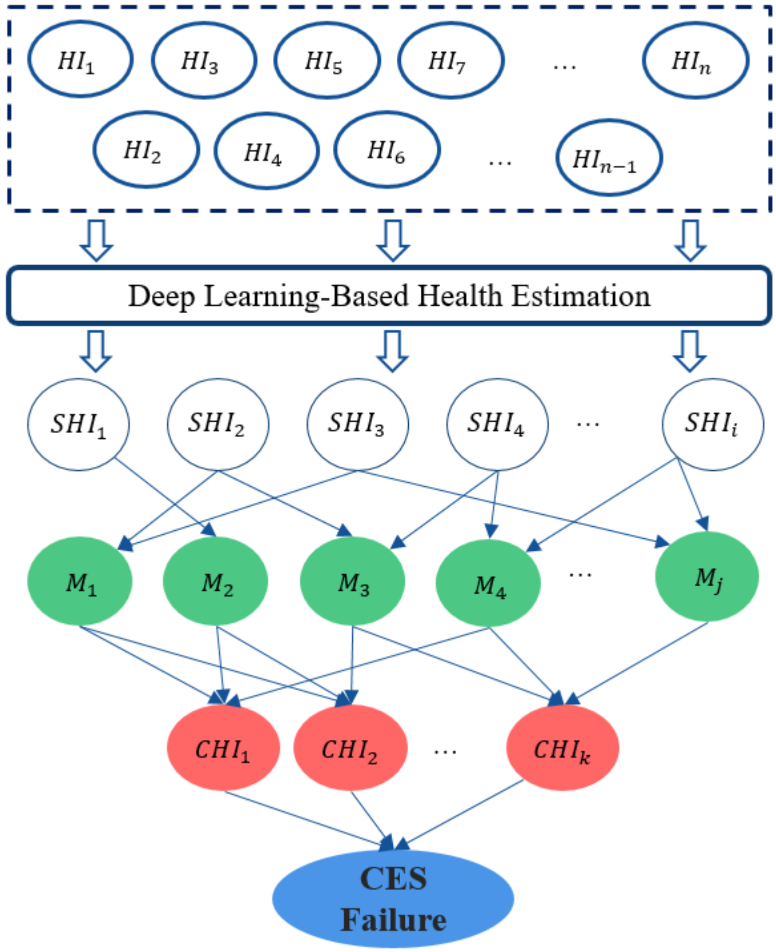

Figure 3.4: Abstract representation of a possible mathematical architecture developed by following SIPPRA. The model encodes relationships between the extracted features or Health Indexes (HI), Subsystem Health Indicators (SHI_i) which are determined by deep learning models, a set of Mediating variables (M), and CES health indexes (CHIs). By taking into account all the CHIs the probability of failure in for the whole CES can be obtained.

Depending on the objective of analysis, logic model would be constructed with FT or ET, or more advanced methods like probabilistic graphical models like Bayesian networks (BNs)[101, 155] and its extensions like DBN [156] and OOBN [157], graph theory, complex networks [158]. In Figure 3.4 we illustrate an example architecture that uses BN to for system's logic modeling, which will allow forward- and backward-propagation of information to enable diagnostic and prognostic reasoning about the system. In this architecture, the extracted features are used by trained deep learning models to estimate a Subsystem Health Indicator (SHI), then the BN implements the CES logic (architecture, function, failure, etc.) by using a set of Mediating variables (M), a number of CES Health Indicators (CHI) would then be produced to ultimately provide us with an overarching CES health level. It may also possible to consider other approaches like parity relations [159], bond graphs, and signed digraphs (SDGs) [160], or Petri nets [161]. An important consideration when developing such models is balancing the level of detail in the models, the size and number of subsystems and data sources incorporated, while keeping the model complexity manageable.

Determination of the system performance criteria is the last task of the system-level analysis. Considering a CES with both PRA and PHM perspectives, the system performance criteria would include PRA measures such as tolerable operational risk threshold, as well as system-level PHM measures including CES efficiency, energy consumption, pollution, products quality if applicable, and rate of occurrence of failure relative to other similar systems which can reveal errors in a CES that may not be discoverable using the bottom-up approach explained earlier and thus are worth considering and monitoring.

Ultimately, a dynamic risk assessment or a risk-informed health assessment framework for the system under study is obtained and is ready for application.

3.3.2 Framework Application

After the framework is constructed, it can be applied to provide decision support. This entails collecting the online operational data, manipulating it, and then passing it into the developed framework, which can then be executed to provide updated, condition-specific instances of the models.

3.3.2.1 Data Collection

In this step, which is drawn directly from PHM, operational data is collected for the subsystems which are being monitored. Data collection in CES usually is done through a network of a large number of different types of sensors distributed across the system as well as other types of data sources. This network is often associated with a distributed data storage and pre-processing/control units. The collected passes through many connections and middleware and therefore it is crucial to guarantee that the data is being reliably transmitted, and received at the destination in a timely manner.

Also, the collected data should be properly stored for future analysis and model updating since as the system operates we would have access to higher amounts of data and therefore higher accuracy levels can be achieved in framework iterations over time. Since we are dealing with industrial big data, the five V's [162] of big data must be considered for proper storage:

- Volume: proper storage capacity
- Variety: proper categorization and structuring of the stored data (human

inputs, operation data, events and failures, maintenance, performance decay scenarios, etc.)

- Velocity: high data storage and access speed
- Value: storing only the most valuable data
- Veracity: making sure the collected data is correct and accurate before storage

3.3.2.2 Data Manipulation

In this step, the data is manipulated to make it ready to be used in the diagnosis and prognosis models. It involves two main tasks which are drawn from PHM: data pre-processing and feature extraction. Data pre-processing involves cleaning, normalization, and integration of streaming operation data before using it for further assessment. This also includes handling missing values and sensor disagreements and data reduction in such a way that enables further computation that maximizes computational efficiency and minimizes possibility of introducing errors into the data analysis. Feature extraction is the real-time process of transforming the pre-processed operation data into *Health Indexes* that are used in the diagnostics and prognostics models.

3.3.2.3 Subsystem-level Assessment

In this step the developed models from Section 3.3.1.3 are used to perform diagnostics and prognostics on subsystems using the streaming operation data. Each subsystem has its own specific trained set of models designed to work with the type of data collected. In some cases, a subsystem have a combined diagnostics and prognostics model, in other cases the subsystem may have two separate models.

Diagnostics models continuously monitor the operation of the subsystems to detect faults/anomalies, isolate the faulty/anomalous component(s), conduct failure mode and failure mechanism identification, and root cause analysis. On the other hand, prognostics models predict the time at which a subsystem or a component will no longer perform as intended and needed. In addition to RUL estimation, computing the probability of failure and relative likelihood of different failure modes are another prognostic tasks at this point.

3.3.2.4 System Logic Model

The model developed in Section 3.3.1.4, which brings together all individual subsystem-level assessments considering all the known interactions and relations among subsystems, is then used to pass the information from subsystems into a system-level indication of system's health or system's operation risks. This combination of the system logic model or models and subsystem-level assessments enables the system-level assessment (this combination is shown with a gray box in Figure 3.3).

3.3.2.5 System-level Assessment

This is the step where the model is used to obtain insights about the health and risk of the system and subsystems. The CES operation and condition is monitored, and health/risk of operation of the CES are assessed and predicted considering all subsystem-level assessments and interactions among subsystems at different levels represented in the logic model. At this point we would know the state of the whole CES by knowing the state of all individual subsystems, have an estimate of when the CES will no longer be operational by fusion of subsystems assessments results. In case of the CES being faulty we would know the possible root causes, severity, and how possibly that fault would propagate throughout the system by forward/backward inference through the logic model. As explained, Figure 3.4 demonstrates a possible architecture for health assessment of a CES.

As far as the system-level prognostics is concerned, following the same architecture of Figure 3.4, one can perform prognostics individually on subsystems and use the logic model as a mean to fuse all these results as shown in Figure 3.5.

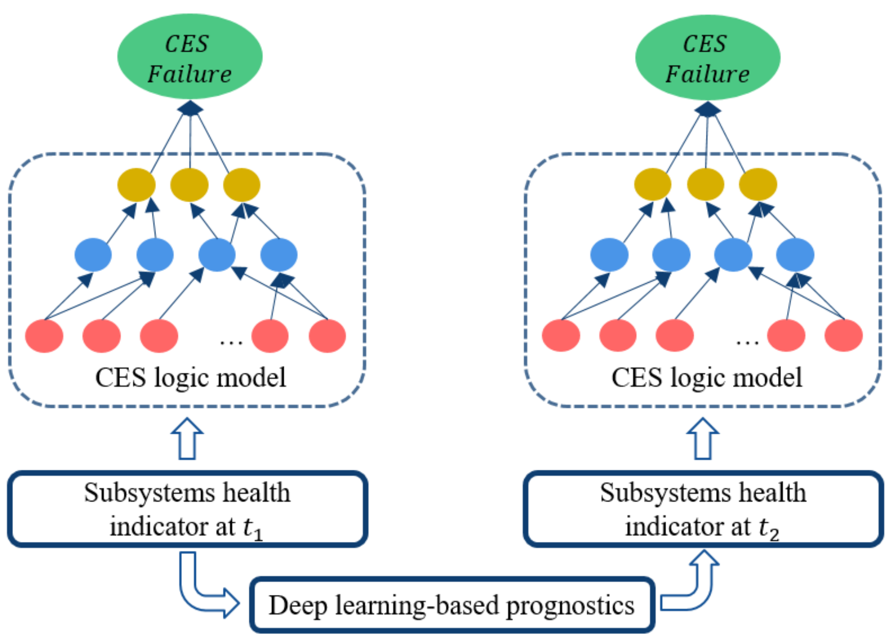

Figure 3.5: Demonstration of system-level prognostics approach using deep learning for subsystems prognostics and logic models for estimating the CES probability of failure in the next time steps.

Ultimately, the outcome of the mentioned assessments will be used as inputs and support for decision making.

3.4 Framework Validation

To verify that our framework contains the necessary elements, and to validate its comprehensiveness and extended perspective, twenty of the most relevant papers discussed in Chapter 2 are mapped onto the main elements of SIPPRA in Table 3.3. As shown, none of the existing studies capture all of the elements considered in SIPPRA. Many studies do not use all the available information sources, with many focusing on either online data or historical data and engineering knowledge, but not both. Model training and development is neglected in several papers. Consequence analysis cannot be found in a majority of the identified studies. In many studies there is not a clear distinction and emphasis on subsystem-level and system-level analysis and assessment does not exist.

3.5 Discussion and Conclusion

In this chapter, we proposed a new health/risk assessment framework that integrates PHM and PRA techniques and concepts. In a sense, we are presenting a hybrid model, one that uses the sum of advantages of both PHM and PRA to provide a forward-looking approach for risk and reliability assessment of complex engineering systems. This work is the first to draw together these two distinct disciplines systematically combining these techniques into a new framework. In

Paper	Objective	Data Collection			Subsystem-level Analysis	Subsystem-level Assessment		System-level Analysis			System-level Assessment	
		Online Operation Data	Historical Database	Engineering Knowledge	Model Training and Development	Diagnostics	Prognostics	Scenario Development	System Logic Model	Consequence Analysis	Health/Risk Assessment	Health/Risk Prognostics
[163]	Safety prognosis	✓	✓	✓	×	✓	✓	✓	✓	×	✓	❍
[95]	Maintenance planning	×	×	×	×	×	❍	×	✓	×	×	❍
[164]	Safety control, Hazard monitoring	✓	×	✓	×	×	×	✓	✓	×	✓	✓
[96]	System-level RUL prediction	×	×	×	×	✓	✓	×	×	×	×	❍
[101]	Real-time Fault diagnosis	✓	✓	✓	✓	✓	×	×	✓	×	✓	×
[97]	Online failure prognosis	✓	✓	✓	✓	✓	✓	×	×	×	✓	×
[165]	Reliability and RUL calculation	✓	×	✓	×	✓	✓	×	×	×	×	❍
[103]	Multi-failure mode Prognosis	×	✓	✓	×	✓	×	×	✓	×	✓	❍
[161]	System safety assessment	×	×	✓	×	×	×	✓	✓	×	✓	×
[100]	Real-time system health monitoring	✓	✓	✓	❍	✓	×	×	✓	×	✓	×
[99]	Estimation & prediction of system functional reliability	✓	×	✓	❍	✓	✓	×	✓	×	✓	❍
[98]	Availability assessment	×	✓	✓	❍	✓	×	×	✓	×	✓	❍
[131]	Fault detection	✓	✓	×	❍	×	×	×	×	×	✓	×
[102]	Fault diagnosis, system RUL prediction	✓	✓	✓	❍	❍	❍	×	✓	×	✓	❍
[166]	DRA	✓	✓	✓	✓	❍	❍	✓	✓	×	❍	❍
[53]	DRA	✓	✓	✓	✓	✓	×	✓	✓	❍	✓	×
[18]	Accident diagnosis	✓	×	✓	✓	✓	×	✓	✓	✓	✓	×
[58]	Deep learning-based risk assessment	✓	✓	✓	✓	×	×	×	×	×	✓	×
[54]	DRA	✓	✓	✓	✓	✓	×	❍	✓	❍	✓	×
[55]	DRA	✓	✓	✓	✓	✓	×	×	✓	×	✓	×

Table 3.3: Mapping the related work against elements of the SIPPRA to demonstrate its comprehensiveness and support the validation of the framework. ✓shows that they have considered that task, ✗means that they have not considered that task, and ❍shows partial consideration. All of these papers are discussed in depth in Chapter 2

doing so, we have laid out the core elements of a more comprehensive approach that is data-informed and model-informed, and scalable to complex systems. We validated the methodology by carefully reviewing the literature and mapping our framework against elements contained in the most related published work.

There are a few limitations that worth discussing. In this chapter, we have assumed that data is available for the CES under study and this data can be real or the result of an accurate simulation. There are existing issues with real data availability and accurate simulation of CES. However, addressing these issues are complementary to this research and this chapter motivates data collection and more accurate simulations.

Also, we proposed the integration of systems logic models with deep learning models for subsystems diagnostics and prognostics. This approach does not fully solve problems such as scalability of some logic models like BNs. However, it would considerably reduce the size of logic models and therefore make them more computationally manageable; allowing us to model larger and more complex systems. Additionally, we have not created a full implementation of the proposed method on a real-world complex system in this chapter.

To conclude, we have introduced a systematic integration of PHM and PRA for the purpose of complex engineering systems risk and reliability assessment and prediction. The product of this integration is both a risk-informed PHM and a data-driven PRA approach for CES. Advantages and contributions of this framework are using all sources of information, the capability of multi-level assessment and prediction, the introduction of deep learning into engineering systems risk assessment and prediction, and holistic view and therefore more convenient decision making. Also, this framework can be re-structured to be applicable on various CES, can be updated whenever there is new information/data available or when the system has minor changes.

Chapter 4: Development of a System Health Monitoring Architecture by Integrating PRA and PHM Techniques

4.1 Introduction

As discussed in Chapter 1 and 2, most of the research conducted in PHM (diagnostics and prognostics) are focused on component-level assessment. Yet one of the main characteristics of CES is that they have multiple, if not numerous, components and subsystems which are interacting with one another. Therefore, to be able to accurately monitor and manage the operation of CES, obtaining a system-level is key. As mentioned in Chapter 2, there are a number of studies on system-level PHM in which methods such as inverse first order reliability method (FORM) [96], Monte Carlo simulation [165], and FT [95], DBN [100], and graph theory [103] are used for systems and components' relations modeling. Sometimes the mentioned models are combined with a Bayesian estimator or a simplistic state estimator for the subsystems. The details of these studies are discussed in Section 2.2.3.

Although, some papers on system-level PHM exist, there remain many challenges to be solved. For example, the mentioned studies are not suitable for online condition monitoring of systems with large amounts of monitoring data. Also, the scalability of the proposed approaches is another remaining issue. PHM approaches rely heavily on the data and none of the mentioned studies incorporates the risks associated with the operation of the system. In this chapter, we propose a mathematical architecture that is suitable for online, multi-level (system-level and subsystem-level) condition monitoring of complex engineering systems. FT is used, as a method widely used in PRA and safety studies, to model the system design and failure logic. Further, to be able to use the abundant available condition monitoring data to perform online state estimation for subsystems, deep learning-based condition monitoring algorithms are trained and integrated with the FT to compute a system-level state estimation at each time step.

This mathematical architecture is our first step towards realization of the SIPPRA framework and enables online and multi-level analysis of CES. In what follows, in Section 4.2, we first present the background and theory of the methods used in chapter. Then in Section 4.3, the steps of the proposed architecture are explained and a case study is performed and discussed in Section 4.4 to exhibit the applicability and usefulness of the proposed architecture. Finally, this chapter is concluded in Section 4.5.

4.2 Preliminaries

4.2.1 Fault Tree (FT)

Fault Tree (FT) is one of the most important methods in risk analysis studies and applications related to economically and safety critical systems. FT consists of two main types of nodes: events and gates. An event is an occurrence within the system, typically the failure of a subsystem down to an individual component. Events can be divided into Basic Events (BE), which occur spontaneously, and intermediate events, which are caused by other events in the tree. The event at the top of the tree, called the Top Event (TE), is the event being analyzed (as depicted in Figure 4.1), when modeling the failure of the (sub)system under consideration. Symbols for other types of events are shown in Figure 4.2. While intermediate events can be useful for documentation, they do not affect the analysis of the FT, and may therefore be omitted. If a FT is too large, triangles are used to transfer events between multiple FTs like a switch mechanism. Finally, sometimes BE are not truly basic and spontaneous, but they are considered so due to insufficient information or lack of importance in the analysis (such an undeveloped event is denoted by a diamond).

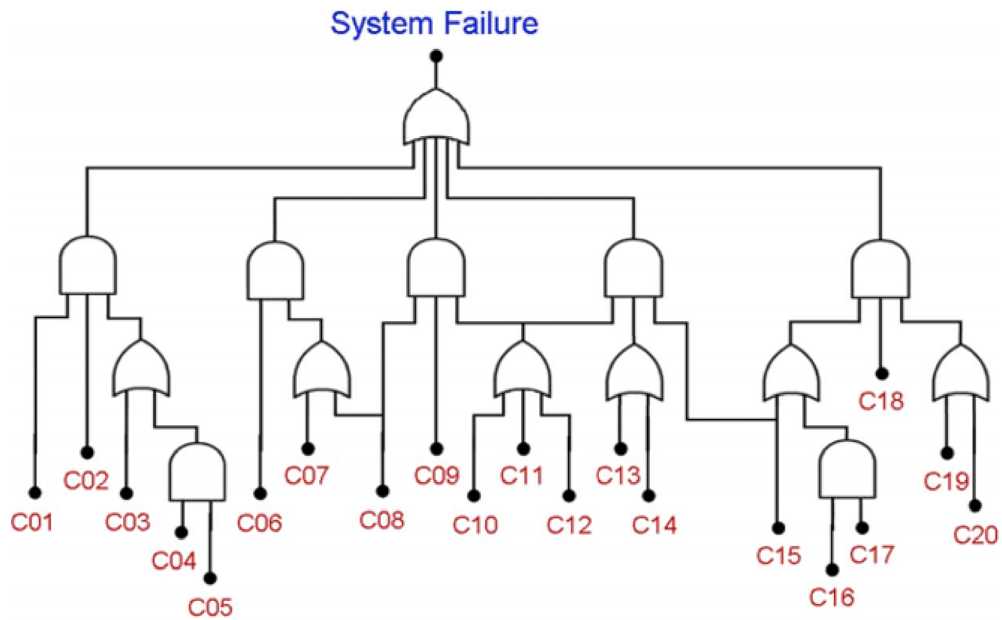

Figure 4.1: A system fault tree representation. Used from Rodrigues et. al. [95]

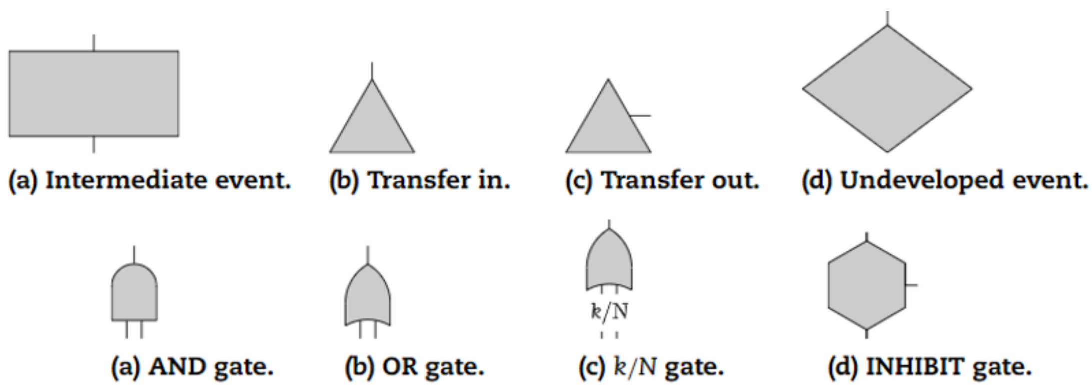

Figure 4.2: Top row: Different types of non-basic events in a fault tree. Bottom row: A number of gate types in a fault tree. Used from Ruijters and Stoelinga [167].

Gates determine how failures propagate through the system, meaning how failures in components can cause a system failure. Each gate has one output and at least one input. A number of gates are shown in Figure 4.2. Several extensions of FT introduce additional gates proper for modeling of repairable systems, non coherent behaviors in which operation of a specific component in a certain condition can cause failure, and more.

Other important topics related to FT are (minimum) cut sets, Dynamic Fault Trees (DFT), FT with temporal requirements, and FT with dependent events (Common Cause Failure (CCF)) which are thoroughly explained by Ruijters and Stoelinga [167].

4.2.2 Deep Learning-Based Condition Monitoring Algorithms Used In This Study

4.2.2.1 Neural Networks

Neural Networks (ANN) (shown in Figure 4.3), are the foundation of deep learning models. Given their hierarchical nature, NN could extract abstract information from a given input data X_0 by stacking multiple layers of nonlinear functions. A layer (h_i) is mathematically described as:

$$h_i = \sigma(W_i^T \cdot X_{i-1} + b_i). \tag{4.1}$$

Where, W_i and b_i are the so-called weights and bias of layer i, respectively. X_{i-1} is the input data to the layer, and σ is known as the activation function, which is a nonlinear function such as the hyperbolic tangent (tanh), rectifier linear unit (ReLU), sigmoid. Each of these nonlinear transformations are known as layers. The number of outputs in each layer (i.e., h_i's dimension) is equal to the number of neurons or hidden units, that the layer has. For a multi-layer neural network (i.e., Deep Neural Network), a layer's outputs h_i correspond to the input data X_{i+1} of the next layer. The output of the whole network (y) is then given by:

$$y = W_l^T \cdot h_l + b_l \tag{4.2}$$

where l corresponds to the number of hidden layers of the network.

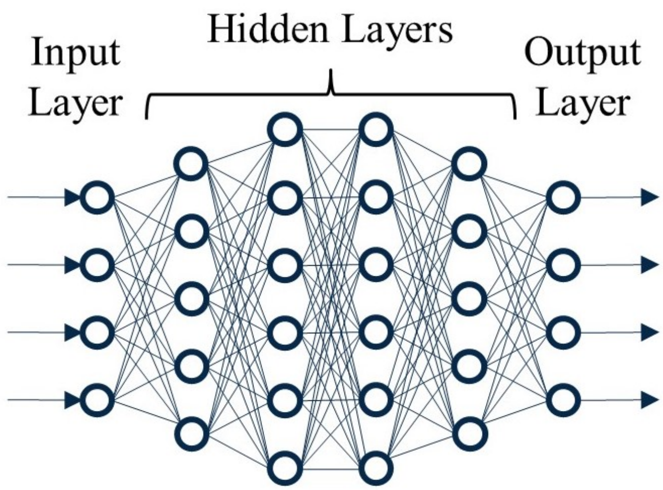

Figure 4.3: A representation of a fully-connected deep neural network.

4.2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are simply neural networks that use a convolution operation instead of matrix multiplication operation, at least in a number of starting layers of the network.

The convolution operation is applied on the input data X and a matrix of weight W_k that is called a kernel, in which k represents a certain Kernel k).

$$y_(i,j) = (W_k * X) = \sum_{a=0}^{\omega} \sum_{b=0}^{h} W_{k(a,b)}.X_{i+a,j+b} \tag{4.3}$$

$y_{(i,j)}$ represents the output of the convolution operation that is also called a feature map, ω and h are the width and height of the Kernel k, and (i,j) refer to a specific point in X. During the training of the CNN, often multiple kernels are used to produce different feature maps. It is common to add a pooling layer in between the CNN layers. Pooling continuously reduces the dimensionality of the model, reduces the number of training parameters, reduces the training time, and controls overfitting.

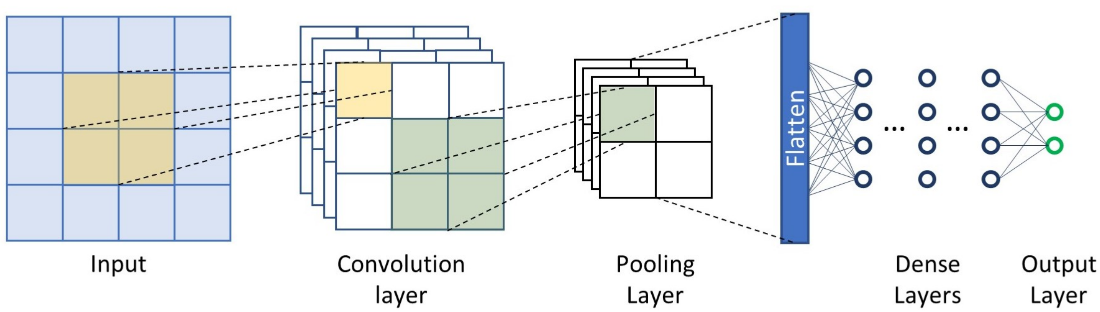

Figure 4.4: Representation of CNN used for classification.

CNN are able to capture spatial dependencies in the data, specially images. However, time series data can be formatted as an images for the CNN to be applicable on them [168]. The time and frequency representation images of the time series signals can be generated using short-time Fourier transform, wavelet transform, and Hilbert-Huang transform, etc. CNN are widely used for different applications since they are able to automatically extract features from the input data.

4.2.2.3 Recurrent Neural Networks

NN and CNN provide great capabilities, however they lack the ability to capture temporal dependencies. As shown in Figure 4.5, in a Recurrent Neural Network (RNN) architecture, the hidden states at each step,depends on the hidden state of the previous step. Therefore, the model gets the actual input at each time step along with the predictions of the previous time steps. Equation 4.4 shows the relation between different variables shown in Figure 4.5.

$$h_t = \sigma_h(U.X_t + V.h_{t-1} + b_h) \tag{4.4}$$

$$y_t = \sigma_y(W.h_t + b_y) \tag{4.5}$$

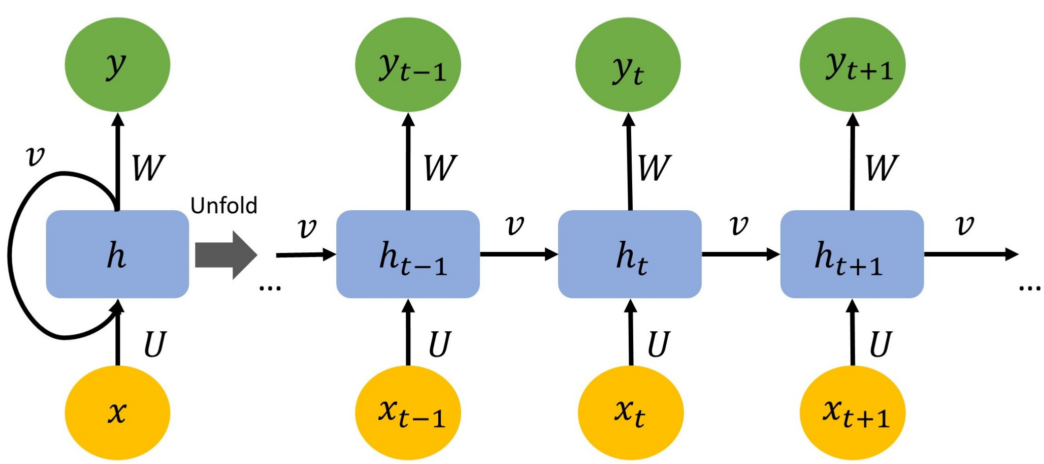

Figure 4.5: Representation of an unfolded recurrent neural network architecture.

Where v is a transition matrix, σ is a nonlinear function (e.g., tanh). RNN is a basis for many other variations that try to improve RNN performance in different ways. Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) are among these variations which are introduced to address the vanishing gradient problem of RNN.

4.3 Proposed Architecture

We previously proposed a framework at the intersection of PHM and PRA [169]. This architecture is the first step to actualize the mentioned framework. Architectures that are designed and built based on the this framework have two major parts, one is the "architecture construction" for each specific system, and the other is the "architecture application" on the CES and for supporting a desired objective. Another important feature in the proposed framework is the clear distinction between system-level analysis and assessment supported by the PRA techniques and subsystem-level analysis and assessment supported by the PHM techniques.

In this section, the proposed architecture for online multi-level condition monitoring of CES is presented. The integration of DL methods with FT is proposed, as shown in Figure 4.6. The required steps of using this architecture for any given system can be summarized below. For constructing the architecture, system-level and subsystem-level analysis has to be performed:

4.3.1 System-level Analysis:

The first step in system-level analysis is developing the systems' logic model. The systems' operation and/or failure logic has to be modeled based on engineering knowledge and systems design. This modeling can be done using PRA techniques

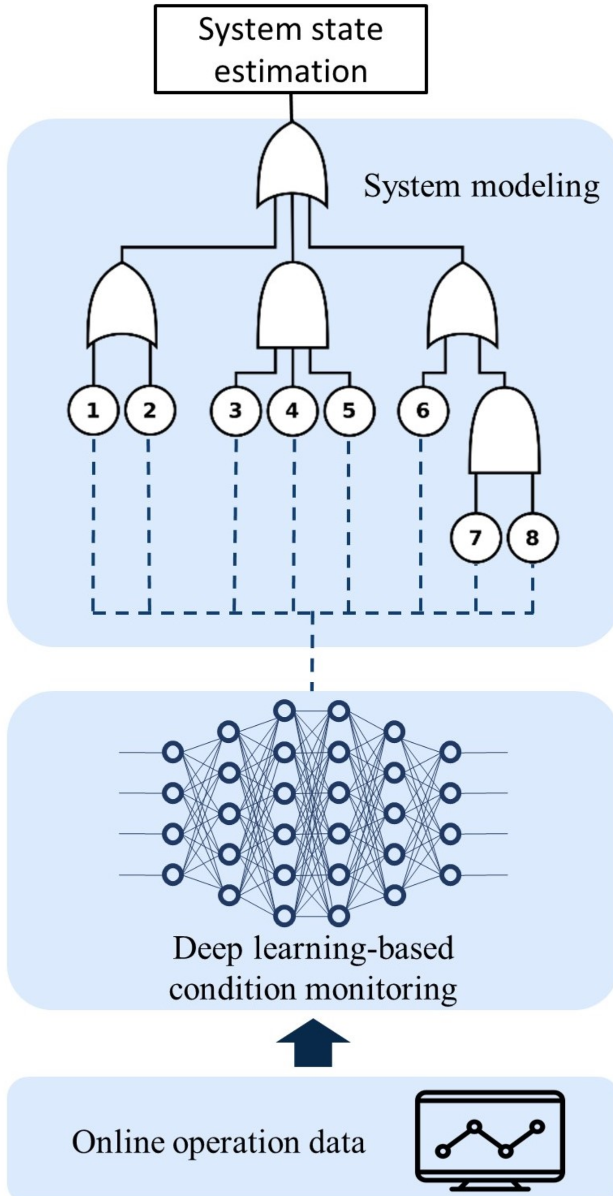

Figure 4.6: The graphical representation of the proposed architecture.

(FT in this case) as a medium for fusion of subsystems monitoring results, modeling the their dependencies, and modeling the possible failure propagation pathways. The basic elements of a FT diagram are an undesired top event, intermediate events, and basic events. Intermediate events represent failures propagated through the system and can be represented as logical combinations of basic events and other intermediate events, usually using the AND and the OR logical operators. Basic events are events that are not further developed in the FT and are usually used for representing component faults which are evaluated using the DL models in this architecture.

Another important consideration in developing the systems logic model using FT is taking into account mutual dependence or independence between the basic events. One of the advantages of fault trees is their ability to account for inter-dependencies and common causes. Therefore, components that are susceptible to common cause failure should be explicitly incorporated into the corresponding fault trees. Also, functional dependencies are another form of dependencies that should be identified and explicitly modeled in the fault trees. Ultimately, the minimum cut sets of the FT should be determined for better and faster calculation and inference of the model. A cut set is a combination of basic events that occurrence of one or more of them leads to the occurrence of the top event. All combinations that lead to the top event are explicitly shown below an OR logical gate in union of cut sets.

4.3.2 Subsystem-level Analysis

To develop the subsystems' CM models, first, historic data and maintenance logs for each component has to be collected. The available stored data for each component, maintenance logs, and the available data on similar components are all valuable sources of information in risk and reliability assessment applications. Collecting all the available data would help making a better subsystem CM model.

Second, considering each subsystem's operational behavior and available knowledge about it, the proper CM model should be selected. Deep Learning models are selected in this research, since they can be used without the need of knowing the exact physics of the problem as well as being a powerful tool when it comes to massive multidimensional data in which a physics-based model either does not exist or it is hard to obtain.

Training and fine-tuning the CM models, in this architecture DL-based CM models, is the next necessary step. In developing DL models for condition monitoring, there is a number of hyper-parameters that have to be chosen. These can be the structure of a DL model (i.e. number of neurons and hidden layers), batch size, learning rate, number of epochs, etc. Prediction accuracy is the most commonly used criteria to guide the hyper-parameter selection process.

The selected and trained/fine-tuned models are then used to assess the performance of components in a system during operation (online). Often, the dynamics of a component's behavior is such that requires the models to be revised/updated after a certain period of time. For DL models, the training process should be repeated after a while including the operation data of that period of time.

Ultimately, the integrated architecture that gets the CM data as input, and provides a subsystem-level and system-level measure operation condition is applied on a given system. This application provides a dynamic and system-level insight about the operation of the CES. This architecture should be expandable to any number of components with various configurations and be computationally efficient for online monitoring purposes.

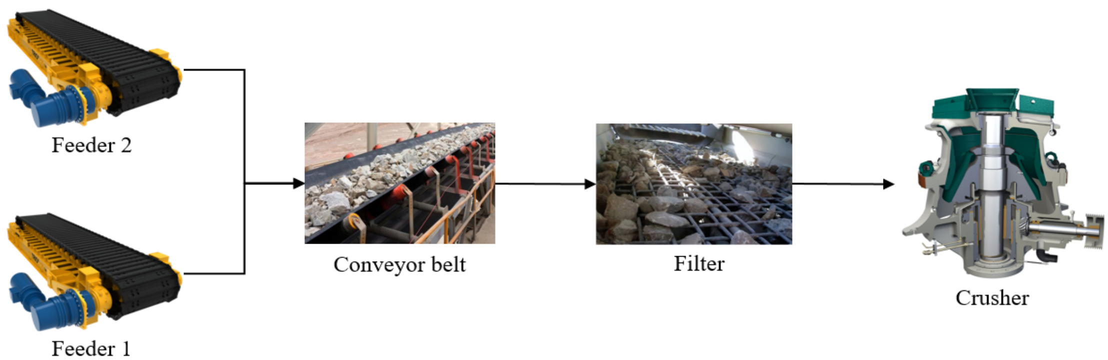

Figure 4.7: Crusher system layout.

4.4 Case Study on a Mining Stone Crusher System

4.4.1 System & Data Description

The proposed architecture is implemented on a real-world copper mine cone crusher system located in Chile. Its layout is shown in Figure 4.7, corresponding to a secondary crushing system. Two chain and pan feeders (Feeder_1 and Feeder_2) transport previously crushed copper ore rocks from the primary crusher system to a conveyor belt, which feeds the secondary cone crusher passing through a vibration separator filter. Each of these components is being continuously monitored by different sensors that are used to assess the health state of each component and the whole crusher system.

To perform multi-level (subsystem-level and system-level) monitoring on this system, a complete record of maintenance logs and sensor readings (spanning one year of operation) is used to train and test the proposed model. We did not have access to more details about the process and the components, so the understanding of the system's degradation process is completely data-driven. The whole system is being monitored by 21 sensors. Five sensors have more than 50% missing information, thus discarded. The remaining 16 sensors used in this case study are summarized in Table 4.1.

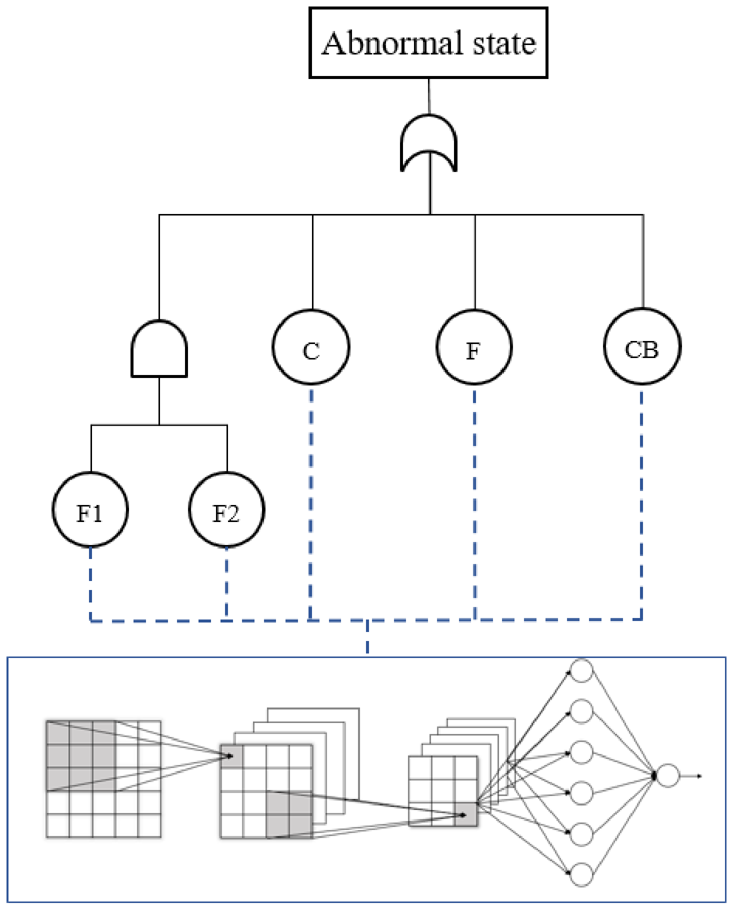

Figure 4.8: Crusher system fault tree.

All the sensors have the same sampling frequency of 30 *samples/hour* and measurements from 01/01/2018 to 01/01/2019 are used for this case study. This 16 sensors and a date-time indication log is provided for $N = 262,786$ time steps (will be referred to as "sensors file"). In addition, a complete maintenance log is provided and will be referred to as "maintenance file". The information provided in the maintenance file is summarized in Table 4.2.

Component	Measurement Type	Number of Sensors	Sensor Tag
Feeder 1	Electrical Current Velocity	1 1	s18 s19
Feeder 2	Electrical Current Velocity	1 1	s20 s21
Conveyor Belt	Load Electrical Current Velocity	1 1 1	s15 s16 s17
Filter	Electrical Current	1	s14
Crusher	Electrical Current Shaft Pressure Temperature Cone Ring Jump	1 1 4 2	s1 s2 s3,4,6,7 s8, 11

Table 4.1: Description of the 16 different sensor measurements used in this case study.

The system is considered operational unless a maintenance (corrective or scheduled) activity happen. A total of 37 maintenance activities were performed in 2018, corresponding to 28,104 time steps (i.e., 936,8h) in the sensors file, which are empty measurement entries. It is important to notice that failures are usually identified before a maintenance activity, but unless they require an immediate stop of the system, it keeps operating in what we will call "failure conditions".

The sensor file is divided into five "component files" with s_i columns, where

Date and time	start date & time end date & time
Type of action	Failure identified Corrective maintenance Scheduled maintenance
Impact	Individual component Whole system

Table 4.2: The columns in the maintenance file of the stone crusher system used to generate labels for the dataset.

s_i is the number of sensors associated to component i. Sensor measurements are mapped to the components health states which are explained below:

- *Normal*: the component is operating under normal conditions.
- *Abnormal*: the component is operating under failure conditions.

A DNN model is used to classify the health state of each component. As explained in the previous section, the health state of each component of the crusher system is used to calculate the top event probability of its corresponding fault tree, shown in Figure 4.8. The top event for the crusher systems FT is "the crusher system being in an abnormal state". The basic events, which represent the abnormal operation of individual system components, are Feeder_1 and Feeder_2 (F1 and F2), Conveyor Belt (CB), Filter (F), and Crusher (C). For example, a malfunction of CB, F, or C will lead to an abnormal operation of the system, while both feeders are set in parallel in the system layout to maintain the crushing rate, so the malfunction of only one feeder will not lead to a malfunction of the whole system; both feeders must be in abnormal conditions for this to happen.

4.4.2 Data Pre-Processing

To feed the proposed condition monitoring architecture with online monitoring data, a proper pre-processing step is needed. When time-series data is used, the model should still provide information of the system health state even when some operational sensors stop working. Therefore, for each component, a time-window is used to diagnose if each component is in normal or abnormal state. This time-window is constructed as a 2-D matrix consisting of $S = \{s_0, ..., s_i\}$ sensor features in its columns for $T = \{t_0, ..., t_n\}$ time steps in its rows. There are different ways to handle missing values in the sensor data. In this case, missing values are replaced by the previous time step measurement. This method is selected according to the Cumulative Density Function (CDF) for each sensor in each component. To illustrate, the CDF of the filter and the feeder_1 sensor measurements are shown in Figures 4.9, 4.10. One can see that there is no clear distinction between Normal and Abnormal sensor measurements, which is also true for the feeders and crusher.

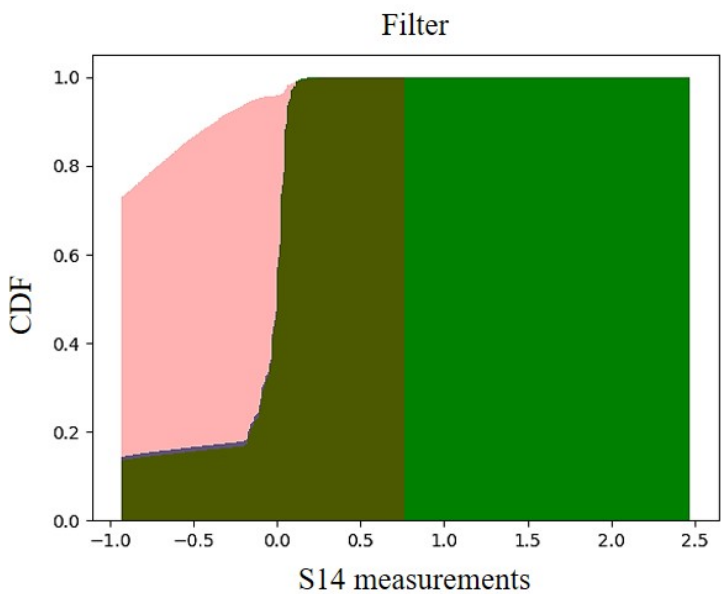

Figure 4.9: Cumulative distribution for the the sensor measurements in the Filter. Green: normal health state sensor measurements. Red: abnormal health state sensor measurements.

Other than the filter, the rest of the components have several sensor measurements. Therefore, sensor fusion should be used to better capture the sensors dependencies while training the DL model. Sensor data fusion demands a proper scaling of the data. As shown in Figures 4.9, 4.10, sensors readings always have at

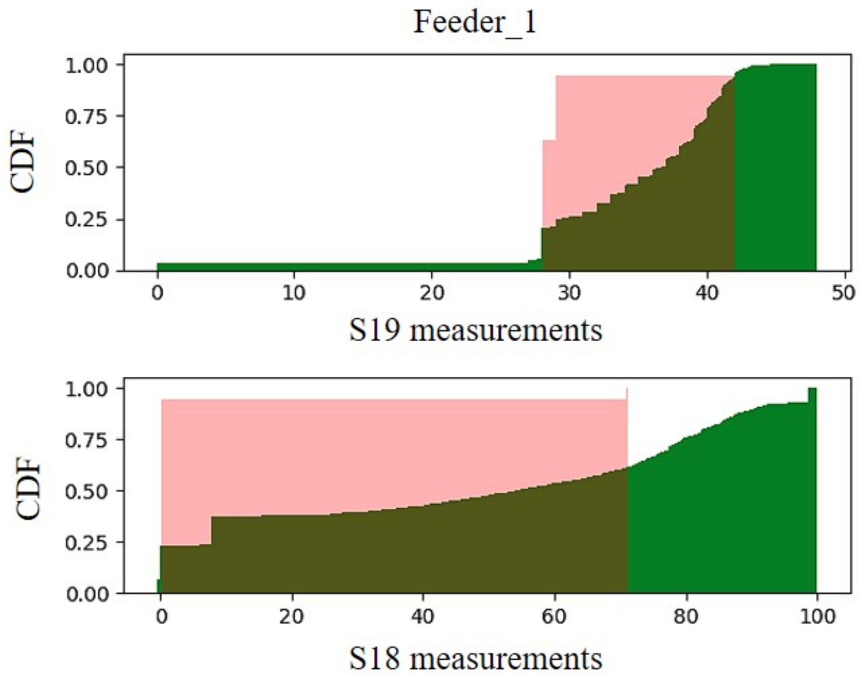

Figure 4.10: Cumulative distribution for the the sensor measurements in the Feeder_1 belt. Green: normal health state sensor measurements. Red: abnormal health state sensor measurements.

least a few outliers, so data should be scaled. In this study, min-max scaler is used to scale the measurements between -1 and 1 using equation 4.6.

$$x_{sc_i} = \frac{2(x_i - min(x))}{max(x) - min(x)} - 1 \tag{4.6}$$

Where x_{sc_i} is the scaled data set and x_i is a data point from the original dataset x. Doing so would bound the range in data and lower the standard deviation in it, diminishing the effect of outliers.

When the system is being maintained, the data collection stops and all values are NaNs. Thus, time steps in the sensor file that correspond to maintenance activities are eliminated as shown in Figure 4.11, generating $Fr = \{frame_1, ..., frame_{M+1}\}$

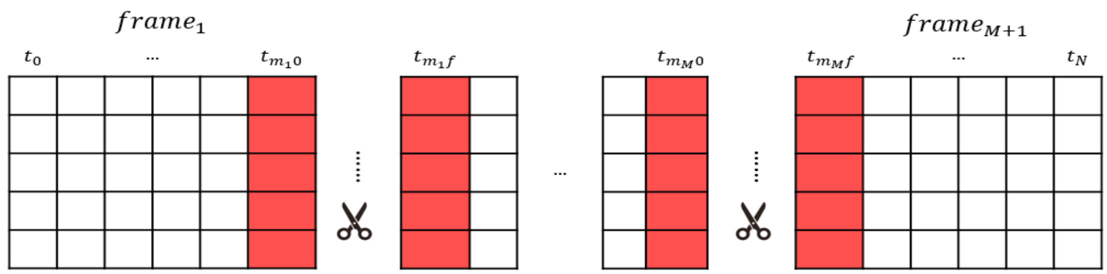

Figure 4.11: The process of removing maintenance data from the data set. $t_{m_i 0}$ and $t_{m_i f}$ are the start and finish times of maintenance respectively and M is the total number of maintenance's.

frames, where M corresponds to the number of maintenance's. Given that $M = 37$, a total of 38 frames are generated, corresponding to all sensor measurements while the system has been operating, these frames will be referred to as "operational frames". Also, using the maintenance file, $H = \{H_0, ..., H_N\}$ health state labels were generated for each time step in the sensor file. In the end, a total of $N = 234,682$ time steps remain to be studied.

Regarding the selection of the time-window size to feed to DL models, the maximum time-window size should correspond to the minimum length of the operational frames, which is 42 time-steps (i.e., the minimum operational length of the system between maintenance's is 1,4hr). Given this, a time-window with shape $S \times 42$ for each component is selected.

To test the trained model, the last two operational frames (i.e., 37 and 38) are used as the test set. Another important point to consider is the number and the percentage of the abnormal health states in the data available for each component. The number and the percentage of abnormal health state labels in each component's training data are the following:

- F1: 19 (0.008% of the total time steps)
- F2: 28 (0.012% of the total time steps)
- CB: 151 (0.064% of the total time steps)
- F: 3,237 (1.379% of the total time steps)
- C:2,448 (1.043% of the total time steps)

As it can be seen, there is a severe imbalance between the number of normal and abnormal health states. To tackle this problem, a data augmentation approach is performed over the training operational frames that contained abnormal health state labels, where each time-window is generated with an overlap of 41 time-steps (i.e., stride of 1), increasing the available abnormal data to better train the model. For the remaining frames in the train sets (which do not have any abnormality), each of them is divided evenly into time windows with equal length and and no overlap. Given this, 6,298 training time windows and labels where generated for F1, 27,943 for F2, 116,674 for CB, 133,824 for F and 167,266 for C. In the rest of this chapter, time-windows are going to be referred as x and health-state labels as y.

4.4.3 Training Process & Evaluation Metrics

To identify the crusher system abnormality, a data-driven DL model is developed for each of the system components to compute their likelihood of being in an abnormal health state. Then, each of these values are used to compute the probability value for the top event using the minimal cut sets of the crusher system FT, which is shown in equation 4.7.

$$\begin{aligned} TE = F1 \cdot F2 + F + CB + C - F1 \cdot F2 \cdot F - \\ F1 \cdot F2 \cdot CB - F1 \cdot F2 \cdot C - F \cdot CB - F \cdot C - \\ C \cdot CB + F1 \cdot F2 \cdot CB \cdot C + F1 \cdot F2 \cdot F \cdot CB + \\ F1 \cdot F2 \cdot F \cdot C + F \cdot CB \cdot C - F1 \cdot F2 \cdot F \cdot CB \cdot C \end{aligned} \tag{4.7}$$

For each of the basic events, $Pr(A) = Out(x, y)$, for $A \in \{F1, F2, F, CB, C\}$ and $Out(\cdot)$ is the output of the component data-driven model, which is the probability of the component being in an abnormal state.

For each component, a different DL-based model was trained. For F1 and F2, given that only two sensors were analyzed in each, the time window was flattened to a vector of size $42 \times 2 = 84$ to feed a DNN. For both the CB, given that 3 sensors where analyzed, a convolutional neural network (CNN) model was used as an automatic pre-processing step in the model, extracting relevant features from the time windows to feed a recurrent neural network (RNN) to analyze time related dependencies in data, whose output is summarized by a DNN. For F, given that it is a time-series measurement for only one sensor, a shallow RNN model is used.

Finally, C was analyzed using a deep CNN, where a first feature extraction step was done summarizing sensor measurements and time-steps (i.e., analyzing time evolution for fused sensors) and then a second step analyzed just time-steps (i.e., time evolution of the previous layer extracted information). All of these models' output is computed by a single neuron activated by a sigmoid function, as shown in equation 4.8, where z is the output of the neuron. Here, $\sigma(z)$ will take values between 0 and 1, which will correspond to the likelihood of estimating an abnormal health state given the trained model, so, as seen in the previous paragraph, $Out(x, y) = \sigma(z) \in [0, 1]$.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4.8}$$

Further information regarding each model architecture and hyperparameters can be found in Section 4.6.

To evaluate the performance of each component condition monitoring model, Confusion Matrix (CM) and Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) score is used. CM positive and negative rates (which correspond to normal and abnormal conditions) are computed as the model accuracy, whereas AUC score will give a summary of the model skill (i.e., will assign, on average, a higher probability to a randomly chosen true positive (TP) occurrence than a true negative one (TN)), were AUC = 1.0 and AUC = 0.0 corresponds to a model with perfect and no skill respectively.

4.4.4 Results

Table 4.3 shows the CM rates, ROC, AUC scores for all components. As it can be seen, TP and false negative (FN) rates are acceptable with the exception of the filter that only has one sensor. On the other hand, TN and false positive (FP) rates are generally acceptable for abnormal state estimation, even though the amount abnormal operation condition data was very small relative to normal condition.

Regarding AUC, it can be seen that the majority of the models present an acceptable skill with the exception of the filter again. It is worth stating again that there is a high class imbalance between normal and abnormal operation condition data, so, although FN rate seems low for each component, they represent a high amount of normal operational condition time steps that are estimated as abnormal. For example, for CB, only 44 [min] out of 58 [min] are correctly estimated as abnormal, but the FN rate corresponds to a total of 30.17 [hrs].

	TP [%]	FN [%]	FP [%]	TN [%]	AUC
F1	99.9	0.1	0	100	0.999
F2	96.8	3.2	23	77	0.944
CB	93.7	6.3	24	76	0.907
F	57.6	42.4	16.4	83.6	0.742
C	89.2	10.8	16.5	83.5	0.914

Table 4.3: Confusion matrix and AUC results for each component of the stone crusher system.

To show the capabilities of this architecture to estimate a system-level probability of being in an abnormal condition, the sensor data from 18:14 hrs of 12/06/2018 to 13:56 hrs of 12/26/2018 is used (i.e., a total of 14,272 time steps). Table 4.4 shows the confusion matrix related to the obtained health state estimations. According to Table 4.4, both positive and negative rates are high, but, as explained previously, this can be misleading. The total amount of abnormal operation time steps is of 10.7 [hrs], but the FN predictions corresponds to a total of 77.8 [hrs], so abnormal operation conditions are over-estimated by a factor of 7.

To illustrate the online monitoring capabilities of the proposed approach, the real-time estimated results from 3:30 hrs to 23:30 hrs in 12/11/2018 are shown in

	TP [%]	FN [%]	FP [%]	TN [%]
System	83.3	16.7	8.4	91.6

Table 4.4: Top event estimations confusion matrix in the crusher system FT.

Figure 4.12. It can be seen that the abnormal operation conditions of the system can be estimated accurately, but there is a non-negligible amount of miss-estimations (false alarms) that should be assessed.

Regarding this particular case study, the main future tasks to be done to improve the amount of miss-estimations are: (1) apply moving average for smoothing the signals and their variations in the pre-processing stage and (2) perform a time window size analysis that better captures degradation processes. Also, eliminating redundant portions of the normal operation can help to alleviate the class imbalance problem

4.5 Discussion & Conclusion

In this chapter, deep learning models are trained to monitor the condition of each subsystem. A fault tree is developed to represent the crusher system's structure (to the best of our knowledge) and is further used to fuse all the individual condition monitoring results and provide an online system-level state estimation. By doing so, a system modeling technique, used in PRA and other areas of engineering risk assessment is integrated with the state-of-the-art deep learning models, frequently used by the PHM community, to assess the condition of various engineering devices.

This mathematical architecture is novel, general, flexible, and easy to imple-

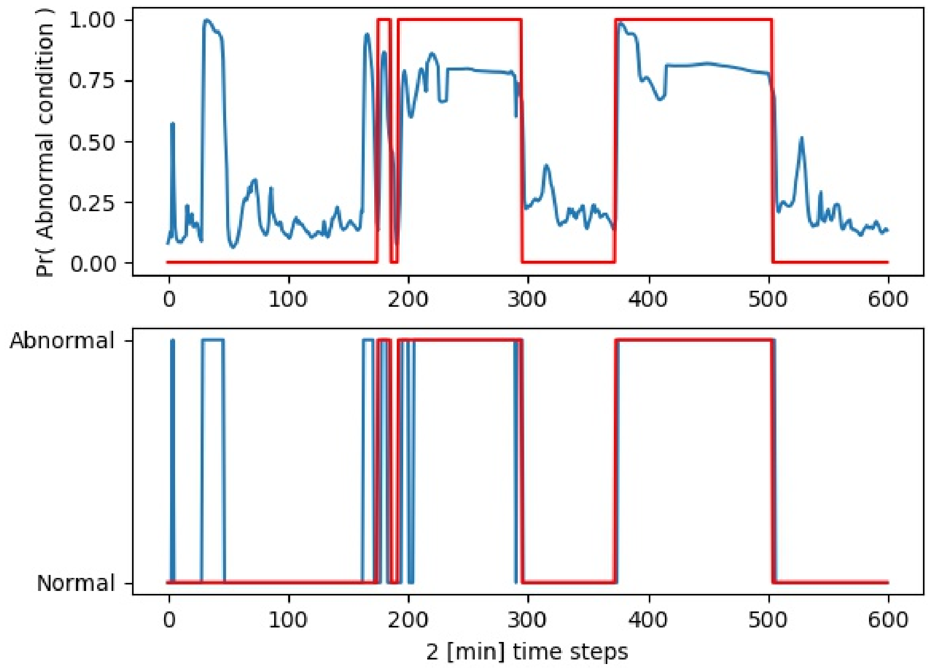

Figure 4.12: Online crusher system FT top event estimation from 3:30 hrs to 23:30 hrs in 12/11/2018. Top: In blue, probability of the model to estimate an abnormal condition. In red, ground truth likelihood. Bottom: In blue, smoothed estimated operational condition. In red, ground truth operational condition.

ment which are all features that make it a potentially useful approach for operation monitoring of CES. Further, the results are promising in the sense that they prove the viability of our proposed framework in Chapter 3 and the objectives defined for it such as the capability to perform dynamic and multi-level analysis.

However, further improvements are needed for establishing it as a solid system-level condition monitoring tool. For example, uncertainty handling and management in such an architecture is not addressed in this chapter and needs future research. Scalability of this methodology and its limits in terms of data quantity and number of system elements is another question that is needed to be addressed. Further, this architecture is not capable of incorporating human and software elements which are both important elements of CES.

This study is the first step to explore the undiscovered potential of such integration. Considering the lessons learned in this chapter, the SIPPRA framework is to be further developed in chapter 5.

4.6 Appendix

Crusher model:

- Layer 1: Convolutional layer. Filter_size = (14,2), filters = 256, activation = tanh, padding = same.
- Layer 2: Convolutional layer. Filter_size = (14,1), filters = 128, activation = tanh, padding = same.
- Layer 3: Convolutional layer. Filter_size = (7,2), filters = 64, activation = tanh, padding = valid.
- Layer 4: Flattening layer.
- Layer 5: Dense layer. Neurons = 128, activation = selu.
- Layer 6: Dropout. Rate = 0.2.
- Layer 7: Dense layer. Neurons = 128, activation = selu.
- Layer 8: Output layer. Neurons = 1, activation = sigmoid.

Here, Adam optimizer and binary cross-entropy loss were used for 100 training epochs with batch size of 1042 time windows. Normal and abnormal labels where balanced in the loss function by a rate of 1/30.

Filter model:

- Layer 1: Recurrent Neural Network layer. Neurons = 256.
- Layer 2: Recurrent Neural Network layer. Neurons = 64.
- Layer 3: Flattening layer.
- Layer 5: Dense layer. Neurons = 128, activation = tanh.
- Layer 8: Output layer. Neurons = 1, activation = sigmoid.

Here, RMSprop optimizer and binary cross-entropy loss were used for 100 training epochs with batch size of 1042 time windows. Normal and abnormal labels where balanced in the loss function by a rate of 1/6.

Conveyor belt:

- Layer 1: Convolutional layer. Filter_size = (14,2), filters = 256, activation = tanh, padding = same.
- Layer 2: Recurrent Neural Network layer. Neurons = 256.
- Layer 3: Flattening layer.
- Layer 4: Dense layer. Neurons = 128, activation = tanh.
- Layer 5: Output layer. Neurons = 1, activation = sigmoid.

Here, Adam optimizer and binary cross-entropy loss were used for 100 training epochs with batch size of 1042 time windows. Normal and abnormal labels where balanced in the loss function by a rate of 1/300.

Feeder_1:

- Layer 1: Dense layer. Neurons = 32, activation = tanh.
- Layer 2: Batch normalization layer.
- Layer 3: Dense layer. Neurons = 32, activation = tanh.
- Layer 4: Dropout. Rate = 0.2.

- Layer 5: Dense layer. Neurons = 32, activation = tanh.
- Layer 6: Dense layer. Neurons = 32, activation = tanh.
- Layer 7: Dense layer. Neurons = 32, activation = tanh.
- Layer 8: Dense layer. Neurons = 32, activation = tanh.
- Layer 9: Dropout. Rate = 0.2.
- Layer 10: Output layer. Neurons = 1, activation = sigmoid.

Here, Adam optimizer and binary cross-entropy loss were used for 100 training epochs with batch size of 1042 time windows. Normal and abnormal labels where balanced in the loss function by a rate of 1/300.

Feeder_2:

- Layer 1: Dense layer. Neurons = 32, activation = tanh.
- Layer 2: Dense layer. Neurons = 32, activation = tanh.
- Layer 3: Dense layer. Neurons = 32, activation = tanh.
- Layer 4: Dense layer. Neurons = 32, activation = tanh.
- Layer 5: Dense layer. Neurons = 32, activation = tanh.
- Layer 6: Dropout. Rate = 0.2.
- Layer 7: Output layer. Neurons = 1, activation = sigmoid.

Here, Adam optimizer and binary cross-entropy loss were used for 100 training epochs with batch size of 1042 time windows. Normal and abnormal labels where balanced in the loss function by a rate of 1/300.

Chapter 5: Development of a Condition and Operation Risk Monitoring Architecture for Complex Engineering Systems by Integrating PRA and PHM Techniques

5.1 Introduction

Complex Engineering Systems (CES) are dynamic and consist of diverse, interconnected, and interdependent machine and human elements. They operate at multiple scales and levels, and are normally governed by non-linear relations. In addition to these CES' characteristics, the operation data collected from such systems is complex as it is high-volume, multi-dimensional, multi-source, and incorporates various data collection frequencies. These features make condition monitoring and risk assessment of CES a difficult challenge for the risk and reliability research community. Risk assessment and Prognostics and Health Management (PHM) are two major and distinct subfields of reliability engineering which seems to have complementary characteristics in the context of CES risk and reliability analysis [169]

Risk assessment studies' primary focus has been providing a system-level perspective with emphasis on engineering knowledge and systems logic modeling in an offline and static manner [3, 170]. The risk community has identified the need for a more dynamic analysis, leading to the emergence of Dynamic Risk Assessment (DRA) concept and studies [34, 36, 39]. An important part of the available literature on DRA is focused on updating the analysis using accident precursor data [49, 171, 172]. Although doing so would improve the accuracy of the risk assessments, it limits the scope of the analysis to only the times that an accident is initiated and in progress. Perhaps the richest source information for updating the risk assessments is operation data or Condition Monitoring data; there are a number of studies that have tried, to some extent, use this source of information [53, 54, 57]. These are valuable advancements, yet there is a need for a comprehensive approach that enables addressing both the system and the data complexity, utilization and fusion of all sources of information, and provides a dynamic, system-level measure of operation risk [169].

On the other hand, PHM research has focused on developing methods for handling high-volumes of multi-dimensional condition monitoring data [9, 80, 173, 174]. However, PHM lacks the system-level perspective, relies heavily on the operation data, and its outcomes are not risk-informed [169]. To overcome these gaps, we previously proposed the systematic integration of PHM and Probabilistic Risk Assessment (PRA) (SIPPRA) methods and practices to enable a data-driven PRA and risk-informed PHM together in one analysis framework [169].

The next step beyond the conceptual framework is to define candidate mathematical and computational architectures that underpin the framework and enable combining data and models from PRA and PHM to support the objectives of system-level insight. We explored the viability of system-level condition monitoring by integrating Deep Learning (DL) models, trained for individual subsystems, and a Fault Tree (FT), representing the operation logic of the system in Chapter 4 [175].

In this chapter, we present a novel mathematical architecture for condition and operation risk monitoring of CES. Here, a Bayesian Network (BN) is used to fuse subsystems condition monitoring results and provide system-level condition monitoring insight. The constructed BN also incorporates the possible accident scenarios and continuously updates the probability of adverse events in the system based on the condition monitoring results. BN is chosen due to its capability to model complex relationships, providing an understandable and compact visual representation, as well as the ability to perform both diagnostics and prognostics [18]. BNs can also perform inference with incomplete data, allow uncertainty analysis/propagation throughout the system, and can compute the occurrence probability of certain events in the system [176]. Within this architecture, DL models are trained to provide a probabilistic measure of health for the main subsystems to support condition monitoring. The outputs of the DL models then become the input values to the root nodes of the BN developed for the system. There is a wide range of DL models to choose from depending on the system, amount of data, availability of labels, and the desired tasks [67, 177, 178]. In this chapter, the applicability of Bayesian Neural Nets (BNNs) are assessed since they allow uncertainty quantification (UQ) in predictions (theory is explained in Section 5.2) which is considerable advantage, specially for risk analysis.

Furthermore, this approach provides system-level insight and enables using various sources of information, handling the abundant condition monitoring data, and modeling the subsystems interactions. Moreover, the use of DL reduces the size of the BN required to model the CES and its computational expense for continuous condition and risk assessment. A detailed description of this architecture is provided in Section 5.3. In Section 5.4, we demonstrate the potential practicality and advantages of our proposed mathematical architecture as an important step towards developing a comprehensive risk and condition monitoring architecture for CES by presenting a case study on a real-world Vapor Recovery Unit (VRU). The future work and implications of this study are discussed in Section 5.5. Finally, the chapter concludes in Section 5.6.

5.2 Preliminaries & Background

This section provides a literature review and an introduction to the related theoretical methods used in this chapter, including Bayesian Network (BN) and Deep Learning (DL) models, particularly BNNs.

5.2.1 Bayesian Network

Various methods and algorithms are used for modeling functional relations between systems' components, different failure scenarios, process modeling, fault diagnosis and reasoning, and risk/reliability assessment of CES. One of the main modeling methods is BNs. In the late 90's, BNs were first adopted for systems' reliability assessment [179, 180]. Since then, numerous studies have used BNs for risk and reliability assessment [181, 182], accident modeling [183], diagnostics [18, 51, 184], prognostics [185], and human reliability analysis [186, 187].

BNs or Bayesian Belief Networks (BBNs) are a type of graphical models that combine graph theory and probability theory. BNs consist of a number of nodes and a number of edges. Each node in the graph represents a unique random variable and the edges between the nodes represent probabilistic dependencies among the corresponding random variables. A BN is a Directed Acyclic Graph (DAG) which means all the edges are directed and there are no cycles in the graph, meaning one cannot travel from any node in the graph, in the right direction, and arrive back at the starting node.

Considering n random variables $X_1, \ldots, X_n$, a BN would have n nodes where node $j (1 \leq j \leq n)$ corresponds to the random variable X_j. The joint probability of random variables in the BN would then be defined as:

$$P(X_1, \ldots, X_n) = \prod_{j=1}^{n} P(X_j | parent X_j) \tag{5.1}$$

The parents of X_j are the set of all variables X_i such that an incoming edge connects node i to node j.

In comparison to other available methods to model a CES such as FT and Event Tree (ET), BN is proven to be a more flexible method to incorporate various sources of information, offers the ability to model dynamic and time-dependent scenarios (Dynamic Bayesian Networks (DBN)), and enables uncertainty analysis. Also, BNs provide diagnostic capabilities using evidence propagation and Most Probable Explanation (MPE) derivation even with limited observations [176, 188].

To use a BN for any given problem, three main steps are required:

1. **BN structure modeling:** the structure of a BN can be determined based on expert opinion or engineering knowledge (in this context) or can be learned from the data via structure learning methods specific to BNs.

2. **BN parameters determination:** BN parameters to be quantified include state probabilities for root nodes and quantifying conditional probabilities (Conditional Probability Tables (CPT) or Conditional Probability Functions (CPF)) which model the probabilistic relationship between connected nodes [189].

3. **BN inference:** a quantified BN can be used to perform forward or backward inference, including diagnostic analyses with various kinds of inference algorithms which are based on Bayes' theorem [190, 191]. Inference algorithms are divided into exact (message-passing, conditioning, junction tree, symbolic probabilistic inference, etc.) and approximate inferences (stochastic sampling, search-based algorithm, and loopy belief propagation algorithm).

For brevity, only a short introduction to the most important aspects of BNs is provided here. For more details, the reader is referred to textbooks, e.g., [192, 193].

As the number of nodes and edges in BNs increases, performing reasoning and belief propagation becomes more computationally expensive as BN inference is an NP hard problem [194]. Thus, BN application for dynamic CES with abundant, heterogeneous data and a large number of sensors and components is limited. To extend the applicability of BNs in the context of reliability engineering, various aspects of BNs have been studied. For example, frameworks have been proposed on how to represent discrete [195] and continuous [196] time representations. To address the computational cost of inference in complex BNs, proper inference algorithms are identified by Iamsumang et al. [100] and Zheng et al.[197]. To gain even more computational advantage, systematic approaches to efficiently model systems using a BN are proposed (e.g., chain-like BN structures [198] and Object-oriented Bayesian Networks [199]).

Despite the recent advances in BN quantification and inference, modeling a CES using only a BN is still problematic. To mention a few problems, modeling a CES starting from condition monitoring sensors to the system-level health indicator requires numerous nodes. The relations between these nodes have to be expressed as CPTs which requires a detailed understanding about the system on different levels and is not practical. Further, for monitoring the condition and operation risk of the system, abundant streaming condition monitoring data should be continuously analyzed and doing so using BNs is still computationally expensive. This motivates the use of DL along with BNs.

5.2.2 Deep Learning

Data-driven methods, particularly machine learning and DL methods, are widely adopted for PHM applications such as anomaly detection [77], fault diagnostics [78], and prognostics [79]. This is due to their unique ability to handle large amounts of highly non-linear data. DL models can process the operation data and automatically generate features for any given task including detection, classification, or prediction of patterns in the data. This minimizes the need for domain expertise and extensive feature engineering in cases where complete, fully representative data is available.

One can categorize the different learning problems into the four main groups of Supervised, Unsupervised, Semi-supervised, and Reinforcement learning. To implement any DL algorithm, three things are required: (1) data for training and testing the model, (2) an objective function to optimize, and (3) an optimization scheme. Variations in these three components would result in generation of numerous distinct DL algorithms and architectures such as Convolutional Neural Networks (CNNs) [80], Recurrent Neural Networks (RNNs) [81], Long Short Term Memory Networks (LSTMs) [82], Generative Adversarial Networks (GANs) [9], and Autoencoders [79].

The mentioned DL algorithms operate as black boxes and mainly in a deterministic manner, making both the interpretability and Uncertainty Quantification (UQ) of their predictions a challenging task. This is an established gap in the reliability engineering field, as highlighted by Khan and Yairi [67] and Xu and Saleh [200].For decision-making processes, particularly when it comes to safety-critical systems, the ability to understand the basis for prediction and quantify prediction uncertainties is crucial.

There are a number of recently introduced methods that perform UQ for deep learning models. One method is to use an ensemble of models; that means training many networks independently on the same data with different initializations and using the output of all models to obtain upper and lower bounds of prediction [201]. Another method is Dropout, a technique in which randomly selected neurons are ignored in the training process. Dropout is a regularization technique and can be interpreted as a Bayesian approximation of a Gaussian process [202]. Finally, probabilistic deep learning, in particular, BNNs [203] are capable of computing a credibility interval for their predictions (explained in Section 5.2.3).

Ensembles can be computationally inefficient as a number of different models are trained and run in parallel. Dropout methods are mathematically sound, yet they can produce overconfident wrong predictions and are less robust when compared to Bayesian methods [201]. Bayesian neural nets are well established and can transform into various widely used architectures such as RNN [204] and LSTM[205]. They provide a formal approach to quantify uncertainties both in the model parameters and the data and, at the same time, are proven to be less sensitive to noise compared to their equivalent deep neural nets. As such, in this chapter, BNN is used to enable UQ as well as condition monitoring.

5.2.3 Bayesian Neural Network

A neural network (NN) consists of a number of interconnected nodes that perform nonlinear operations to map a n-dimensional input space to a desired output space. A BNN is a stochastic neural network that is trained using Bayesian inference. Stochastic NNs have stochastic components in them such as stochastic activation functions or stochastic weights as shown in Figure 5.1. This stochasticity provides

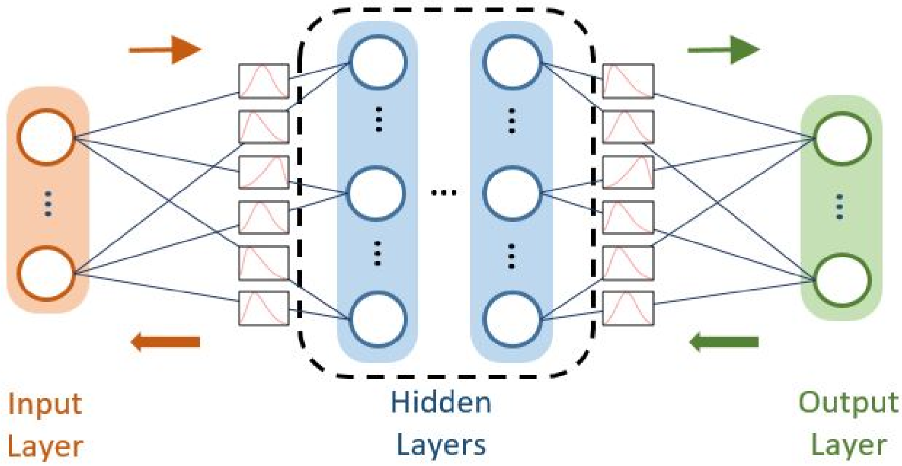

Figure 5.1: A deep Bayesian neural network architecture with weights shown as distributions. The arrows represent the training process using Bayes-by-backprop algorithm.

an infinite ensemble of models with their associated probability distributions [206]. BNN can provide uncertainty estimation and better handle overfitting relative to regular a NN.

The parameters in a BNN (i.e., weights and biases (W)) are initialized with a certain prior distribution $p(W)$. During model training, using data (D), the prior distributions are updated ($p(W|D)$) in every iteration following the Bayes' rule:

$$p(W|D) = \frac{p(D|W)p(W)}{p(D)} \tag{5.2}$$

where $p(D)$ is the model evidence and $p(D|W)$ is its likelihood. After training the model, the predictive distribution $p(y|x, D)$ for any input x can be obtained as follows:

$$p(y|x, D) = \int p(y|x, W)p(W|D)dW. \tag{5.3}$$

where y is the model output. In the context of NNs, calculation of the posterior $p(W|D)$ is mathematically intractable due to the high dimensionality of the sampling space. To address this challenge, two main algorithms are used: (1) Markov Chain Monte Carlo (MCMC) methods, or (2) variational inference, which provides an approximation of the posterior distribution. Considering the better scalability of variational inference, it has become more popular in the context of DL and is used frequently to train BNNs [207].

Variational inference approximates the posterior $p(W|D)$ using a family of densities over the latent variables, parameterized by free variational parameters (θ) (e.g., for Gaussian distributions, θ includes vectors of means μ and variances Σ). These parameters are learned such that the variational distribution $q_\theta(W)$ is as close as possible to the exact posterior distribution $p(W|D)$. Kullback–Leibler (KL) divergence is the commonly used measure to compare how close the two distributions are:

$$D_{KL}(q_\theta||p) = \int_W q_\theta(W')log(\frac{q_\theta(W')}{p(W'|D)})dW'. \tag{5.4}$$

where, W' is the integration variable. Looking at Equation 5.4, $p(W|D)$ still needs to be calculated. To address this issue, evidence lower bound (ELBO) is used in the optimization scheme [174]:

$$\begin{aligned} ELBO &= \int_W q_\theta(W') log \frac{p(W', D)}{q_\theta(W')} dW' \\ &= log(p(D)) - D_{KL}(q_\theta || p) \end{aligned} \tag{5.5}$$

As shown in Equation 5.5, $log(p(D))$ only depends on the priors, meaning that minimizing $D_{KL}(q_\theta||p)$ is similar to maximizing the ELBO. The stochasticity introduced to the BNNs parameters is not compatible with the traditional back-propagation method to train the network [208]. Bayes-by-backprop [203] is the commonly used algorithm to address this problem (Figure 5.1).

Bayes-by-backprop algorithm is based on the primary idea of using some random variable $\varepsilon \sim q(\varepsilon)$ to indirectly sample θ through a deterministic transformation $t(\varepsilon, \phi)$, such that $\theta = t(\varepsilon, \phi)$ follows $q_\phi(\theta)$. This transformation makes all the variables non-stochastic, allowing the back-propagation to work on BNNs as well. This change of variable is also referred to as the reparameterization trick [209]. BNNs are more computationally expensive to train relative to regular neural nets, as a considerable amount of sampling and estimation is performed, and the additional stochasticity of the model results in delayed convergence during optimization. For this reason, novel approaches such as Flipout [210], which decorrelates the gradients within a mini-batch are introduced to accelerate the training process.

5.3 Proposed Architecture

In this chapter, we present a mathematical and computational architecture that enables combining data and models from PRA and PHM to support the objectives of system-level insight. Following the proposed framework in [169], a CES is analyzed in this architecture in two major parts: system-level and subsystem-level. As shown in Figure 3.3, with a white box in the bottom left corner, we begin with collecting information about the system. Information such as operation data, failure data, maintenance records, inspections, system description and functional logic, and possible consequences of failures and accidents are collected. Also, since real-world operation data is used, data cleaning and preprocessing has to be performed. Essentially, the union of the needed information for PHM and PRA applications is collected and prepared at this step.

5.3.1 System-level Analysis

As explained in [169], in system-level analysis, the focus is on the system functional logic, scenario development, and consequence analysis. The system-level analysis and modeling enables system-level condition monitoring and operation risk monitoring. In this part of the architecture, indicated with blue color in Figure 3.3, a Hazard and Operability (HAZOP) study is performed to gain a thorough understanding of the system from both the functional and the risk assessment perspectives.

Second, a complete system BN is developed with the general outline presented in Figure 5.3. The BN incorporates all the sensors and other possible sources of system information at hand. However, since modeling every component and sensor in the system using a BN will be computationally intractable in modern CES, the BN is reduced by aggregating nodes related to one subsystem into one single node. This aggregated node represents the probability of the subsystem being in an abnormal condition. Although the BN node aggregation reduces the computational cost, as the BN is reduced and integrated with DL models, the traceability of the model is reduced as well. Therefore, depending on the application, the amount of node reduction should be carefully determined. The single aggregated node is evaluated using the trained DL models or any other data-based approach depending on the amount of available data. These models are the outcome of the subsystem-level analysis explained in Section 5.3.2.

5.3.2 Subsystem-level Analysis

The collected operation, maintenance, and failure data are used in this part of the architecture to develop PHM models (monitoring, diagnostics, and prognostics models) for the individual subsystems. As indicated in Figure 3.3 with green color, depending on the availability of the labeled data, a machine learning algorithm is selected and trained to provide a measure for the operation condition of the subsystem. For example, an Auto-Encoder for unsupervised anomaly detection [79], Vanilla deep neural networks or Convolutional Neural Networks (CNN) for supervised fault classification in subsystems [211], Physics-Informed Neural Networks

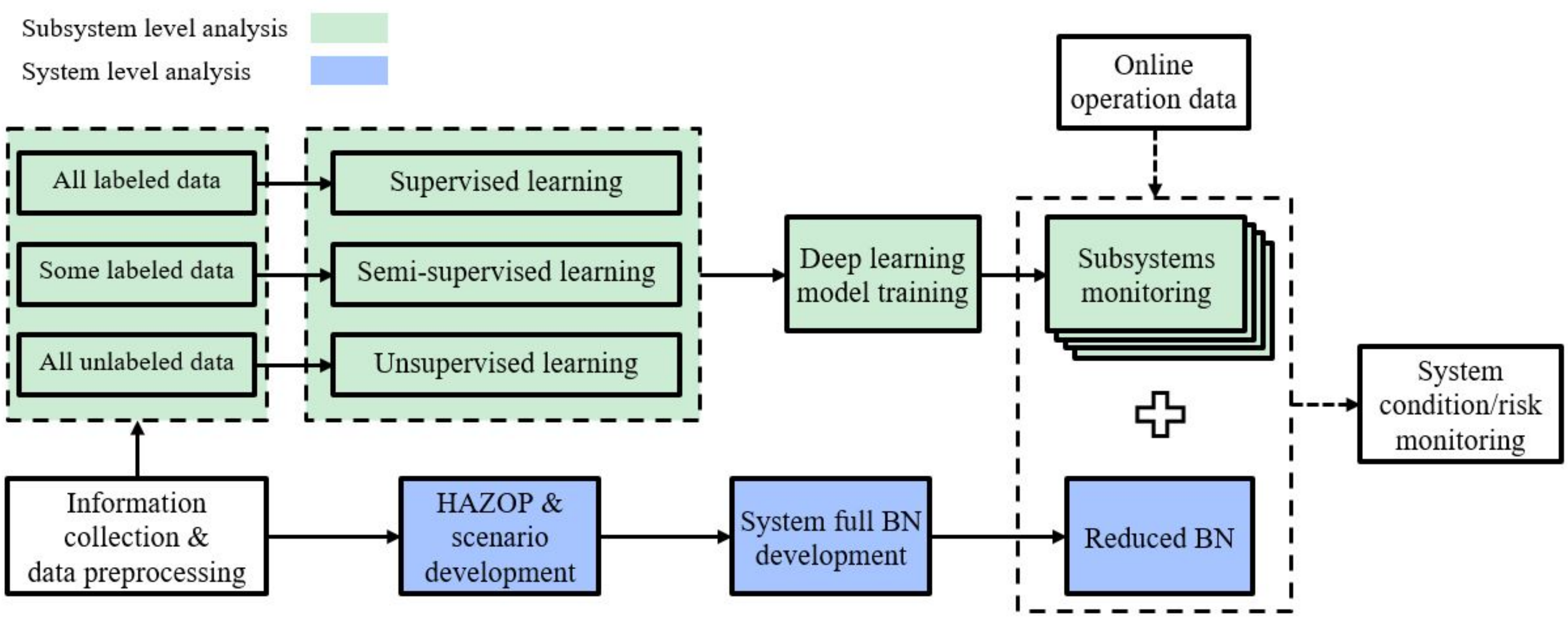

Figure 5.2: The proposed architecture for CES operation and risk monitoring. Subsystem-level analysis is indicated using green color and system-level analysis is indicated using the blue color.

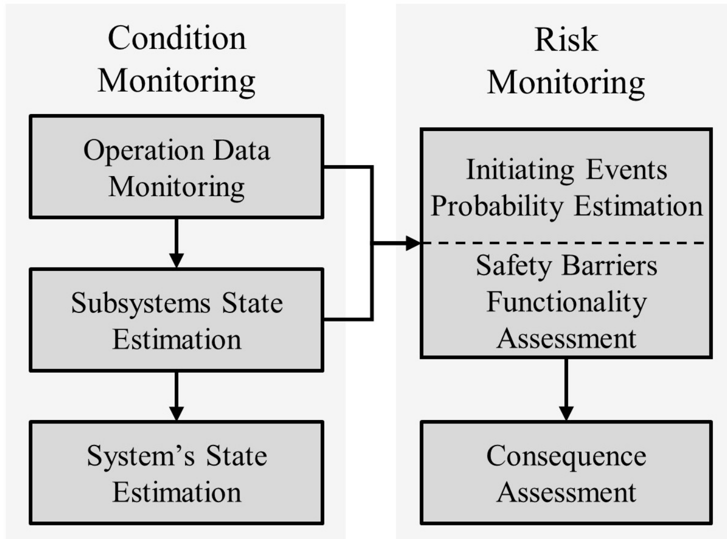

Figure 5.3: The general outline for modeling the system using a Bayesian network in the proposed architecture of this chapter.

(PINN) to incorporate the available equations for degradation or normal operation into the DL models [212, 213], Recurrent Neural Networks and Long Short Term Memory (LSTM) [82, 214] networks for predicting the operation condition of the components in future time steps which introduces predictive abilities to the whole architecture.

When monitoring the operation of the system as a whole, the monitoring data is fed to the trained monitoring algorithms and the output of these algorithms quantifies the root nodes of the BN, called "subsystem-level nodes" in this study. By doing so, the large amount of multi-dimensional operation data is handled by pre-trained DL algorithms and not the BN, while the system's functional logic and subsystems' relations is represented by the BN.

5.4 Case Study

To demonstrate the applicability of the proposed architecture, a Vapor Recovery Unit (VRU) is studied. This system has multiple heterogeneous components and subsystems, behaves non linearly, and has several sources of uncertainty. Other complexity characteristics either do not exist or we have neglected them in our study due to lack of information. One year of real-world operation data of a VRU system in a natural gas compression system is used. These types of natural gas compression systems are an integral part of oil and gas production, storage, and delivery processes. Figure 5.4 demonstrates the role of a VRU in the production process of oil and gas.

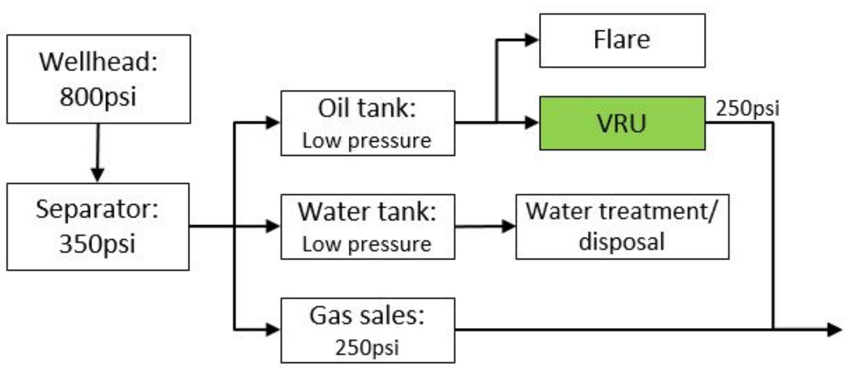

Figure 5.4: The role and installment place of a VRU unit in oil and gas production processes.

5.4.1 System Description

The studied VRU is composed of two rotary compressors, two heat exchangers, and two scrubbers, along with many sensors, valves, and other types of safety equipment (as shown in Figure 5.5). The main purpose of this unit is to recover vapors formed inside completely sealed crude oil or condensate tanks. A switch detects pressure variations inside the tanks and turns the compressor on and off. The vapors are sucked through a scrubber, where the liquid residues are trapped and returned to the liquid pipeline system or to the tanks, and the vapor recovered is sent to the gas lines.

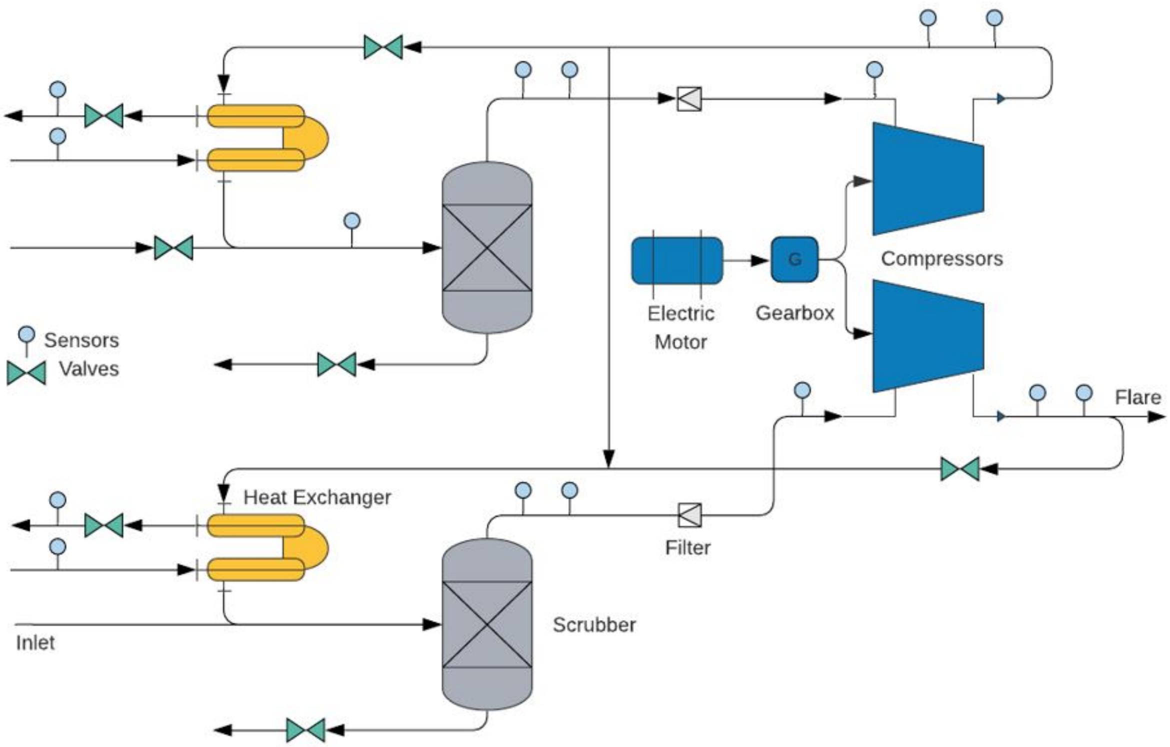

Figure 5.5: P&ID of the VRU system which includes two heat exchangers, two scrubbers, a compressor module (blue color); which incorporates a two stage compressor (two compressors in series), an electric motor, and a gearbox, along with numerous different sensors and valves.

5.4.2 System-level Analysis & Modeling

To properly model the system using a BN, system Piping and Instrumentation Diagrams (P&IDs) and maintenance records were studied. Next, the common and most important abnormal states, systems' safety barriers, and possible adverse consequences of the abnormal states are identified as shown in Table 5.1. Evidently, the most important subsystems are the two compressors due to a higher number of significant abnormal states and the severity of the possible consequences of these states.

The purpose of this modeling is to enable system-level condition and operation risk monitoring by incorporating the interactions among the subsystems in terms of system abnormal behavior and/or failure. The general architecture illustrated in Figure 5.3 is followed to build a BN for the VRU system. The constructed BN is Figure 5.6. This BN has two main sections; namely condition monitoring and risk monitoring. The important sensor measurements (in this case compressors suction and discharge pressures) and subsystems condition monitoring nodes (the two heat exchangers, two scrubbers, and the compressor module which incorporates the two compressors and the electric motor) are placed in the middle of the network and are called "subsystem-level nodes," indicated with black outline. Among the subsystem-level nodes, the ones with black dashed outlines are "live" and their values are updated at each time step. The "Scrubber1&2," and "Compressor Module" nodes are quantified using the output of the trained BNNs, and the measured suction and

Component	Abnormality	Consequences	Safety Barriers
Scrubber	High Level	Liquid overflow to compressor, compressor damage, Compressor vibration	Operator action & alarm, compressor trip
Compressor	1-High discharge pressure 2-High suction pressure 3-Low suction pressure 4-Other abnormalities (mainly vibration)	1-Damage to pipes & flanges, leakage, potential for fire 2-Vacuum in storage vessel 3-Compressor vibration & damage to compressor 4-Damage to compressor	1-Pressure safety valve at discharge, Operator action & alarm 2-Control system, Operator action & alarm 3-Operator action & alarm, compressor trip 4-Compressor drive protection, operator response
Storage vessel	Low pressure	Implosion of vessel due to vacuum	Vent line, operator action & alarm, compressor trip

Table 5.1: Most important abnormalities in the VRU's operation, their corresponding consequences, and safety barriers.

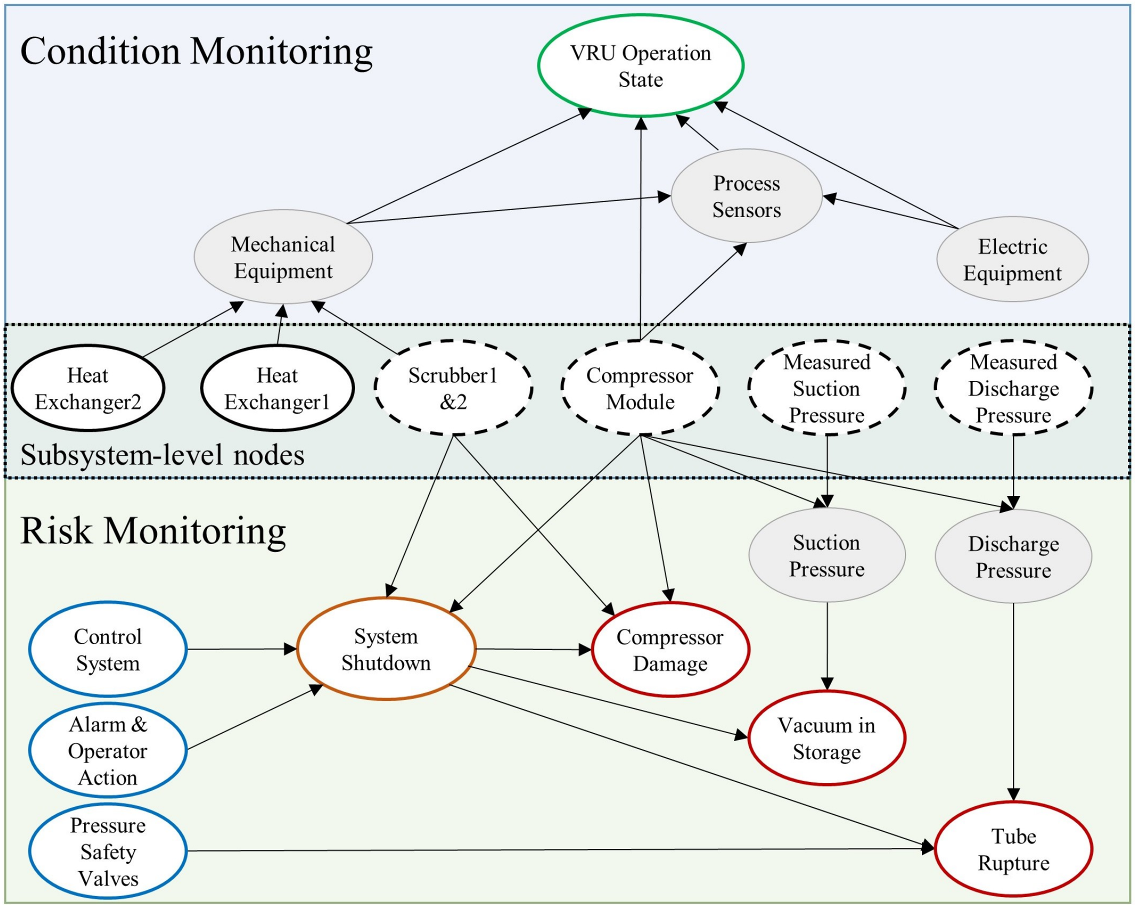

Figure 5.6: The VRU's developed BN for condition and operation risk monitoring (not including the monitoring sensors for each subsystem). The nodes in the middle are called "subsystem-level nodes" which are then used to dynamically assess the condition and operation risk of the system. The gray nodes are all auxiliary and used to integrate different sources of information and implement dependencies and conditions. The system-level condition monitoring node is indicated with green outline, the safety barriers with blue outline, and consequences with orange (less severe) and red outlines.

discharge pressure nodes are quantified based on the sensor measurements and a simple limit checking to classify the sensor values as either normal or high. The reason for incorporating the compressors suction and discharge pressures among the subsystem-level nodes is that they allow us to monitor two of the most important identified abnormalities with potentially severe consequences. The other two nodes that represent the condition of the two heat exchangers are static and assumed to be "normal" since there was no recorded failures for these subsystems in the available operation data.

These subsystem-level nodes are then used for both system-level condition monitoring, the top section of the BN model in Figure 5.6, and operation risk monitoring, the bottom section.

In the top section, the subsystems are categorized into the three main groups of machinery, mechanical and electrical equipment. This categorization is based on the offshore reliability data handbook (OREDA) [215]. Since any failure in subsystems results in VRU's abnormal state, the CPTs are quantified such that they represent AND/OR gates. Due to lack of failure and maintenance data for all individual subsystems, and the fact that the main contribution of this study is to demonstrate the viability of the integration of DL and BN rather than building an exact model for a VRU system, the electric motor and the two compressors are considered in one node in the final analysis and are called the "Compressor Module". To account for the errors that may happen in sensor readings, the "Process Sensors" node is considered in the BN. Once again, the OREDA handbook is used to obtain sensor failure probabilities for each category of equipment. Ultimately, all the nodes in the condition monitoring section are connected to the "VRU Operation State" node which estimates the system-level condition at each time step.

In the bottom section, the subsystem-level nodes are used as indicators of abnormalities which can potentially become adverse initiating events. These dynamic indicators enable monitoring the estimated probability of occurrence of different consequences. The risk monitoring section of the BN includes nodes that represent safety barriers in the system (blue outline nodes) and probability of the different consequences (orange and red outline nodes). According to the literature [216], in the VRU system, the safety barriers' main mitigative action is to trip the compressors and shut the system down. However, a system shutdown can be costly and this is why it is considered as a consequence, although less severe, in our model (orange outline). If the safety barriers fail to shut down the system, then more severe consequences such as compressor damage, vacuum in the storage vessel, and/or tube rupture may occur (red outline).

To take into account the impact of safety barriers, the success probability of the control system in general, and the pressure safety valves are obtained from the OREDA handbook. Due to lack of information, the success probability of the alarm and operator action is assumed to be 0.8. Also, using a number of auxiliary nodes may be required in the BN, as identified with gray color in Figure 5.6. For example, two auxiliary nodes are used in the risk monitoring section to condition abnormal suction and discharge pressures on the state of the compressor module. In other words, the abnormal state of the compressor module increases the probability that the abnormal measured suction and discharge pressures actually result in consequences.

5.4.3 Condition Monitoring Data Preprocessing

As the first step of the architecture, a thorough preprocessing of the data is performed to identify the most relevant sensor measurements regarding the detection of faults and anomalies in the system. The original dataset comprises of 188 sensors with data logs registered every 15 seconds over one year of system operation. The sensor network monitors a variety of measurands such as: temperature, vibration, flux, pressure, valve opening, and scrubber liquid level, among others. The layout of the most relevant sensors is shown in Figure 5.5, indicated as light blue circles before and after the system's main components. For sensor reliability purposes, the sensor network is composed of multiple redundant sensors, and thus many variables are highly correlated. The collected data comes from a real-world operational environment, thus noise contamination and missing data logs exist in the dataset. As such, to train a DL model, it is necessary to clean and prepare the data. Then, health state labels are generated to train a supervised model.

5.4.3.1 Feature Reduction and Selection

Since the sensor monitoring data will be used as the input to a DL model, feature reduction is an important step in the data preprocessing. A smaller number of features will reduce the number of parameters of the DL model, thus reducing training and evaluation times. Further, it is important to discard redundant variables, as well as features with low variability in time. This reduces the bias of the trained model and the input noise contamination. Equally important is to identify features that are irrelevant to the system's operation, which are commonly found in real-world datasets, such as sensors with a large number of missing logs.

Feature selection and reduction is carried out through three different methods.

First, engineering knowledge from the company's field engineers is used to identify and... remove variables unrelated to the system's operation from the training dataset (33 sensors are removed). Second, variables presenting 50% or more missing data entries or non-numerical values (i.e., NaN values) are discarded (27 additional sensors are removed). Finally, a statistical analysis is performed to compute the correlation between variables, as well as their individual temporal variation. In the former, sensors presented an absolute Pearson correlation value of 0.9 or higher were discarded, since they are considered to provide redundant information. In the latter, the Coefficient of Variability (CV), defined as the ratio of standard deviation to the mean, is used to determine features with a low temporal variability. Features with $CV < 0.05$ were removed from the dataset. The statistical analysis resulted in the removal of another 62 sensors, leaving 66 informative condition monitoring variables from the original 188.

5.4.3.2 Label Generation

When analyzing sensor monitoring data through supervised classification models, a common practice is to use Health Indexes (HIs) or Key Performance Indexes (KPIs) to define thresholds and create health state labels (e.g., healthy and unhealthy) [153]. In the VRU case study, due to the system's complexity, there are no HIs or KPIs that can indicate the start of the system's degradation towards failure. Also, no specific information on how to assess the health state of each of the subsystem is provided. Considering the common data collection practices in industry, not having defined HIs or KPIs is likely to be the case for many real complex systems. A reasonable alternative is to label the monitoring data based on operation time and registered failure times. Using the information from the maintenance logs, which contains the failure times and failure modes of the component that provoked the stoppage of the system, the system's degraded state is defined as the period immediately before each failure event. Labels created through this approach can be used to determine the healthy or abnormal state of the subsystems and the system as a whole.

The effectiveness of the proposed label generation approach relies on the ability to determine how long before a failure can the system be considered as degraded. From a practical point of view, field engineers would like to detect a degraded state at the early stages of the degradation process, which would mean setting the start of degradation days (or even weeks) before the failure event. However, the longer the time before a failure is considered as the degradation initiation point, the more likely it is to mislabel healthy states. Mislabeling healthy states can have a negative impact on the model's performance and is likely to increase false negative and false positive predictions. As such, a relatively small time-window of two hours before the failure events is considered to represent the degraded states. A three-hour transition state between healthy and degraded states is also defined to minimize the probability of the two classes overlapping. The degradation time-window is enough for engineers to react preventively to the failure events and ensure that all the points contained correspond to the degraded state. An example of this approach is shown in Figure 5.7, where the temporal values of a suction pressure sensor are shown. The three defined states (i.e., healthy, transition, and degraded) are shown in blue, yellow, and red, respectively.

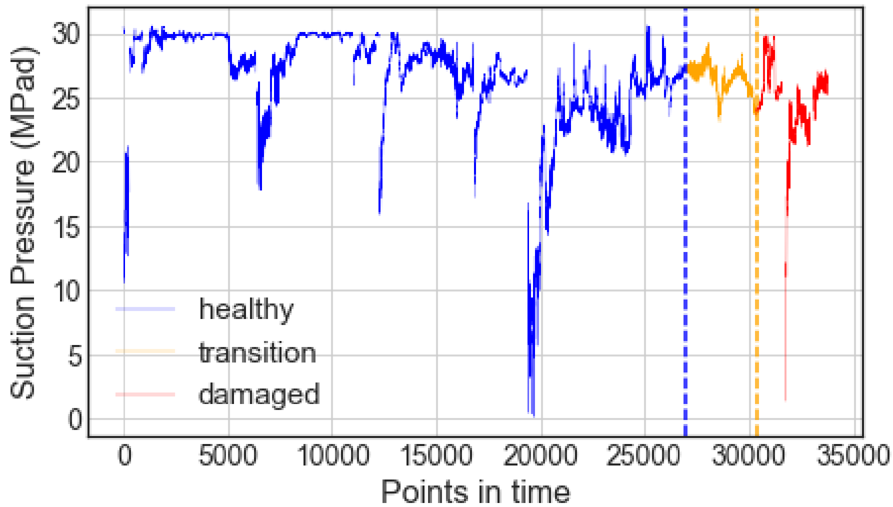

Figure 5.7: Example of health state time-windows on suction pressure measurements.

5.4.3.3 Outlier Detection and Removal Based on Temporal Analysis

Detecting and removing outlier points can improve the representativeness of the models for reasons similar to the ones presented for the feature selection. Commonly, when analyzing condition monitoring data, data points that lie outside of a determined confidence interval (normally 95th percentile) are considered to be outliers. This approach has two major drawbacks. First, it assumes that all sensor measurement values follow a normal distribution. Second, it completely ignores the temporal nature of the data. Hence, a better approach should consider both the temporal behavior of the measurements and statistical metrics to find and discard outlier points. This approach is more time consuming, since each sensor variable needs to be analyzed individually. It also requires determining the thresholds for

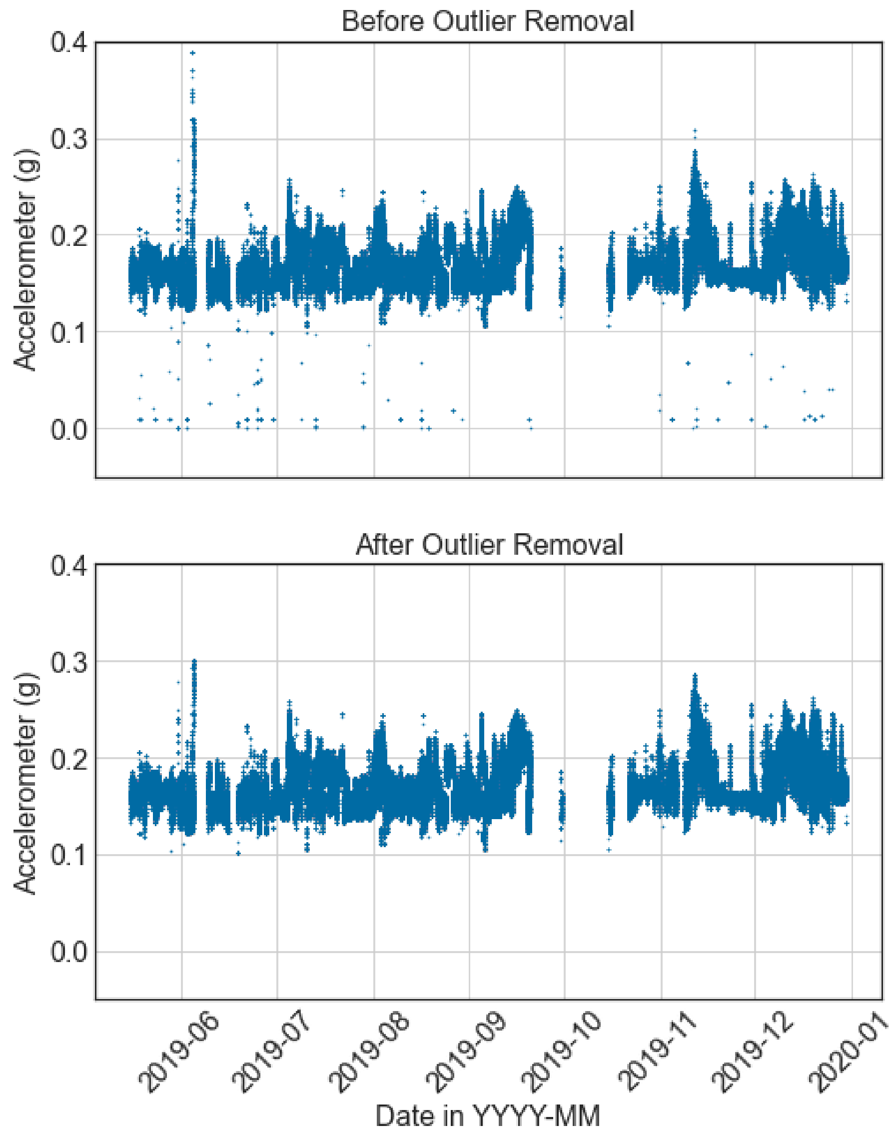

Figure 5.8: Example of outlier removal based on temporal and statistical analysis for healthy labeled data.

points to be considered as outliers, and thus engineering knowledge can bring important insights for this decision. Nevertheless, the resulting dataset using both statistical and temporal metrics, is more representative of the actual system's operation which will result in a more robust model.

An example of this approach is given in Figures 5.8, which shows the temporal behavior of an accelerometer (measures the vibration) before (above) and after (be-

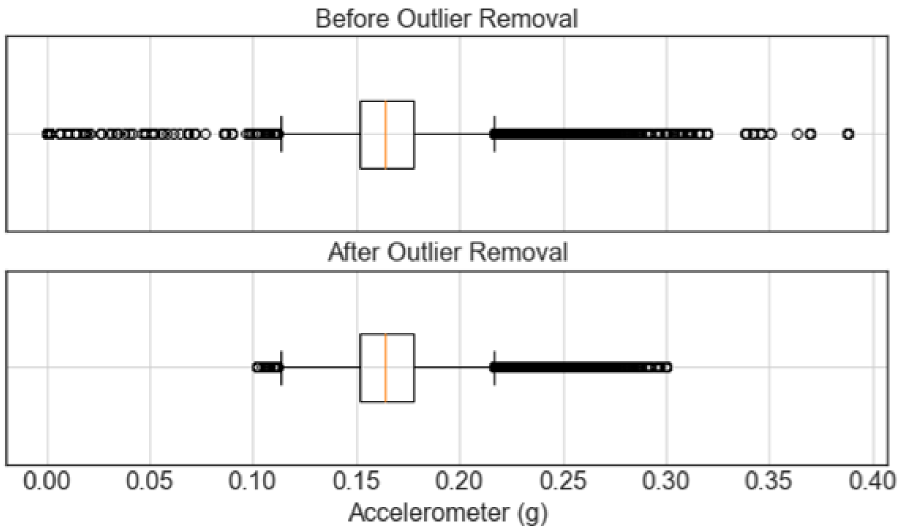

Figure 5.9: Example of 95% confidence interval for sensor data before and after outlier removal.

low) the outlier detection process. Figure 5.9 also shows the corresponding box plot at a 95% confidence interval. It can be observed that if only a statistical analysis was considered for the outlier detection (i.e., dots on Figure 5.9), several normal operational values would be discarded that do not correspond to outliers. For instance, Figure 5.9 suggests that any value above 0.25 g should be considered as an outlier. However, we can see that these values are part of the normal operation in Figure 5.8.

Since the DL model is expected to detect both abnormal and faulty states, the outlier detection methodology is applied only to the data labeled as healthy. This approach will also make the model's predictions more conservative and risk-averse, since it is less likely for the model to yield false negative values, i.e., classify a degraded state as healthy.

5.4.4 Developing Subsystems Condition Monitoring Models

One of the most important steps when training DL models is the selection of the model's hyperparameters. Hyperparameters are non-trainable parameters that need to be set by the user, and cannot be learned from the data. In BNNs, some of the most important hyperparameters are the number of layers, the number of neurons (outputs) per layer, the activation function between layers, and the learning rate used during the training process. In this regard, k-fold cross-validation and grid search are common approaches to find the optimal hyperparameters. Both approaches require defining a hyperparameter space with the possible values for each hyperparameter of the model. This can result in thousands, if not millions of combinations, making it unfeasible to train and compare every possible combination. It has been shown that performing a random search over the hyperparameter space significantly increases the convergence to the best set of hyperparameters [217]. Stochastic grid search randomly samples hyperparameters uniformly in the defined space.

Therefore, stochastic grid search is performed to find the best hyperparameters of the BNN model. For simplicity, all layers are considered to have the same number of neurons. The lower and upper bounds of the hyperparameters' search space is presented in Table 5.2. The activation function between layers is also randomly sampled between hyperbolic tangent (tanh), rectifier linear unit (ReLU), and sigmoid functions. The stochastic grid search is applied for the scrubbers and the compressor module separately. Table 5.2 shows the selected hyperparameters for each model, where the sigmoid activation function showed the best results.

Hyper-parameter	**Lower Bound**	**Upper Bound**	**Selected Scrubber**	**Selected Compressor**
Layers	2	10	3	3
Neurons	3	20	5	5
Learning rate	10^{-4}	10^{-1}	10^{-2}	10^{-2}
Batch size	2	2^7	64	8
Epochs	10	500	50	100

Table 5.2: Hyperparameters' search space lower and upper bounds, and the corresponding selected hyperparameters values for the Scrubbers and Compressor module models.

The training curves for the scrubbers BNN models with the selected hyperparameters are shown in Figure 5.10. It can be observed that the models take around 20 epochs to start learning from the data and improve its training metrics. We associate this behavior to the several stochastic elements involved in the training of the model. Indeed, all weights in the BNN have their own distributions that need to be adjusted, while models are trained with the Adam variation of stochastic gradient descent (SGD). Thus, at the beginning of the training process, several training steps are likely to be inefficient with stagnant loss and accuracy values.

Table 5.3 presents the loss and accuracy metrics for the Scrubber and Compressor BNNs, respectively. In both cases, models show negligible overfitting, with accuracy values above 95% for the training, validation, and test sets.

All DL models are trained in Python 3.7 using TensorFlow and Tensorflow-Probability 2.1. Windows 10 is used as the operating system. The hardware is equipped with an Intel i7 9700k CPU, a 24GB Titan RTX GPU, and 32GB of DDR4 RAM.

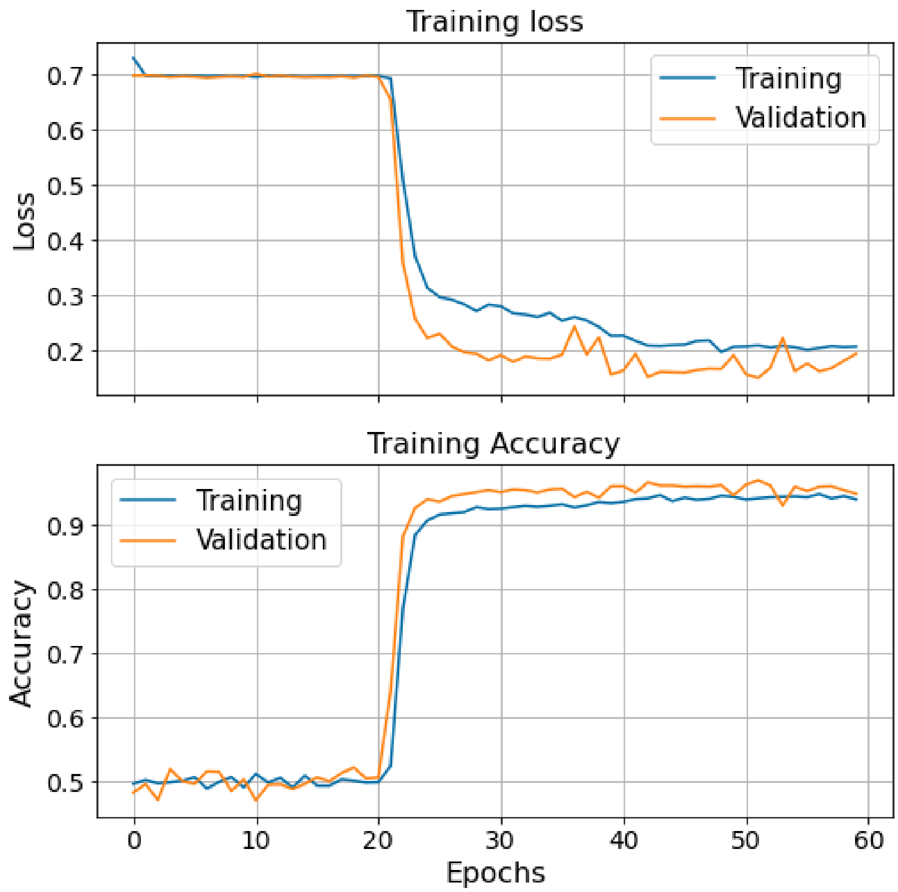

Figure 5.10: Loss and accuracy values for the train and validation sets during the training process of the Scrubbers BNN.

	Scrubber		**Compressor**	
	Loss	**Accuracy**	**Loss**	**Accuracy**
Training	0.1859	95.83%	0.1679	98.03%
Validation	0.1879	95.89%	0.1881	96.80%
Test	0.2134	95.12%	0.1284	98.40%

Table 5.3: Training, validation, and test results for the trained Scrubber and Compressor BNN, respectively.

5.4.5 Full Model Implementation and Results

To demonstrate the performance of our proposed approach, a snapshot of the system operation with the length of 1900 time steps is considered. The outputs of

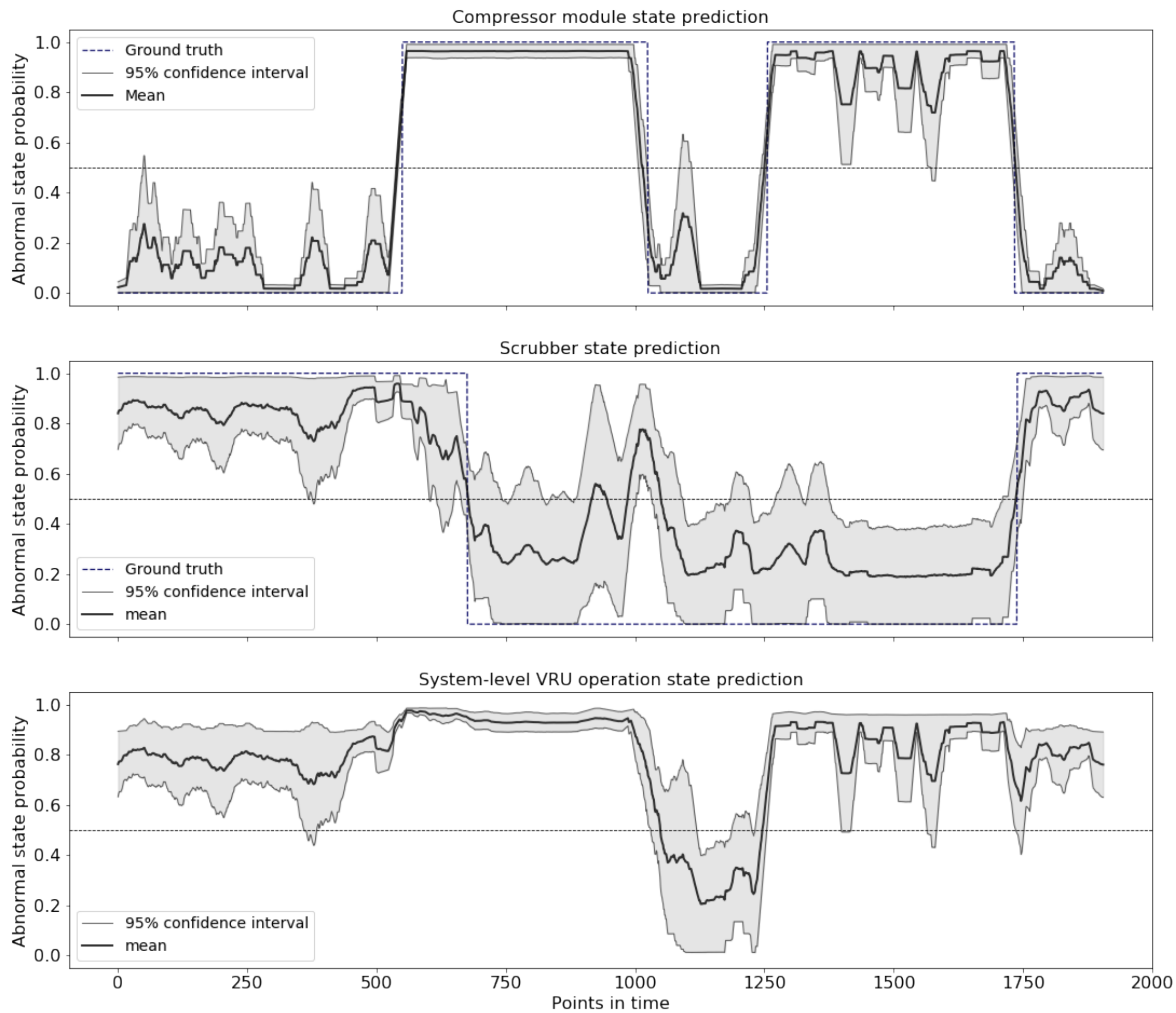

Figure 5.11: Top: a 1900 time step snapshot of the "Compressor Module" condition monitoring, while it goes through two abnormal phases. Middle: a 1900 time step snapshot of the "Scrubber 1&2" condition monitoring, while it transitions from an abnormal state to normal and again abnormal. Bottom: the system-level "VRU Operation State" monitoring.

the trained BNNs for condition monitoring of the "Compressor Module" and the "Scrubber 1&2" are generated. The prediction snapshots for the mentioned two subsystems are selected such that they show the transition in the BNNs prediction from normal to abnormal states and they are not synchronous. As explained in Section 2.2.2.2, BNNs allow uncertainty quantification, therefore the 95% credibility interval for prediction at each time step, along with the true states, are demonstrated

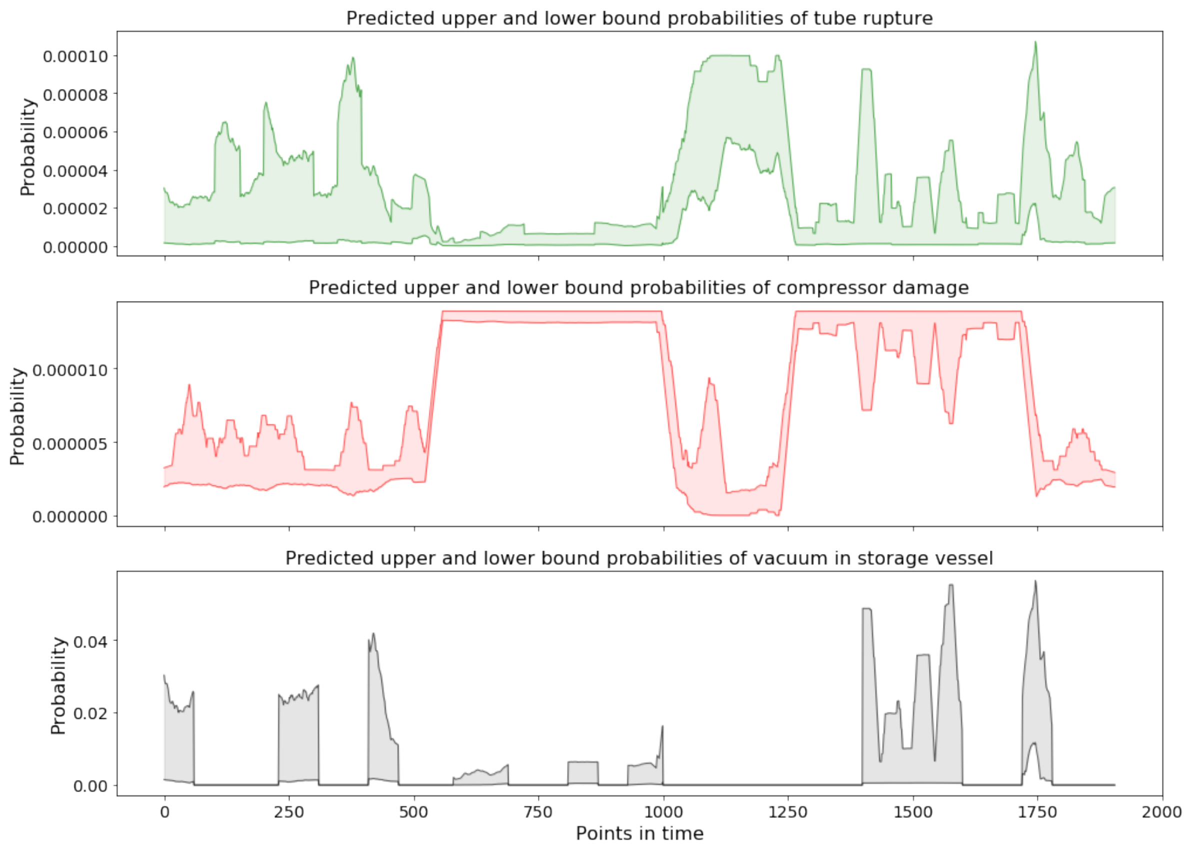

Figure 5.12: Top: the estimated upper and lower bound probabilities of occurrence of tube rupture over time. Middle: the estimated upper and lower bound probabilities of occurrence of compressor damage over time. Bottom: the estimated upper and lower bound probabilities of occurrence of vacuum in the storage vessel that is connected to the VRU system.

for the two subsystems in Figure 5.11. Also, the value of the top node of the BN, "VRU Operation State," at each time step is calculated by propagating the upper and lower bound predictions for the subsystems' states that enables a system-level condition monitoring with uncertainty quantification (Figure 5.11).

Assuming a threshold of 0.5 for determining the state of the subsystems, the trained BNNs are highly accurate in predicting their states (refer to Table 5.3). Looking at the outcome of the BNNs along with the ground truth and the decision threshold, shows that in majority of occasions even the upper and lower bounds of the predictions' credibility intervals fall either below or above the threshold which makes it convenient for a decision maker to infer and decide upon. Considering the design and operation logic of the VRU system, whenever a subsystem is abnormal, the whole system will be in an abnormal state and corrective and/or mitigative measures should take place. The bottom plot in Figure 5.11 demonstrates this behavior at the system-level. As a clarification, in this specific monitoring snapshot, the system is mostly in an abnormal state. This does not mean that the VRU system is unreliable and the reason for this observation is that the subsystem-level monitoring snapshots are not synchronous (explained earlier in this section).

As far as the operation risk monitoring of the system is concerned, with this architecture continuous assessment of the different consequences' probabilities is possible. Since the safety barriers have a high success probability and shut the system down if an abnormal operation is detected, the "System Shutdown" node values are almost identical with the "VRU Operation State" values, therefore we do not include its plot. For the more severe consequences, the monitoring results are presented in Figure 5.12.

High suction and high discharge pressures are preconditions for "Vacuum in Storage" and "Tube Rupture" consequences respectively. In addition, the BNNs are not able to separate different failure modes due to lack of suitable training data and they can only detect abnormal operation. Therefore, to demonstrate how incorporating a number of key sensor measurements can enable identification of different failure modes, the measured suction and discharge pressures are augmented to have a number of abnormal values. The lower plot presented in Figure 5.12 which is the result of monitoring the probability of having a vacuum in the storage vessel shows the impact of the introduced abnormal suction and discharge pressures. This probability is zero unless the suction pressure is abnormal.

5.5 Discussion

PHM and PRA are two of the most important sub-fields of reliability engineering that are being developed in an independent manner, each having their own advantages and limitations. The proposed approach in [169] utilizes the complementary characteristics of these two sub-fields and introduces a novel framework that enables system-level condition monitoring and prognostics, and system operation risk monitoring for CES.

We operationalized this framework by proposing a mathematical architecture that uses BNs combined with DL techniques. State-of-the-art BNNs are used to construct subsystems condition monitoring models that provide a probabilistic measure of operation condition of the subsystems and allows uncertainty quantification. The developed subsystems condition monitoring models are then used to analyze the abundant streaming condition monitoring data and continuously update the overall model, enabling a dynamic assessment.

We used BN to model the CES and the scenarios that can lead to an adverse outcome in the system. We demonstrated how the subsystems and system-level condition monitoring results, along with estimated probabilities of occurrence for different risk-relevant consequences, can be presented as the outcome one single architecture. This underlines the potential of the proposed algorithm to be a powerful decision-making support tool in the context of risk-informed PHM and a data-driven PRA. It was shown that various sources of information such as abundant multi-dimensional monitoring data, the role of the human operator and their input, expert opinion, system's functional and failure logic, and various dependencies among subsystems can be incorporated in one model to generate useful, dynamic insights for the decision-makers.

To the best of the authors' knowledge, this is the first architecture that integrates BN and DL for both condition and risk monitoring in reliability engineering. Furthermore, BNNs' ability to quantify their prediction uncertainty is used to generate a credibility interval for subsystems' state predictions which is then propagated through the BN and taken into account for system-level predictions. This is one of the first studies in our field that considers uncertainty quantification in online monitoring of systems using deep learning algorithms.

This study is an important step towards a comprehensive, data-informed risk and reliability analysis framework for CES. Provided enough information and operation data, this architecture can be applied on CES with a larger number of subsystems with more complex interactions among them.

It is anticipated that with very large BNs, computational speed for online monitoring of the system become a problem. In those cases, more efficient BN designs and inference algorithms should be utilized or developed as discussed in Section 5.2.1.

Also, as far as the consequence analysis is concerned, the probability of occurrence values presented in Figure 5.12, should be multiplied by their induced outcomes in terms of costs, loss of lives, etc. to provide a complete risk picture. This process is system-dependent and requires expert knowledge. Further research on the interpretation of these probabilities and ways to make this process less system-dependent is another direction that would help improve the current mathematical architecture.

In demonstrating the applicability of the proposed architecture in this chapter, we have modeled component functional interdependencies, but the software and human elements are not incorporated explicitly, nor are emergent behaviors or other sources of dependency. However, BNs are capable of modeling human operators and incorporating software elements and these may also manifest as patterns in the system data used in deep learning. Likewise, BNs allow explicit modeling of common cause failures and other sorts of dependencies. Regarding the emergent behaviors, the deep learning models can be used to detect new patterns in the system and at the very least label these new pattern as an emergent behavior which needs to be studied further. Full incorporation of these elements can be among the important next steps of this research.

5.6 Conclusion

The realization of the proposed concept of integrating the advancements in PHM and PRA is pursued in this Chapter. A novel mathematical architecture for condition and operation risk monitoring of CES by integrating BN and DL is proposed. This architecture enables operation data-driven PRA and risk-informed PHM simultaneously for CES. Moreover, it provides a holistic view on different aspects of a CES operation and supports decision-making. The architecture is successfully implemented on a real-world complicated system, which serves as the first example of a condition and operation risk monitoring model that provides a powerful decision-making support tool. The results are promising and the architecture could be implemented on different systems given enough operation, maintenance, and system design information.

5.7 Appendix

The conditional probability tables used in the VRU system's BN are presented in this appendix. The nodes which are left out are the "subsystem-level nodes" since they are quantified in time using condition monitoring algorithms.

Heat exchanger 2	Normal				Abnormal			
Heat exchanger 1	Normal		Abnormal		Normal		Abnormal	
Scrubber	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal
Normal	0.99	0.05	0.05	0.01	0.05	0.01	0.01	0
Abnormal	0.01	0.95	0.95	0.99	0.95	0.99	0.99	1

Table 5.4: "Mechanical Equipment" Node CPT used in the systems' BN.

Alarm & Operator	
Success	0.8
Failure	0.2

Control System	
Success	0.99993
Failure	6.99e-05

Pressure safety valve	
Functional	0.999
Not functional	0.001

Table 5.5: Safety Barriers' probability of success in shutting down the system. The Control system and the Pressure safety valve success probabilities are determined using the OREDA handbook [215], the alarm and operator action success probability is assumed to be 0.8.

Electric Equipment	Normal				Abnormal			
Mechanical Equipment	Normal		Abnormal		Normal		Abnormal	
Compressor Module	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal
Normal	1	0.9999	0.999	0.999	0.999	0.999	0.999	0.999
Abnormal	0	0.0001	1.399e-05	1.399e-05	3.239e-06	3.239e-06	1.722e-05	1.722e-05

Table 5.6: Process sensors node CPT, values obtained from OREDA handbook [215].

Compressor Module	Normal							
Mechanical Equipment	Normal				Abnormal			
Electric Equipment	Normal		Abnormal		Normal		Abnormal	
Process Sensors	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal
Normal	1	0.95	0.05	0.05	0.05	0.05	0.01	0
Abnormal	0	0.05	0.95	0.95	0.95	0.95	0.99	1

Table 5.7: System-level condition monitoring node, "VRU abnormal operation" CPT, part 1.

Compressor Module	Abnormal							
Mechanical Equipment	Normal				Abnormal			
Electric Equipment	Normal		Abnormal		Normal		Abnormal	
Process Sensors	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal
Normal	0.05	0.05	0.01	0	0.01	0.01	0	0
Abnormal	0.95	0.95	0.99	1	0.99	0.99	1	1

Table 5.8: System-level condition monitoring node, "VRU abnormal operation" CPT, part 2.

Compressor_Module	Normal				Abnormal			
Scrubber	Normal		Abnormal		Normal		Abnormal	
System Shutdown	Yes	No	Yes	No	Yes	No	Yes	No
Damage	0	0	0	0.2	0	1	0	1
No Damage	1	1	1	0.8	1	0	1	0

Table 5.9: "Compressor Damage" node CPT.

Suction Pressure	Normal_Suction		High_Suction	
System Shutdown	Yes	No	Yes	No
Vacuum	0	0	0	1
No Vacuum	1	1	1	0

Table 5.10: "Vaccum in storage vessel" node CPT.

Discharge Pressure	Normal Discharge Pressure				High Discharge Pressure			
System Shutdown	Yes		No		Yes		No	
Pressure Safety Valve	Func.	Not Func.	Func.	Not Func.	Func.	Not Func.	Func.	Not Func.
Rupture	0	0	0	0.1	0	0	0	1
No Rupture	1	1	1	0.9	1	1	1	0

Table 5.11: "Tube rupture node" CPT. Func. stands for Functional.

Chapter 6: Conclusion

6.1 Main Conclusions

This dissertation presents a novel framework for condition and operation risk monitoring of Complex Engineering Systems (CES) by drawing concepts and techniques from two of the major subfields of reliability engineering, namely Prognostics and Health Management (PHM) and Probabilistic Risk Assessment (PRA). This research bridges the gap between PRA and PHM and establishes a new risk and reliability analysis modeling paradigm. To do so, a thorough literature review is performed to identify the gaps and research needs in the context of CES. Of this review, the major needs that emerged as most pressing are 1) ability to perform dynamic or online risk and reliability assessment, 2) providing both a subsystem-level and system-level insight (multi-level analysis), 3) fusing the analysis results of numerous heterogeneous elements of a CES, and 4) using all available sources of information. These pressing research needs have shaped the main objectives of this dissertation.

To address the mentioned research needs, PRA and PHM characteristics are critically assessed and unified definitions and concepts of PRA and PHM are presented. Further, the steps and tasks involved in PRA and PHM are disintegrated and reassembled to construct a conceptual framework for condition and risk monitoring of CES. This framework has two major parts, one is the "framework construction" for each specific system, and the other is the "framework application" on the CES and for supporting a desired objective. The framework is capable of supporting both PHM and PRA objectives, such as dynamic risk monitoring and accident prevention, decision support for risk-informed health management, or both. Another important feature in the proposed framework is the clear distinction between system-level analysis and assessment supported by the PRA techniques and subsystem-level analysis and assessment supported by the PHM techniques.

This Systematic Integration of PHM and PRA (SIPPRA) framework serves as a roadmap for developing detailed numerical and mathematical architectures that would operationalize it. In developing SIPPRA, we have assumed that sufficient data is available for the CES under study. Data can be real or the result of accurate simulations. There are existing issues with real data availability and accurate simulation of CES. However, addressing these issues are complementary to this research and SIPPRA motivates data collection and more accurate simulations.

As the first step towards realization of SIPPRA, a mathematical architecture is developed for system operation condition/health monitoring for complicated systems. This framework is able to perform an end-to-end condition monitoring, meaning that the streaming monitoring data is given to the architecture as the input and system-level and subsystem-level health state estimations are computed as the output. This architecture integrates Fault Tree as the system modeling method and deep learning as the subsystems condition monitoring method. To demonstrate the applicability of this architecture, a number of different deep learning models are trained using operation and maintenance data for the independent subsystems of a mining stone crusher system. The Fault Tree then fuses the subsystem-level assessment results to provide a system-level measure of the mining stone crusher's condition. This architecture reduces the computational expense of using only logic-based models (such as the FT) in a dynamic manner, however, the scalability limitations should be identified and addressed to enable extending its application to CES. Also, FT has limitations in dependency modeling, forward and backward evidence propagation, uncertainty quantification, and incorporating expert opinion and qualitative measures.

To further develop SIPPRA, we developed an enhanced mathematical architecture applicable to CES. This architecture is designed to perform condition monitoring and operation risk monitoring simultaneously on systems with multiple interdependent subsystems, components, and data streams. This architecture uses Bayesian networks as the system modeling tool due to several advantages that these networks offer relative to other system modeling methods such as Fault Trees (BNs can replicate any FT and offer more advantages). Similar to the previous architecture, deep learning is used to develop condition monitoring models for the subsystems. To show the practicality of this architecture, a case study on a real-world vapor recovery unit is successfully performed. In this case study, Bayesian neural networks are used and trained for condition monitoring of key subsystems to enable uncertainty quantification and propagation. This uncertainty handling makes this architecture more suitable for risk assessment purposes. Further, use of BNs allows system-level prognostics and dynamic subsystems dependencies can be modeled using dynamic Bayesian networks and evidences can be propagated to different time steps.

Subsystem-level prognostics is possible in both of the developed architectures using predictive DL models. However, the latter architecture based on BNs and deep learning enables diagnostics and prognostics to be conducted not just only at the subsystem-level but at the system-level as well. To reason about the subsystems and the whole system in one architecture offers an important advancement to the field of reliability engineering.

As mentioned in the first chapter, case studies on actual complex engineering systems and addressing all of their characteristics are a next research step beyond this dissertation. We have addressed fusion of various data sources, analysis of heterogeneous elements, multi-level analysis and assessment, and partially considered uncertainty quantification in our case studies. However, the rest of the CES characteristics are not considered and have to be explored in future studies.

6.2 Contributions

We extensively reviewed the state of the art PRA and PHM literature and identified the gaps relevant to CES. This review suggested that the integration of PHM and PRA models and techniques has great unexplored potential. The contributions made in this research are discussed with respect to each of three main objectives of this dissertation:

O1: Develop and establish a unifying conceptual and mathematical framework for connecting PHM data and algorithms to PRA models and decisions in the context of complex engineering systems.

Contributions: a novel conceptual framework (SIPPRA) for risk and reliability analysis of complex engineering systems is developed.

1. Provided a unified definition for PRA and PHM steps and practices.

2. Systematically deconstructed and reassembled PHM and PRA steps and defined a comprehensive, dynamic, and multi-level risk and reliability analysis framework (SIPPRA) as a roadmap for developing a new generation of mathematical architectures.

3. Established a guideline on how to incorporate various sources of information and data such as expert opinion, system design, recorded condition monitoring data, and maintenance and failure records into the SIPPRA modeling paradigm.

O2: Develop a condition monitoring mathematical architecture for complicated systems that is capable of performing dynamic condition monitoring and providing multi-level (system-level and subsystem-level) measures of operation condition.

Contributions: developed a mathematical architecture for complicated systems condition monitoring as the first step towards complete realization of SIPPRA.

1. Developed a multi-level (subsystem-level and system-level) online condition monitoring mathematical architecture. We used FT (commonly used in PRA)

to model the system and fuse subsystem-level information. Further, developed subsystems' DL-based condition monitoring models and integrated them into the FT to enable online condition monitoring.

2. Validated this architecture and demonstrated its applicability by applying it on a real-world mining stone crusher system with 5 independent subsystems and 21 monitoring sensors.

O3: Develop an enhanced mathematical architecture that in addition to the target features of Objective 2, is capable of monitoring the complex engineering systems operation condition and risk simultaneously. This algorithm should be able to integrate all available sources of information, enable diagnosis and prognosis, and model interdependencies among subsystems.

Contributions: through this work, the realization of the full SIPPRA mathematical architecture for complex engineering system operation condition and risk monitoring is developed.

1. Developed a novel condition and risk monitoring architecture that is capable of multi-level system analysis, dynamic risk assessment. We used BNs for system-level modeling that allows modeling subsystems' complicated relationships, using all available sources of information (including qualitative information), and enables system-level prognostics. Further, probabilistic DL models are developed for condition monitoring of key subsystems and enable uncertainty quantification.

2. Demonstrated how both the subsystems and system-level condition monitor-

ing results and estimated probabilities of occurrence for different risk-relevant consequences, can be achieved in one single architecture.

3. Introduced probabilistic deep learning to the field of engineering systems dynamic risk assessment.

4. Validated this architecture and demonstrated its applicability by applying it on a real-world vapor recovery unit used in oil and gas production facilities with 8 subsystems and 188 monitoring sensors.

6.3 Publications, Presentations, & Awards

Journal Papers

Included in this dissertation:

- Moradi, Ramin, Ruiz-Tagle Palazuelos, Andrés, Lopez Droguett, Enrique, and Katrina Groth. "Towards a Framework for Risk Monitoring of Complex Engineering Systems with Online Operation Data: A Deep Learning Based Solution". To be submitted to *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* (Invited paper, submitted on June 15, 2021 for special invited issue).

- Moradi, Ramin, Cofre-Martel, Sergio, Lopez Droguett, Enrique, Modarres, Mohammad, and Katrina Groth. "Integration of Deep Learning and Bayesian networks for condition and operation risk monitoring of complex engineering systems". Submitted May 2021 to *Reliability Engineering & System Safety* .

- Ramin Moradi and Katrina M Groth. “Modernizing risk assessment: A systematic integration of PRA and PHM techniques”. In: *Reliability Engineering & System Safety* 204 (2020), p. 107194

Others:

- Ramin Moradi and Katrina M Groth. “Hydrogen storage and delivery: Review of the state of the art technologies and risk and reliability analysis”. In: *International Journal of Hydrogen Energy* 44.23 (2019), pp. 12254–12269
- Katrina M Groth, Reuel Smith, and Ramin Moradi. “A hybrid algorithm for developing third generation HRA methods using simulator data, causal models, and cognitive science”. In: *Reliability Engineering & System Safety* 191 (2019), p. 106507

Conference Papers:

Included in this dissertation:

- Ramin Moradi, Andrés Ruiz-Tagle Palazuelos, Enrique López Droguett, et al. “Towards a Framework for Risk Monitoring of Complex Engineering Systems with Online Operation Data: A Deep Learning Based Solution”. In: *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference*. 2020

Others:

- Karimian, Fouad, Moradi, Ramin, Cofre-Martel, Sergio, Groth, Katrina, and Mohammad Modarres. "Neural Network and Particle Filtering: A Hybrid

Framework for Crack Propagation Prediction". In: *Proceedings of the Structural Health Monitoring & Non-Destructive Testing (SHM-NDT)*. November 2020.

- Ramin Moradi and Katrina Groth. "On the application of transfer learning in prognostics and health management". In: *Annual Conference of the PHM Society*. Vol. 12. 1. 2020, pp. 8–8

Presented at:

- "Methodology for System-level Risk Monitoring Using Simulated and Sensor Data". Society for Risk Analysis (SRA) Annual Meeting. December 2018, New Orleans, LA USA.

- "Neural Network and Particle Filtering: A Hybrid Framework for Crack Propagation Prediction". Annual Conference of the Prognostics and Health Management Society. Sept. 2019, Phoenix, AZ USA.

- "On the application of transfer learning in prognostics and health management". Annual Conference of the Prognostics and Health Management Society. Sept. 2020, Nashville, TN USA.

Awards:

- Best Paper Award at the 3rd "Structural Health Monitoring and Non-Destructive Testing" Conference, December 2020.

- Doctoral Symposium Best Presentation Award, Annual conference of the PHM Society, November 2020.

- Graduate School's Outstanding Research Assistant Award for Academic Year of 2018-19. Issued by the Graduate School, University of Maryland, December 2018.
- Applied Risk Management Specialty Group Student Merit Award. Society for Risk Analysis Annual Meeting, November 2018.
- Society for Risk Analysis (SRA) 2018 Annual Meeting Travel Award, Nov. 2018.

6.4 Future Research Directions

The diagnostics and condition and risk monitoring capabilities of the proposed algorithms were shown in this dissertation. Using the same architectures, prognostics is also possible. Therefore, further **exploration of prognostics using the introduced framework is suggested**. Prognostics can be explored using system modeling methods such as Dynamic Bayesian Networks (DBN), or developing predictive models for subsystems condition monitoring.

Research on the interpretation of the outcomes of the SIPPRA framework can be an impactful next step. As shown in Chapter 5, the developed architecture is able to estimate the probability of occurrence of adverse events and accidents. Yet, the interpretation of these probabilities and how they would translate into decisions requires further exploration. One approach, that was explored in this dissertation is using probabilistic deep learning which enables UQ in both diagnostics and prognostics. UQ naturally makes the predictions more reliable and to some extent, more interpretable. Yet, this may not be enough for safety critical systems. Further work on the interpretability of deep learning models and trustability of their outcomes is merited.

A characteristic feature of CES is having a large number of diverse components, variables, and data streams. The integration of DL models with system logic modeling methods partially addresses the challenge of computation cost and speed. However, the **scalability limits of the proposed architecture have to be determined and improved** without sacrificing interpretability. In particular, attention should be paid on the scalability challenges introduced by increasing the modeling of interdependency among subsystems, which manifests as large interconnected BNs and brings with it significant computational cost. One main approach to address the computational issues is **exploring additional algorithms rigorously to determine if they can improve upon the performance of SIPPRA**; some candidate approaches are fast inference algorithms for BNs, BNs structure learning algorithms to improve system modeling accuracy and speed, applicability of other graph theory models and analytical methods for streamlining system modeling.

To further improve the proposed architectures, **applying these algorithms on various systems is recommended**. Piloting these algorithms on different CES provides a wider set of case studies for vetting the algorithms, as well as validation of the algorithms, their use, and pathways for making risk-informed decisions a more integrated aspect of systems management.

The proposed architectures can be used to model and incorporate important CES elements such as humans and their behavior and inputs. Humans are an important part of risk analysis for many CES such as Nuclear power plants. Thus, **exploring the incorporation of human activities in the SIPPRA framework proposed in this dissertation is recommended**. This is a more rigorous incorporation of the human "component" than the current Human Reliability Analysis (HRA) approaches used in PRA alone. Also, incorporating more diverse information sources and data types, extending or validating the ability to explicitly model software and human elements, modeling complex dependencies, and having a larger number of uncertainty sources in the system are among the ways that these architectures can be further advanced.

Ultimately, after constructing such architectures for a system, there is a need to be able to assess the suitability and performance of them. Therefore, **metrics are needed to be developed to enable the comparison of alternative architectures.** These metrics should consider the balance between computational efficiency, accuracy, precision, and also consider the data requirements imposed by the algorithms. Also, characterizing the performance of models in a multidimensional way is needed, considering not just traditional metrics like computational performance, accuracy, and speed, but softer, qualitative metrics such as informativeness in a decision-making process, and comprehensiveness of models.

Bibliography

[1] Nicholas J Bahr. *System safety engineering and risk assessment: a practical approach*. CRC press, 2014.

[2] Mohammad Modarres. *Risk analysis in engineering: techniques, tools, and trends*. CRC press, 2006.

[3] Terje Aven. *Quantitative risk assessment: the scientific platform*. Cambridge University Press, 2011.

[4] Hans Pasman and Genserik Reniers. "Past, present and future of Quantitative Risk Assessment (QRA) and the incentive it obtained from Land-Use Planning (LUP)". In: *Journal of loss prevention in the process industries* 28 (2014), pp. 2–9.

[5] Mohammad Ali Farsi and Enrico Zio. "Industry 4.0: Some challenges and opportunities for Reliability Engineering". In: *International Journal of Reliability, Risk and Safety: Theory and Application* (2019).

[6] Kwok L Tsui, Nan Chen, Qiang Zhou, Yizhen Hai, and Wenbin Wang. "Prognostics and health management: A review on data driven approaches". In: *Mathematical Problems in Engineering* 2015 (2015).

[7] Vepa Atamuradov, Kamal Medjaher, Pierre Dersin, Benjamin Lamoureux, and Noureddine Zerhouni. "Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation". In: *International Journal of Prognostics and Health Management* 8.060 (2017), pp. 1–31.

[8] Liangwei Zhang, Jing Lin, Bin Liu, Zhicong Zhang, Xiaohui Yan, and Muheng Wei. "A review on deep learning applications in prognostics and health management". In: *IEEE Access* 7 (2019), pp. 162415–162438.

[9] David Benjamin Verstraete, Enrique López Droguett, Viviana Meruane, Mohammad Modarres, and Andrés Ferrada. "Deep semi-supervised generative adversarial fault diagnostics of rolling element bearings". In: *Structural Health Monitoring* 19.2 (2020), pp. 390–411.

[10] Dong Wang, Kwok-Leung Tsui, and Qiang Miao. "Prognostics and health management: A review of vibration based bearing and gear health indicators". In: *IEEE Access* 6 (2017), pp. 665–676.

[11] Huixing Meng and Yan-Fu Li. “A review on prognostics and health management (PHM) methods of lithium-ion batteries”. In: *Renewable and Sustainable Energy Reviews* 116 (2019), p. 109405.

[12] Jay Lee, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Linxia Liao, and David Siegel. “Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications”. In: *Mechanical systems and signal processing* 42.1-2 (2014), pp. 314–334.

[13] J Montero-Mayorga, C Queral, and J Gonzalez-Cadelo. “Effects of delayed RCP trip during SBLOCA in PWR”. In: *Annals of Nuclear Energy* 63 (2014), pp. 107–125.

[14] Pierre-Etienne Labeau. “Probabilistic dynamics: estimation of generalized unreliability through efficient Monte Carlo simulation”. In: *Annals of Nuclear Energy* 23.17 (1996), pp. 1355–1369.

[15] Marzio Marseguerra and Enrico Zio. “Monte Carlo approach to PSA for dynamic process systems”. In: *Reliability Engineering & System Safety* 52.3 (1996), pp. 227–241.

[16] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*. Vol. 10. John Wiley & Sons, 2016.

[17] MATSUOKA Takeshi. “A Monte Carlo simulation method for system reliability analysis”. In: *Nucl. Saf. Simul* 4.1 (2013), pp. 44–52.

[18] Katrina M. Groth, Matthew R. Denman, Thomas B. Jones, Michael C. Darling, and George F. Luger. “Building and using dynamic risk-informed diagnosis procedures for complex system accidents”. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 234.1 (2020), pp. 193–207. DOI: 10.1177/1748006X18803836.

[19] Gerardo Rubino, Bruno Tuffin, et al. *Rare event simulation using Monte Carlo methods*. Vol. 73. Wiley Online Library, 2009.

[20] James Bucklew. *Introduction to rare event simulation*. Springer Science & Business Media, 2013.

[21] Tunc Aldemir. “A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants”. In: *Annals of Nuclear Energy* 52 (Feb. 2013). Nuclear Reactor Safety Simulation and Uncertainty Analysis, pp. 113–124. ISSN: 0306-4549. DOI: 10.1016/j.anucene.2012.08.001.

[22] Zhegang Ma, Curtis Smith, and Steve Prescott. “A simulation-based dynamic approach for external flooding analysis in nuclear power plants”. In: *International Confernece Pacific Basin Nuclear Conference*. Springer. 2016, pp. 55–69.

[23] N Siu. “Risk assessment for dynamic systems: an overview”. In: *Reliability Engineering & System Safety* 43.1 (1994), pp. 43–73.

[24] D Mandelli, C Smith, C Rabiti, A Alfonsi, R Youngblood, V Pascucci, B Wang, D Maljovec, PT Bremer, T Aldemir, et al. "Dynamic PRA: an overview of new algorithms to generate, analyze and visualize data". In: *Proceeding of American Nuclear Society* (2013).

[25] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. "Visualizing high-dimensional data: Advances in the past decade". In: *IEEE transactions on visualization and computer graphics* 23.3 (2016), pp. 1249–1268.

[26] Diego Mandelli, Alper Yilmaz, Tunc Aldemir, Kyle Metzroth, and Richard Denning. "Scenario clustering and dynamic probabilistic risk assessment". In: *Reliability Engineering & System Safety* 115 (2013), pp. 146–160.

[27] Dan Maljovec, Bei Wang, Valerio Pascucci, Peer-Timo Bremer, Diego Mandelli, Michael Pernice, and Robert Nourgaliev. *Exploratory Nuclear Reactor Safety Analysis and Visualization via Integrated Topological and Geometric Techniques*. Tech. rep. Idaho National Lab.(INL), Idaho Falls, ID (United States), 2013.

[28] Pietro Turati, Nicola Pedroni, and Enrico Zio. "Advanced RESTART method for the estimation of the probability of failure of highly reliable hybrid dynamic systems". In: *Reliability Engineering & System Safety* 154 (2016), pp. 117–126.

[29] Pietro Turati, Nicola Pedroni, and Enrico Zio. "Simulation-based exploration of high-dimensional system models for identifying unexpected events". In: *Reliability Engineering & System Safety* 165 (2017), pp. 317–330.

[30] Francesco Cadini, Francisco Santos, and Enrico Zio. "An improved adaptive kriging-based importance technique for sampling multiple failure regions of low probability". In: *Reliability Engineering & System Safety* 131 (2014), pp. 109–117.

[31] Frédéric Cérou and Arnaud Guyader. "Adaptive multilevel splitting for rare event analysis". In: *Stochastic Analysis and Applications* 25.2 (2007), pp. 417–443.

[32] Benjamin Echard, Nicolas Gayton, and Maurice Lemaire. "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation". In: *Structural Safety* 33.2 (2011), pp. 145–154.

[33] M Munoz Zuniga, Josselin Garnier, Emmanuel Remy, and Etienne de Rocquigny. "Adaptive directional stratification for controlled estimation of the probability of a rare event". In: *Reliability Engineering & System Safety* 96.12 (2011), pp. 1691–1712.

[34] Valeria Villa, Nicola Paltrinieri, Faisal Khan, and Valerio Cozzani. "Towards dynamic risk analysis: A review of the risk assessment approach and its limitations in the chemical process industry". In: *Safety Science* 89 (2016), pp. 77–93. ISSN: 0925-7535. DOI: 10.1016/j.ssci.2016.06.002.

[35] Arjan J Van Der Schaft and Johannes Maria Schumacher. *An introduction to hybrid dynamical systems*. Vol. 251. Springer London, 2000.

[36] Enrico Zio. "The future of risk assessment". In: *Reliability Engineering & System Safety* 177 (2018), pp. 176–190.

[37] Grégoire Jourdan. "Using risk-based regulations for licensing nuclear power plants: case study of gas-cooled fast reactor". PhD thesis. Massachusetts Institute of Technology, 2005.

[38] Vaibhav Yadav, Vivek Agarwal, Anderi V Gribok, and Curtis L Smith. *Dynamic PRA with component aging and degradation modeled utilizing plant risk monitoring data*. Tech. rep. Idaho National Lab.(INL), Idaho Falls, ID (United States), 2017.

[39] Faisal Khan, Samith Rathnayaka, and Salim Ahmed. "Methods and models in process safety and risk management: Past, present and future". In: *Process safety and environmental protection* 98 (2015), pp. 116–147.

[40] Anjana Meel and Warren D Seider. "Plant-specific dynamic failure assessment using Bayesian theory". In: *Chemical engineering science* 61.21 (2006), pp. 7036–7056.

[41] Maryam Kalantarnia, Faisal Khan, and Kelly Hawboldt. "Dynamic risk assessment using failure assessment and Bayesian theory". In: *Journal of Loss Prevention in the Process Industries* 22.5 (2009), pp. 600–606.

[42] Ankur Pariyani, Warren D Seider, Ulku G Oktem, and Masoud Soroush. "Dynamic risk analysis using alarm databases to improve process safety and product quality: Part II—Bayesian analysis". In: *AIChE Journal* 58.3 (2012), pp. 826–841.

[43] Arnab Roy, Prashant Srivastava, and Shishir Sinha. "Dynamic failure assessment of an ammonia storage unit: A case study". In: *Process Safety and Environmental Protection* 94 (2015), pp. 385–401.

[44] Nima Khakzad, Faisal Khan, and Paul Amyotte. "Dynamic risk analysis using bow-tie approach". In: *Reliability Engineering & System Safety* 104 (2012), pp. 36–44.

[45] Kun Chen, Xin Wei, Hui Li, Hao Lin, and Faisal Khan. "Operational risk analysis of blowout scenario in offshore drilling operation". In: *Process Safety and Environmental Protection* 149 (2021), pp. 422–431.

[46] Xinhong Li, Guoming Chen, Faisal Khan, and Changhang Xu. "Dynamic risk assessment of subsea pipelines leak using precursor data". In: *Ocean Engineering* 178 (2019), pp. 156–169.

[47] Nima Khakzad, Faisal Khan, and Paul Amyotte. "Dynamic safety analysis of process systems by mapping bow-tie into Bayesian network". In: *Process Safety and Environmental Protection* 91.1-2 (2013), pp. 46–53.

[48] Nima Khakzad, Faisal Khan, and Paul Amyotte. “Quantitative risk analysis of offshore drilling operations: A Bayesian approach”. In: *Safety science* 57 (2013), pp. 108–117.

[49] Nima Khakzad, Gabriele Landucci, and Genserik Reniers. “Application of dynamic Bayesian network to performance assessment of fire protection systems during domino effects”. In: *Reliability Engineering & System Safety* 167 (2017), pp. 232–247.

[50] Juan Chen, Ping-An Zhong, Ru An, Feilin Zhu, and Bin Xu. “Risk analysis for real-time flood control operation of a multi-reservoir system using a dynamic Bayesian network”. In: *Environmental Modelling & Software* 111 (2019), pp. 409–420.

[51] Yunfei Zhao, Jiejuan Tong, Liguo Zhang, and Guohua Wu. “Diagnosis of operational failures and on-demand failures in nuclear power plants: An approach based on dynamic Bayesian networks”. In: *Annals of Nuclear Energy* 138 (2020), p. 107181.

[52] Yuanjiang Chang, Xiangfei Wu, Changshuai Zhang, Guoming Chen, Xiuquan Liu, Jiayi Li, Baoping Cai, and Liangbin Xu. “Dynamic Bayesian networks based approach for risk analysis of subsea wellhead fatigue failure during service life”. In: *Reliability Engineering & System Safety* 188 (2019), pp. 454–462.

[53] Zhiguo Zeng and Enrico Zio. “Dynamic risk assessment based on statistical failure data and condition-monitoring degradation data”. In: *IEEE Transactions on Reliability* 67.2 (2018), pp. 609–622.

[54] Jinduo Xing, Zhiguo Zeng, and Enrico Zio. “A framework for dynamic risk assessment with condition monitoring data and inspection data”. In: *Reliability Engineering & System Safety* 191 (2019), p. 106552.

[55] Mohammed Taleb-Berrouane, Faisal Khan, and Paul Amyotte. “Bayesian Stochastic Petri Nets (BSPN)-A new modelling tool for dynamic safety and reliability analysis”. In: *Reliability Engineering & System Safety* 193 (2020), p. 106587.

[56] Hongyang Yu, Faisal Khan, Vikram Garaniya, and Arshad Ahmad. “Self-organizing map based fault diagnosis technique for non-Gaussian processes”. In: *Industrial & Engineering Chemistry Research* 53.21 (2014), pp. 8831–8843.

[57] O Zadakbar, S Imtiaz, and F Khan. “Dynamic risk assessment and fault detection using a multivariate technique”. In: *Process Safety Progress* 32.4 (2013), pp. 365–375.

[58] Nicola Paltrinieri, Louise Comfort, and Genserik Reniers. “Learning about risk: Machine learning for risk assessment”. In: *Safety science* 118 (2019), pp. 475–486.

[59] Steven H Rich and V Venkatasubramanian. "Model-based reasoning in diagnostic expert systems for chemical process plants". In: *Computers & Chemical Engineering* 11.2 (1987), pp. 111–122.

[60] Paul Paris and Fazil Erdogan. "A critical analysis of crack propagation laws". In: (1963).

[61] Charles H Oppenheimer and Kenneth A Loparo. "Physically based diagnosis and prognosis of cracked rotor shafts". In: *Component and Systems Diagnostics, Prognostics, and Health Management II*. Vol. 4733. International Society for Optics and Photonics. 2002, pp. 122–132.

[62] Rolf F Orsagh, Jeremy Sheldon, and Christopher J Klenke. *Prognostics/diagnostics for gas turbine engine bearings*. Vol. 36843. 2003.

[63] Adrian Cubillo, Suresh Perinpanayagam, and Manuel Esperon-Miguez. "A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery". In: *Advances in Mechanical Engineering* 8.8 (2016), p. 1687814016664660.

[64] Marina Rocha and Eugen Brühwiler. "Prediction of fatigue life of reinforced concrete bridges using fracture mechanics". In: *Proceedings bridge maintenance, safety, management, resilience and sustainability* 1.CONF (2012), pp. 3755–3761.

[65] Gunnar Härkegård, Josef Denk, and Klaus Stärk. "Growth of naturally initiated fatigue cracks in ferritic gas turbine rotor steels". In: *International journal of fatigue* 27.6 (2005), pp. 715–726.

[66] Claude Bathias and Paul C Paris. "Gigacycle fatigue of metallic aircraft components". In: *International Journal of Fatigue* 32.6 (2010), pp. 894–897.

[67] Samir Khan and Takehisa Yairi. "A review on the application of deep learning in system health management". In: *Mechanical Systems and Signal Processing* 107 (2018), pp. 241–265.

[68] V Muralidharan and V Sugumaran. "A comparative study of Naive Bayes classifier and Bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis". In: *Applied Soft Computing* 12.8 (2012), pp. 2023–2029.

[69] Alicia Robles-Velasco, Pablo Cortés, Jesús Muñuzuri, and Luis Onieva. "Prediction of pipe failures in water supply networks using logistic regression and support vector classification". In: *Reliability Engineering & System Safety* 196 (2020), p. 106754.

[70] Seunggyu Lee. "Monte Carlo simulation using support vector machine and kernel density for failure probability estimation". In: *Reliability Engineering & System Safety* 209 (2021), p. 107481.

[71] Jing Li, Rebecca J Stones, Gang Wang, Xiaoguang Liu, Zhongwei Li, and Ming Xu. "Hard drive failure prediction using decision trees". In: *Reliability Engineering & System Safety* 164 (2017), pp. 55–65.

[72] Jason Wu and Jack W Baker. “Statistical learning techniques for the estimation of lifeline network performance and retrofit selection”. In: *Reliability Engineering & System Safety* 200 (2020), p. 106921.

[73] Wei Cai, Jingyi Zhao, and Ming Zhu. “A real time methodology of cluster-system theory-based reliability estimation using k-means clustering”. In: *Reliability Engineering & System Safety* 202 (2020), p. 107045.

[74] Sébastien Schwartz, Juan José Montero Jimenez, Michel Salaün, and Rob Vingerhoeds. “A fault mode identification methodology based on self-organizing map”. In: *Neural Computing and Applications* 32.17 (2020), pp. 13405–13423.

[75] Yunfei Zhao, Wei Gao, and Carol Smidts. “Sequential Bayesian inference of transition rates in the hidden Markov model for multi-state system degradation”. In: *Reliability Engineering & System Safety* (2021), p. 107662.

[76] Weiting Zhang, Dong Yang, and Hongchao Wang. “Data-driven methods for predictive maintenance of industrial equipment: a survey”. In: *IEEE Systems Journal* 13.3 (2019), pp. 2213–2227.

[77] Jianxiao Mao, Hao Wang, and Billie F Spencer Jr. “Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders”. In: *Structural Health Monitoring* (2020), p. 1475921720924601.

[78] Xu Wang, Changqing Shen, Min Xia, Dong Wang, Jun Zhu, and Zhongkui Zhu. “Multi-scale deep intra-class transfer learning for bearing fault diagnosis”. In: *Reliability Engineering & System Safety* 202 (2020), p. 107050.

[79] Fan Xu, Fangfang Yang, Zicheng Fei, Zhelin Huang, and Kwok-Leung Tsui. “Life prediction of lithium-ion batteries based on stacked denoising autoencoders”. In: *Reliability Engineering & System Safety* 208 (2021), p. 107396.

[80] Sergio Cofre-Martel, Philip Kobrich, Enrique Lopez Droguett, and Viviana Meruane. “Deep convolutional neural network-based structural damage localization and quantification using transmissibility data”. In: *Shock and Vibration* 2019 (2019).

[81] Jinglong Chen, Hongjie Jing, Yuanhong Chang, and Qian Liu. “Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process”. In: *Reliability Engineering & System Safety* 185 (2019), pp. 372–382.

[82] Zunya Shi and Abdallah Chehade. “A dual-LSTM framework combining change point detection and remaining useful life prediction”. In: *Reliability Engineering & System Safety* 205 (2021), p. 107257.

[83] André Listou Ellefsen, Emil Bjørlykhaug, Vilmar Æsøy, Sergey Ushakov, and Houxiang Zhang. “Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture”. In: *Reliability Engineering & System Safety* 183 (2019), pp. 240–251.

[84] CP Andriotis and KG Papakonstantinou. "Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints". In: *Reliability Engineering & System Safety* 212 (2021), p. 107551.

[85] Chi Zhang, Chetan Gupta, Ahmed Farahat, Kosta Ristovski, and Dipanjan Ghosh. "Equipment health indicator learning using deep reinforcement learning". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2018, pp. 488–504.

[86] Roberto Rocchetta, L Bellani, M Compare, Enrico Zio, and E Patelli. "A reinforcement learning framework for optimal operation and maintenance of power grids". In: *Applied energy* 241 (2019), pp. 291–301.

[87] Ramin Moradi and Katrina Groth. "On the application of transfer learning in prognostics and health management". In: *Annual Conference of the PHM Society*. Vol. 12. 1. 2020, pp. 8–8.

[88] Ansi Zhang, Honglei Wang, Shaobo Li, Yuxin Cui, Zhonghao Liu, Guanci Yang, and Jianjun Hu. "Transfer learning with deep recurrent neural networks for remaining useful life estimation". In: *Applied Sciences* 8.12 (2018), p. 2416.

[89] Gabriel Michau and Olga Fink. "Domain Adaptation for One-Class Classification: Monitoring the Health of Critical Systems Under Limited Information". In: *International Journal of Prognostics and Health Management* 10 (2019), p. 11.

[90] Weijun Li, Sai Gu, Xiangping Zhang, and Tao Chen. "Transfer Learning for Process Fault Diagnosis: Knowledge Transfer from Simulation to Physical Processes". In: *Computers & Chemical Engineering* (2020), p. 106904.

[91] Chi Keong Reuben Lim and David Mba. "Switching Kalman filter for failure prognostic". In: *Mechanical Systems and Signal Processing* 52 (2015), pp. 426–435.

[92] Dawn An, Joo-Ho Choi, and Nam Ho Kim. "Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab". In: *Reliability Engineering & System Safety* 115 (2013), pp. 161–169.

[93] Seyed Fouad Karimian, Ramin Moradi, Sergio Cofre-Martel, Katrina M Groth, and Mohammad Modarres. "Neural network and particle filtering: a hybrid framework for crack propagation prediction". In: *arXiv preprint arXiv:2004.13556* (2020).

[94] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. "Particle filter-based prognostics: Review, discussion and perspectives". In: *Mechanical Systems and Signal Processing* 72 (2016), pp. 2–31.

[95] Leonardo R Rodrigues, Joao PP Gomes, Felipe AS Ferri, Ivo P Medeiros, Roberto KH Galvao, and Cairo L Nascimento Júnior. "Use of PHM information and system architecture for optimized aircraft maintenance planning". In: *IEEE Systems Journal* 9.4 (2014), pp. 1197–1207.

[96] Hamed Khorasgani, Gautam Biswas, and Shankar Sankararaman. "Methodologies for system-level remaining useful life prediction". In: *Reliability Engineering & System Safety* 154 (2016), pp. 8–18.

[97] A Abu-Samah, Muhammad Kashif Shahzad, and Eric Zamai. "Bayesian based methodology for the extraction and validation of time bound failure signatures for online failure prediction". In: *Reliability Engineering & System Safety* 167 (2017), pp. 616–628.

[98] Md. Tanjin Amin, Faisal Khan, and Syed Imtiaz. "Dynamic availability assessment of safety critical systems using a dynamic Bayesian network". In: *Reliability Engineering & System Safety* 178 (2018), pp. 108–117.

[99] Sinda Rebello, Hongyang Yu, and Lin Ma. "An integrated approach for system functional reliability assessment using Dynamic Bayesian Network and Hidden Markov Model". In: *Reliability Engineering & System Safety* 180 (2018), pp. 124–135.

[100] Chonlagarn Iamsumang, Ali Mosleh, and Mohammad Modarres. "Monitoring and learning algorithms for dynamic hybrid Bayesian network in on-line system health management applications". In: *Reliability Engineering & System Safety* 178 (2018), pp. 118–129.

[101] Baoping Cai, Hanlin Liu, and Min Xie. "A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks". In: *Mechanical Systems and Signal Processing* 80 (2016), pp. 31–44.

[102] Laibin Zhang, Xin Zhang, Jinqiu Hu, and Huizhou Liu. "A comprehensive method for safety management of a complex pump injection system used for shale-gas well fracturing". In: *Process Safety and Environmental Protection* 120 (2018), pp. 370–387.

[103] Olivier Blancke, Antoine Tahan, Dragan Komljenovic, Normand Amyot, Mélanie Lévesque, and Claude Hudon. "A holistic multi-failure mode prognosis approach for complex equipment". In: *Reliability Engineering & System Safety* 180 (2018), pp. 136–151.

[104] Lotfi A Zadeh. "A fuzzy-algorithmic approach to the definition of complex or imprecise concepts". In: *Systems Theory in the Social Sciences*. Springer, 1976, pp. 202–282.

[105] Yaneer Bar-Yam. *Dynamics of complex systems*. Vol. 213. Addison-Wesley Reading, MA, 1997.

[106] Maxi San Miguel, Jeffrey H Johnson, Janos Kertesz, Kimmo Kaski, Albert Dıéaz-Guilera, Robert S MacKay, Vittorio Loreto, Péter Érdi, and Dirk Helbing. "Challenges in complex systems science". In: *The European Physical Journal Special Topics* 214.1 (2012), pp. 245–271.

[107] Jinghai Li and Mooson Kwauk. "Exploring complex systems in chemical engineering—the multi-scale methodology". In: *Chemical Engineering Science* 58.3-6 (2003), pp. 521–535.

[108] JC Charpentier and TF McKenna. “Managing complex systems: some trends for the future of chemical and process engineering”. In: *Chemical Engineering Science* 59.8-9 (2004), pp. 1617–1640.

[109] Christophe Simon, Philippe Weber, and Eric Levrat. “Bayesian networks and evidence theory to model complex systems reliability”. In: *Journal of Computers (JCP)* 2.1 (2007), pp. 33–43.

[110] John H Miller and Scott E Page. *Complex adaptive systems: An introduction to computational models of social life*. Vol. 17. Princeton university press, 2009.

[111] Sarah A Sheard and Ali Mostashari. “Principles of complex systems for systems engineering”. In: *Systems Engineering* 12.4 (2009), pp. 295–311.

[112] Cuong D Dao and Ming J Zuo. “Selective maintenance of multi-state systems with structural dependence”. In: *Reliability engineering & system safety* 159 (2017), pp. 184–195.

[113] Liudong Xing, Brock A Morrissette, and Joanne Bechta Dugan. “Combinatorial reliability analysis of imperfect coverage systems subject to functional dependence”. In: *IEEE Transactions on Reliability* 63.1 (2014), pp. 367–382.

[114] Minou CA Olde Keizer, Ruud H Teunter, Jasper Veldman, and M Zied Babai. “Condition-based maintenance for systems with economic dependence and load sharing”. In: *International Journal of Production Economics* 195 (2018), pp. 319–327.

[115] Mengkai Xu, Xiaoning Jin, Sagar Kamarthi, and Md Noor-E-Alam. “A failure-dependency modeling and state discretization approach for condition-based maintenance optimization of multi-component systems”. In: *Journal of manufacturing systems* 47 (2018), pp. 141–152.

[116] Yong Sun, Lin Ma, Joseph Mathew, and Sheng Zhang. “An analytical model for interactive failures”. In: *Reliability Engineering & System Safety* 91.5 (2006), pp. 495–504.

[117] Nipat Rasmekomen and Ajith Kumar Parlikad. “Condition-based maintenance of multi-component systems with degradation state-rate interactions”. In: *Reliability Engineering & System Safety* 148 (2016), pp. 1–10.

[118] Nuria C Caballé, IT Castro, CJ Pérez, and José Manuel Lanza-Gutiérrez. “A condition-based maintenance of a dependent degradation-threshold-shock model in a system with multiple degradation processes”. In: *Reliability Engineering & System Safety* 134 (2015), pp. 98–109.

[119] Mengfei Fan, Zhiguo Zeng, Enrico Zio, and Rui Kang. “Modeling dependent competing failure processes with degradation-shock dependence”. In: *Reliability Engineering & System Safety* 165 (2017), pp. 422–430.

[120] Xiaolei Fang, Nagi Z Gebraeel, and Kamran Paynabar. “Scalable prognostic models for large-scale condition monitoring applications”. In: *IISE Transactions* 49.7 (2017), pp. 698–710.

[121] Tatsuya Sakurahara, Grant Schumock, Seyed Reihani, Ernie Kee, and Zahra Mohaghegh. "Simulation-informed probabilistic methodology for common cause failure analysis". In: *Reliability Engineering & System Safety* 185 (2019), pp. 84–99.

[122] Enrique López Droguett and Ali Mosleh. "Bayesian methodology for model uncertainty using model performance data". In: *Risk Analysis: An International Journal* 28.5 (2008), pp. 1457–1476.

[123] Piero Baraldi and Enrico Zio. "A combined Monte Carlo and possibilistic approach to uncertainty propagation in event tree analysis". In: *Risk Analysis: An International Journal* 28.5 (2008), pp. 1309–1326.

[124] Enrique López Droguett and Ali Mosleh. "Bayesian treatment of model uncertainty for partially applicable models". In: *Risk Analysis* 34.2 (2014), pp. 252–270.

[125] Gabriel Terejanu, Puneet Singla, Tarunraj Singh, and Peter D Scott. "Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models". In: *Journal of Guidance, Control, and Dynamics* 31.6 (2008), pp. 1623–1633.

[126] Shankar Sankararaman and Kai Goebel. "Uncertainty in prognostics and systems health management". In: *International Journal of Prognostics and Health Management* 6.010 (2015).

[127] Saurabh Mittal, Saikou Diallo, and Andreas Tolk. *Emergent behavior in complex systems engineering: a modeling and simulation approach*. John Wiley & Sons, 2018.

[128] W Ross Ashby. *An introduction to cybernetics*. Chapman & Hall Ltd, 1961.

[129] Payuna Uday, Rakshit Chandrahasa, and Karen Marais. "System importance measures: Definitions and application to system-of-systems analysis". In: *Reliability Engineering & System Safety* 191 (2019), p. 106582.

[130] Scott Thacker, Raghav Pant, and Jim W Hall. "System-of-systems formulation and disruption analysis for multi-scale critical national infrastructures". In: *Reliability Engineering & System Safety* 167 (2017), pp. 30–41.

[131] Hao Ren, Yi Chai, Jianfeng Qu, Xin Ye, and Qiu Tang. "A novel adaptive fault detection methodology for complex system using deep belief networks and multiple models: A case study on cryogenic propellant loading system". In: *Neurocomputing* 275 (2018), pp. 2111–2125.

[132] Andrés Ruiz-Tagle Palazuelos and Enrique López Droguett. "System-level prognostics and health management: a graph neural network based framework". In: *Accepted for publication in proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* (2020).

[133] Jamie Coble, Pradeep Ramuhalli, Leonard J. Bond, J. Hines, and Belle Ipadhyaya. “A review of prognostics and health management applications in nuclear power plants”. In: *International Journal of prognostics and health management* 6 (2015), pp. 16–37.

[134] Roy Assaf, Phuc Do, Philip Scarf, and Samia Nefti-Meziani. “Diagnosis for systems with multi-component wear interactions”. In: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2017, pp. 96–102.

[135] Tangbin Xia, Yifan Dong, Lei Xiao, Shichang Du, Ershun Pan, and Lifeng Xi. “Recent advances in prognostics and health management for advanced manufacturing paradigms”. In: *Reliability Engineering & System Safety* 178 (2018), pp. 255–268.

[136] Enrico Zio. “Integrated deterministic and probabilistic safety assessment: Concepts, challenges, research directions”. In: *Nuclear Engineering and Design* 280 (2014), pp. 413–419. ISSN: 0029-5493. DOI: 10.1016/j.nucengdes.2014.09.004.

[137] Ali Mosleh. “PRA: a perspective on strengths, current limitations, and possible improvements”. In: *Nuclear Engineering and Technology* 46.1 (2014), pp. 1–10. ISSN: 1738-5733. DOI: 10.5516/NET.03.2014.700.

[138] Nathan Siu, Don Marksberry, Susan Cooper, Kevin Coyne, and Martin Stutzke. “PSA technology challenges revealed by the Great East Japan Earthquake”. In: *Proceedings of the PSAM Topical Conference in Light of the Fukushima Dai-Ichi Accident* (Apr. 15–17, 2013). 2013.

[139] Norman Rasmussen. *Reactor safety study: An assessment of accident risks in U.S. commercial nuclear power plants.* Tech. rep. WASH-1400 (NUREG-75/014). U.S. Nuclear Regulatory Commission, Oct. 1975.

[140] MM Manjurul Islam and Jong-Myon Kim. “Reliable multiple combined fault diagnosis of bearings using heterogeneous feature models and multiclass support vector Machines”. In: *Reliability Engineering & System Safety* 184 (2019), pp. 55–66.

[141] Hongru Li, Jian Sun, Hui Ma, Zaike Tian, and Yifan Li. “A novel method based upon modified composite spectrum and relative entropy for degradation feature extraction of hydraulic pump”. In: *Mechanical Systems and Signal Processing* 114 (2019), pp. 399–412.

[142] Tao Yuan, Xinying Wu, Suk Joo Bae, and Xiaoyan Zhu. “Reliability assessment of a continuous-state fuel cell stack system with multiple degrading components”. In: *Reliability Engineering & System Safety* 189 (2019), pp. 157–164.

[143] Nam-Ho Kim, Dawn An, and Joo-Ho Choi. *Prognostics and health management of engineering systems: An introduction.* Springer, 2016.

[144] Dawn An, Nam H Kim, and Joo-Ho Choi. "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews". In: *Reliability Engineering & System Safety* 133 (2015), pp. 223–236.

[145] Chaochao Chen, George Vachtsevanos, and Marcos E Orchard. "Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach". In: *Mechanical Systems and Signal Processing* 28 (2012), pp. 597–607.

[146] Datong Liu, Jingyue Pang, Jianbao Zhou, Yu Peng, and Michael Pecht. "Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression". In: *Microelectronics Reliability* 53.6 (2013), pp. 832–839.

[147] Khanh Le Son, Mitra Fouladirad, and Anne Barros. "Remaining useful lifetime estimation and noisy gamma deterioration process". In: *Reliability Engineering & System Safety* 149 (2016), pp. 76–87.

[148] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Mao-Yin Chen, and Dong-Hua Zhou. "A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation". In: *Mechanical Systems and Signal Processing* 35.1-2 (2013), pp. 219–237.

[149] Diego Alejandro Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, and Gerard Tripot. "A data-driven failure prognostics method based on mixture of Gaussian hidden Markov models". In: *IEEE Transactions on Reliability* 61.2 (2012), pp. 491–503.

[150] Andrés Ruiz-Tagle Palazuelos, Enrique López Droguett, and Rodrigo Pascual. "A novel deep capsule neural network for remaining useful life estimation". In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 234.1 (2020), pp. 151–167.

[151] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. "Meta-transfer learning for few-shot learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 403–412.

[152] Piero Baraldi, Michele Compare, Sergio Sauco, and Enrico Zio. "Ensemble neural network-based particle filtering for prognostics". In: *Mechanical Systems and Signal Processing* 41.1-2 (2013), pp. 288–300.

[153] Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". In: *Mechanical Systems and Signal Processing* 104 (2018), pp. 799–834.

[154] Jaouher Ben Ali, Nader Fnaiech, Lotfi Saidi, Brigitte Chebel-Morello, and Farhat Fnaiech. "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals". In: *Applied Acoustics* 89 (2015), pp. 16–27.

[155] Ji-Eun Byun, Kilian Zwirglmaier, Daniel Straub, and Junho Song. "Matrix-based Bayesian Network for efficient memory storage and flexible inference". In: *Reliability Engineering & System Safety* 185 (2019), pp. 533–545.

[156] Omar Kammouh, Paolo Gardoni, and Gian Paolo Cimellaro. "Probabilistic framework to evaluate the resilience of engineering systems using Bayesian and dynamic Bayesian networks". In: *Reliability Engineering & System Safety* 198 (2020), p. 106813.

[157] Rodrigo Costa, Terje Haukaas, Stephanie E Chang, and Hadi Dowlatabadi. "Object-oriented model of the seismic vulnerability of the fuel distribution network in coastal British Columbia". In: *Reliability Engineering & System Safety* 186 (2019), pp. 11–23.

[158] Renaud Lambiotte, Martin Rosvall, and Ingo Scholtes. "From networks to optimal higher-order models of complex systems". In: *Nature physics* (2019), p. 1.

[159] Hendrik M Odendaal and Thomas Jones. "Actuator fault detection and isolation: An optimised parity space approach". In: *Control Engineering Practice* 26 (2014), pp. 222–232.

[160] Nizar Chatti, Belkacem Ould-Bouamama, Anne-Lise Gehin, and Rochdi Merzouki. "Signed bond graph for multiple faults diagnosis". In: *Engineering Applications of Artificial Intelligence* 36 (2014), pp. 134–147.

[161] Sohag Kabir, Martin Walker, and Yiannis Papadopoulos. "Dynamic system safety analysis in HiP-HOPS with Petri nets and Bayesian networks". In: *Safety science* 105 (2018), pp. 55–70.

[162] Shen Yin and Okyay Kaynak. "Big data for modern industry: challenges and trends [point of view]". In: *Proceedings of the IEEE* 103.2 (2015), pp. 143–146.

[163] Jinqiu Hu, Laibin Zhang, Lin Ma, and Wei Liang. "An integrated safety prognosis model for complex system based on dynamic Bayesian network and ant colony algorithm". In: *Expert Systems with Applications* 38.3 (2011), pp. 1431–1446.

[164] Francesca M Favaroò and Joseph H Saleh. "Toward risk assessment 2.0: Safety supervisory control and model-based hazard monitoring for risk-informed safety interventions". In: *Reliability Engineering & System Safety* 152 (2016), pp. 316–330.

[165] Jie Liu and Enrico Zio. "System dynamic reliability assessment and failure prognostics". In: *Reliability Engineering & System Safety* 160 (2017), pp. 21–36.

[166] Hyeonmin Kim, Jung Taek Kim, and Gyunyoung Heo. "Failure rate updates using condition-based prognostics in probabilistic safety assessments". In: *Reliability Engineering & System Safety* 175 (2018), pp. 225–233.

[167] Enno Ruijters and Mariëlle Stoelinga. "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools". In: *Computer science review* 15 (2015), pp. 29–62.

[168] Nima Hatami, Yann Gavet, and Johan Debayle. "Classification of time-series images using deep convolutional neural networks". In: *Tenth international conference on machine vision (ICMV 2017)*. Vol. 10696. International Society for Optics and Photonics. 2018, 106960Y.

[169] Ramin Moradi and Katrina M Groth. "Modernizing risk assessment: A systematic integration of PRA and PHM techniques". In: *Reliability Engineering & System Safety* 204 (2020), p. 107194.

[170] Mohammad Modarres, Mark P Kaminskiy, and Vasiliy Krivtsov. *Reliability engineering and risk analysis: a practical guide*. CRC press, 2016.

[171] Nicola Paltrinieri, Gabriele Landucci, and Pierluigi Rossi. "An Integrated Approach to Support the Dynamic Risk Assessment of Complex Industrial Accidents". In: *Chemical Engineering Transactions* 77 (2019), pp. 265–270.

[172] Aihua Liu, Ke Chen, Xiaofei Huang, Didi Li, and Xiaochun Zhang. "Dynamic risk assessment model of buried gas pipelines based on system dynamics". In: *Reliability Engineering & System Safety* 208 (2021), p. 107326.

[173] Joaquién Figueroa Barraza, Enrique Lopez Droguett, Viviana Meruane Naranjo, and Marcelo Ramos Martins. "Capsule neural networks for structural damage localization and quantification using transmissibility data". In: *Applied Soft Computing* 97 (2020), p. 106732.

[174] Gabriel San Martin, Enrique López Droguett, Viviane Meruane, and Márcio das Chagas Moura. "Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis". In: *Structural Health Monitoring* 18.4 (2019), pp. 1092–1128.

[175] Ramin Moradi, Andrés Ruiz-Tagle Palazuelos, Enrique López Droguett, and Katrina M Groth. "Towards a Framework for Risk Monitoring of Complex Engineering Systems with Online Operation Data: A Deep Learning Based Solution". In: *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference*. 2020.

[176] Philippe Weber, Gabriela Medina-Oliva, Christophe Simon, and Benoit Iung. "Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas". In: *Engineering Applications of Artificial Intelligence* 25.4 (2012), pp. 671–682.

[177] Ruonan Liu, Boyuan Yang, Enrico Zio, and Xuefeng Chen. "Artificial intelligence for fault diagnosis of rotating machinery: A review". In: *Mechanical Systems and Signal Processing* 108 (2018), pp. 33–47.

[178] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X Gao. “Deep learning and its applications to machine health monitoring”. In: *Mechanical Systems and Signal Processing* 115 (2019), pp. 213–237.

[179] David C Yu, Thanh C Nguyen, and Peter Haddawy. “Bayesian network model for reliability assessment of power systems”. In: *IEEE transactions on power systems* 14.2 (1999), pp. 426–432.

[180] José Gerardo Torres-Toledano and Luis Enrique Sucar. “Bayesian networks for reliability analysis of complex systems”. In: *Ibero-American Conference on Artificial Intelligence*. Springer. 1998, pp. 195–206.

[181] Esmaeil Zarei, Ali Azadeh, Nima Khakzad, Mostafa Mirzaei Aliabadi, and Iraj Mohammadfam. “Dynamic safety assessment of natural gas stations using Bayesian network”. In: *Journal of hazardous materials* 321 (2017), pp. 830–840.

[182] Iris Tien and Armen Der Kiureghian. “Algorithms for Bayesian network modeling and reliability assessment of infrastructure systems”. In: *Reliability Engineering & System Safety* 156 (2016), pp. 134–147.

[183] Nima Khakzad. “Application of dynamic Bayesian network to risk analysis of domino effects in chemical infrastructures”. In: *Reliability Engineering & System Safety* 138 (2015), pp. 263–272.

[184] Baoping Cai, Lei Huang, and Min Xie. “Bayesian networks in fault diagnosis”. In: *IEEE Transactions on Industrial Informatics* 13.5 (2017), pp. 2227–2240.

[185] Elaheh Rabiei, Enrique Lopez Droguett, and Mohammad Modarres. “A prognostics approach based on the evolution of damage precursors using dynamic Bayesian networks”. In: *Advances in mechanical engineering* 8.9 (2016), 168 7814016666747.

[186] Katrina M Groth and Ali Mosleh. “Deriving causal Bayesian networks from human reliability analysis data: A methodology and example model”. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 226.4 (2012), pp. 361–379.

[187] Katrina M Groth, Reuel Smith, and Ramin Moradi. “A hybrid algorithm for developing third generation HRA methods using simulator data, causal models, and cognitive science”. In: *Reliability Engineering & System Safety* 191 (2019), p. 106507.

[188] Pedro Larranaga, Hossein Karshenas, Concha Bielza, and Roberto Santana. “A review on evolutionary algorithms in Bayesian network learning and inference tasks”. In: *Information Sciences* 233 (2013), pp. 109–125.

[189] Zhiwei Ji, Qibiao Xia, and Guanmin Meng. “A review of parameter learning methods in Bayesian network”. In: *International Conference on Intelligent Computing*. Springer. 2015, pp. 3–12.

[190] D Codetta-Raiteri. "Editorial for the Special Issue on Bayesian Networks: Inference Algorithms, Applications, and Software Tools"". In: *Algorithms* 14 (2021), p. 138.

[191] Andres Ruiz-Tagle, Enrique Lopez Droguett, and Katrina M Groth. "Exploiting the Capabilities of Bayesian Networks for Engineering Risk Assessment: Causal Reasoning through Interventions". In: *Risk Analysis* (2021).

[192] Richard E Neapolitan et al. *Learning Bayesian networks*. Vol. 38. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[193] Thomas Dyhre Nielsen and Finn Verner Jensen. *Bayesian networks and decision graphs*. Springer Science & Business Media, 2009.

[194] Haipeng Guo and William Hsu. "A survey of algorithms for real-time Bayesian network inference". In: *Join Workshop on Real Time Decision Support and Diagnosis Systems*. 2002.

[195] Hichem Boudali and Joanne Bechta Dugan. "A discrete-time Bayesian network reliability modeling and analysis framework". In: *Reliability Engineering & System Safety* 87.3 (2005), pp. 337–349.

[196] Daniele Codetta-Raiteri and Luigi Portinale. "Generalized Continuous Time Bayesian Networks as a modelling and analysis formalism for dependable systems". In: *Reliability Engineering & System Safety* 167 (2017), pp. 639–651.

[197] Xiaohu Zheng, Wen Yao, Yingchun Xu, and Xianqi Chen. "Improved compression inference algorithm for reliability analysis of complex multi-state satellite system based on multilevel Bayesian network". In: *Reliability Engineering & System Safety* 189 (2019), pp. 123–142.

[198] Michelle Bensi, Armen Der Kiureghian, and Daniel Straub. "Efficient Bayesian network modeling of systems". In: *Reliability Engineering & System Safety* 112 (2013), pp. 200–213.

[199] Xiaobing Yuan, Baoping Cai, Yunpeng Ma, Junyan Zhang, Kabwe Mulenga, Yonghong Liu, and Guoming Chen. "Reliability evaluation methodology of complex systems based on dynamic object-oriented Bayesian networks". In: *IEEE Access* 6 (2018), pp. 11289–11300.

[200] Zhaoyi Xu and Joseph Homer Saleh. "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities". In: *Reliability Engineering & System Safety* (2021), p. 107530.

[201] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *NIPS*. 2017, pp. 6405–6416. URL: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.

[202] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[203] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. “Weight uncertainty in neural network”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1613–1622.

[204] Meire Fortunato, Charles Blundell, and Oriol Vinyals. “Bayesian recurrent neural networks”. In: *arXiv preprint arXiv:1704.02798* (2017).

[205] Biao Huang, Qiao Ding, Guozi Sun, and Huakang Li. “Stock prediction based on Bayesian-LSTM”. In: *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*. 2018, pp. 128–133.

[206] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. “Hands-on Bayesian Neural Networks–a Tutorial for Deep Learning Users”. In: *arXiv preprint arXiv:2007.06823* (2020).

[207] Alex Graves. “Practical variational inference for neural networks”. In: *Advances in neural information processing systems*. Citeseer. 2011, pp. 2348–2356.

[208] Wray L Buntine. “Operations for learning with graphical models”. In: *Journal of artificial intelligence research* 2 (1994), pp. 159–225.

[209] Diederik P Kingma, Max Welling, et al. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[210] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. “Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches”. In: *International Conference on Learning Representations*. 2018. URL: `https://openreview.net/forum?id=rJNpifWAb`.

[211] Luyang Jing, Ming Zhao, Pin Li, and Xiaoqiang Xu. “A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox”. In: *Measurement* 111 (2017), pp. 1–10.

[212] Sergio Cofre-Martel, Enrique Lopez Droguett, and Mohammad Modarres. “Uncovering the Underlying Physics of Degrading System Behavior Through a Deep Neural Network Framework: The Case of Remaining Useful Life Prognosis”. In: *arXiv preprint arXiv:2006.09288* (2020).

[213] Liu Yang, Xuhui Meng, and George Em Karniadakis. “B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data”. In: *Journal of Computational Physics* 425 (2021), p. 109913.

[214] Wennian Yu, II Yong Kim, and Chris Mechefske. “An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme”. In: *Reliability Engineering & System Safety* 199 (2020), p. 106926.

[215] OREDA. *OREDA (2015). Offshore and Onshore Reliability Data 6th edition.* 2015.

[216] Saeid Mokhatab and William A Poe. *Handbook of natural gas transmission and processing*. Gulf professional publishing, 2012.

[217] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2 (2012).

[218] Ramin Moradi and Katrina M Groth. "Hydrogen storage and delivery: Review of the state of the art technologies and risk and reliability analysis". In: *International Journal of Hydrogen Energy* 44.23 (2019), pp. 12254–12269.