ABSTRACT

Title of dissertation: DYNAMIC BAYESIAN NETWORK UPDATING APPROACHES FOR ENABLING CAUSAL PROGNOSTICS AND HEALTH MANAGEMENT OF COMPLEX ENGINEERING SYSTEMS

Austin Drisko Lewis
Doctor of Philosophy, 2022

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

Complex engineering systems (CESes), such as nuclear power plants or manufacturing plants, are critical to a wide range of industries and utilities; as such, it is important to be able to monitor their system health and make informed decisions on maintenance and risk management practices. However, currently available system-level monitoring approaches either ignore complex dependencies in their probabilistic risk assessments (PRA) or are prognostics and health management (PHM) techniques intended for simpler systems. The gap in CES health management needs to be closed through the development of techniques and models built from a systematic integration of PHM and PRA (SIPPRA) approach that considers a system's causal factors and operational context when generating health assessments.

The following dissertation describes a concentrated study that addresses one of the challenges facing SIPPRA: how to appropriately discretize a CES's oper-

ational timeline derived from multiple data streams to create discrete time-series data for use as model inputs over meaningful time periods. This research studies how different time scales and discretization approaches impact the performance of dynamic Bayesian Networks (DBNs), models that are increasingly used for causal-based inferences and system-level assessments, specifically built for SIPPRA health management. The impact of this research offers new insight into how to construct such DBNs to better support system-level health management for CESes.

DYNAMIC BAYESIAN NETWORK DATA UPDATING APPROACHES FOR ENABLING CAUSAL PROGNOSTICS AND HEALTH MANAGEMENT OF COMPLEX ENGINEERING SYSTEMS

by

Austin Drisko Lewis

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
2022

Advisory Committee:
Associate Professor Katrina Groth, Chair/Advisor
Assistant Professor Michelle Bensi
Professor Jeffrey Herrmann
Professor Mohammed Modarres
Professor Gregory Baecher, Dean's Representative

Acknowledgments

Although my name is front and center on this document, none of this would have been possible without the support, advice, and guidance from so many. Here is my feeble attempt at acknowledging all of you who have been beside me throughout this incredible journey.

First and foremost, I would like to thank my advisor, Professor Katrina Groth, for the incredible amount of support and mentorship over these past few years. It is incredible to see just how much I have learned and grown as a independent thinker and researcher, and it is largely in part to her openness in sharing her experiences and insight with me. It has been a pleasure to work with and learn from her, and I can only hope that I will one day be as formative a presence in someone's career as she has been for me.

I would also like to thank Professors Baecher, Bensi, Herrmann, and Modarres for agreeing to serve on my dissertation committee and for sparing their invaluable time reviewing my manuscript. Their feedback during the proposal process helped to strengthen the arguments I have presented in this final version. I would like to specifically call out Professors Herrmann, Bensi, and Modarres for taking time to read and share their feedback on specific aspects of my research.

My colleagues in the Systems Risk and Reliability (SyRRA) lab have seen me everyday for the past four years; they know my work at least as well as anybody else. They have all served as brainstorming partners, paper reviewers, and presentation critics. I am constantly inspired by their fresh takes and continual efforts to help

improve my research. In particular, I want to thank Vinnie Paglioni and Andres Ruiz-Tagle for all their time answering my endless questions and being an incredible support network as my colleagues and friends.

Even though they have been on the other side of the country for most of my time working on this research, I am so grateful for my family's support and willingness to read my drafts or sit through dry runs of my presentations. A special thanks is also needed for my boyfriend Nick, who has been my rock over the past four years. I could not have done this without any of you.

I would like to acknowledge financial support from the Clark Foundation. Additionally, I want to acknowledge support from Zachary Jankovsky and Matthew Denman for providing insight and data from their previous research at SANDIA. Without either, this research literally would have not been possible.

Lastly to everyone else that I've failed to call out specifically - thank you so much for your support and providing safe outlets for me to discuss my research, celebrate my successes, and vent my frustrations when research progress seemed slow. It has truly meant so much to me.

Thank you all.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ASME	American Society of Mechanical Engineers
BN	Bayesian Network
CBM	Condition-based Monitoring
CES	Complex Engineering System
CPT	Conditional Probability Table
DBN	Dynamic Bayesian Network
DET	Dynamic Event Tree
DRACS	Direct Reactor Auxiliary Cooling System
ESD	Event Sequence Diagram
ESREL	European Safety and Reliability Conference
FDIR	Failure Detection, Identification, and Recovery
FT	Fault Tree
GQM	Goal-Question-Metric
PHM	Prognostics and Health Management
PRA	Probabilistic Risk Assessment
PSAM	Probabilistic Safety and Assessment Management
RAMS	Reliability and Maintainability Symposium
RESS	Reliability Engineering and System Safety
RPS	Reactor Protection System
RUL	Remaining Useful Life
SERAD	Safety Engineering and Risk Analysis Division
SFR	Sodium Fast Reactor
SIPPRA	Systematic Integration of PHM and PRA
SNL	Sandia National Laboratories
SyRRA	Systems Risk and Reliability Analysis
TOP	Transient Overpower
TTBN	Two-Time Slice Bayesian Network
UMD-CRR	University of Maryland - Center for Risk and Reliability

Chapter 1: Introduction

1.1 Background and Motivation

1.1.1 Complex Engineering Systems (CESes)

Complex engineering systems (CESes) are comprised of interconnected and interdependent human, hardware, and software components. Examples include nuclear power plants, chemical processing facilities, and transportation infrastructure. CESes are distinct from other engineering systems in that they rely heavily on human involvement to maintain their functionality. As a result, the relational logic structure behind CESes are usually quite intricate; between human operators, machinery, and software programming, deep interconnections within CESes make it challenging to assess system health from a single component.

CESes have become ubiquitous and serve as key and firmly integrated aspects of critical infrastructure [1]. There is an increasing reliance on these systems; for many CESes, their prominence in critical industries and utilities makes the consequences of system failure exorbitantly expensive and undesirable. Failure of these services could lead to increased risks in the public safety, national security, and economic sectors [2, 3].

1.1.2 CES Health Management

Rather than allow a CES to operate until failure, engineers monitor such systems and generate current health assessments based on operational data. By tracking CES health, engineers can be more informed when making maintenance decisions designed to maintain or extend the system's lifespan. For CES operators, the ability to assess its current health and forecast future health results in informed operational decisions and contributes to responsive system maintenance and risk management practices that improve the system's safety, availability, and reliability.

When systems consist of highly interdependent components, as is the case for CESes, the health and functionality of one component is dependent upon the health and functionality of others in the system [4]. These relationships may lead system components to degrade or operate differently than otherwise expected. Sensors that monitor the health of strategic system components generate data about the system's current performance. These data have the potential to provide diagnostic information about the current health of the overall system as well as a prognostic assessment of future health states given its current health status. Understanding the current and potential future health states of a system allows operators and maintainers to make more informed decisions to prolong a system's operating life before it loses critical functionality. This is a priority for systems that are mission critical, expensive to repair or replace from a failed state, or pose a safety risk to humans and other associated systems if not fully operational. The process of converting data and expert knowledge from CESes into informed diagnostic and

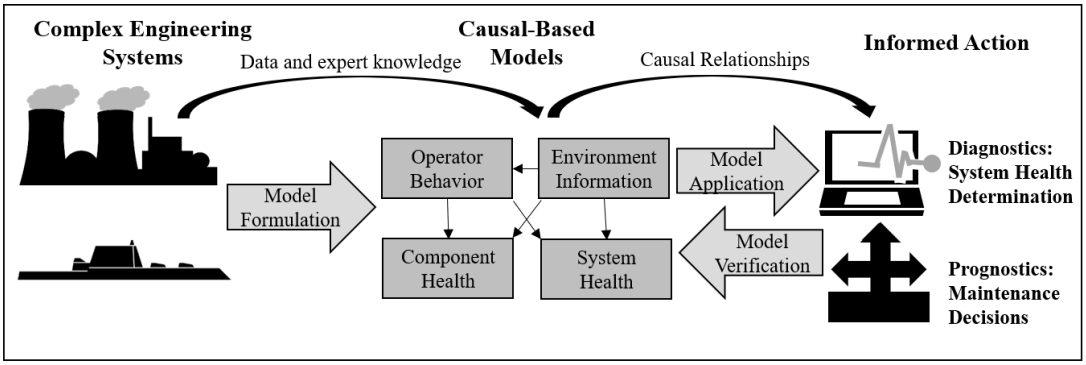


Figure 1.1: Modeling complex engineering systems (CESes), such as power plants and large maritime vessels, with causal-based models like Dynamic Bayesian Networks (DBNs), provides system operators and maintainers improved diagnostic and prognostic awareness.

prognostic decisions is illustrated in Figure 1.1. Such knowledge about a CES's health is particularly important following an accident event where action is needed to mitigate further system damage and loss of functionality.

1.1.3 Health Management through Modeling

A common way to manage system health is to represent system behavior and functionality through the use of models and simulations. This allows operators to explore potentially catastrophic scenarios without having to subject the actual system to those kinds of conditions. There are three forms of health management modeling and simulations; risk-based modeling, health-based modeling, and a systematic integration between the two.

1.1.3.1 Risk-based Modeling (PRA)

One approach for modeling system health is to represent it through system risk, i.e., the likelihood of system failure under different operational scenarios and their resulting consequences. Historically, however, the intricacies and dependencies within CES components made it too difficult to capture specific operational characteristics from available modeling tools and techniques. As a result, system health and risk management practices for CESes have traditionally relied on logic-based system representations that are greatly simplified and assume minimal or independent relationships between components [5]. These system-level models, such as fault trees (FTs) and event sequence diagrams (ESDs) depicting system accident sequences, are well integrated into probabilistic risk assessment (PRA) practices. For CESes, however, operational assessments from these models are not easily updated with new information made available from on-line sensors and health readings.

1.1.3.2 Health-based Modeling (PHM)

In addition to the logic-based representations found in risk-based models, monitoring systems through sensors and regular data collection makes it possible to derive health assessments from operational data. Using measured operational data values and current or historical system and component health states [6], correlations and associations inherent in a system can be identified using machine learning techniques. These relationships may be determined to be causal in nature via causal discovery methods [7]. When combined with an understanding of the system's com-

ponent structure and relevant physics-of-failure relationships, the resulting causal models are powerful tools for providing insight into a system's health [8]. Such tools have led to the development of prognostics and health management (PHM) as a field in its own right. PHM research, however, has primarily focused on simple systems that follow physical failure and degradation processes outside of human involvement. This raises potential questions as to whether PHM techniques can be scaled from a simple system scenario to support CES health management.

1.1.3.3 Systematic Integration of PHM and PRA (SIPPRA)

PHM and PRA have both been demonstrated as valuable tools for understanding system health and safety, respectively. However, both fields are limited in the scope of their abilities. This is particularly evident when applying either approach to CESes, in which safety and operation are both critical. To address these shortcomings, a greater effort has been placed on better modeling CES health by performing a systematic integration of PHM and PRA (SIPPRA) methods. In incorporating system safety and health into a modeling approach, SIPPRA model designs better capture and represent CESes for improved management.

One SIPPRA approach is to use Bayesian networks (BNs), graphical representations of the causal relationships within a system [9]. In this way, BNs and their temporal counterparts, dynamic Bayesian Networks (DBNs), serve as potential model structures for connecting causal relationships with available operational data. DBNs have been applied to a number of CESes and operational scenarios

[10–12]. Due to their clear logic structure and inference capabilities, DBNs offer valuable insight as health monitoring models.

As SIPPRA health management relies on available system data, different model structures may emphasize certain system information over others. This would result in many viable models generating distinct health assessments leading to different operational and maintenance decisions. It is important, therefore, to understand the performance of available model alternatives and the impact that modeling choices have on SIPPRA health assessments. This is certainly the case for DBNs, which represents temporal relationships through conditional probabilities and distributions over a defined time period.

1.2 Research Overview

1.2.1 Problem Statement

There is a need for model designers to develop effective CES health monitoring models for improved system management. However, the current methods for system health and safety management, PRA and PHM, are not sufficiently complex or scalable enough to be effectively applied to these systems. Although there are many approaches to systematically integrating the two techniques as SIPPRA, modeling CESes using DBNs shows promise as the causal-based networks offer powerful inference capabilities alongside a clear logic structure. The novelty of this approach raises many questions surrounding effective DBN model designs for SIPPRA health management. One area in particular is the effect that different approaches to dis-

cretizing operational system data streams for DBN construction and updating has on model performance.

1.2.2 Research Objectives

This research has four main objectives designed to address the current limitations in understanding the impact different strategies for discretizing continuous operational system data streams has on the performance of DBN models used in SIPPRA health management:

1. Define metrics for comparing the performance of SIPPRA methods.

2. Identify and define methods that discretize continuous time-series data for use in SIPPRA-focused DBNs.

3. Develop a real-world case study that demonstrates the feasibility of using DBNs for assessing a CES's operational health and facilitates the comparison of multiple DBN data stream discretization methods.

4. Compare the performance of DBNs built with different data stream discretization strategies.

These four objectives, presented in the visual in Figure 1.2, align to different aspects of understanding how to develop SIPPRA models that respond to changes in system operations and effectively capture system dependencies and relationships. The framework presented shows how the research objectives are not independent from one another; Objective 4 requires the discretization methods identified in Ob-

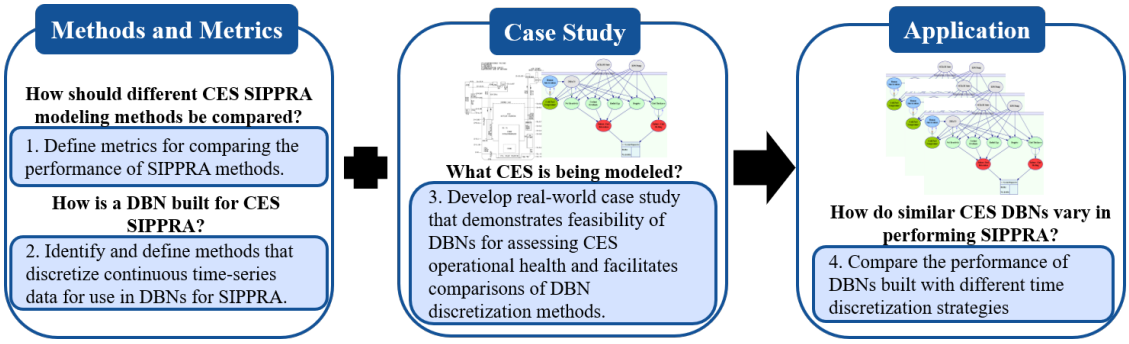


Figure 1.2: The objectives of this research build off one another and allow greater insight into the impact that different discretization methods have on the performance of DBN models used for SIPPRA health management.

jective 2, the case study developed for Objective 3, and the comparison metrics from Objective 1. The outcome of this research is a greater level of understanding of how to develop models for SIPPRA health management that respond to changes in system operations and effectively capture system dependencies and relationships.

1.3 Research Methodology and Data

The objectives described above outline a general process for the research presented in this dissertation; to meet these research objectives, four distinct research activities were carried out.

1. The first activity was to define system-level metrics that could be used to compare the performance of different models built for SIPPRA health management. First, a preliminary literature review on methods for evaluating PRA and PHM model performance was performed. Next, a structured process was developed to identify performance metrics as indicators of successfully com-

pleting SIPPRA processes. The resulting set of metrics were then verified by discussions with experts in the PRA and PHM communities.

2. The second activity was to identify and define the methods for data stream discretization. This was done in two phases. First, an initial literature review was performed to find previously defined methods and identify current gaps in discretization strategies and applications. Specific attention was paid to the discretization methods used to convert system data into usable model time steps. Following the literature search, different discretization methods were identified and categorized based on common traits. The second phase involved developing a new discretization method for DBNs used in SIPPRA health management that minimizes weaknesses identified in current practices for discretizing operational data into model information found in the initial literature review. After reviewing the literature on how to formulate DBN time steps, a framework was developed for an adaptive DBN data stream discretization method that provides responsive insight into a system's health.

3. The third activity was to define a case study to effectively study the performances of DBNs constructed using the different data stream discretization methods. The case study used in this research consists of a simulated nuclear reactor as an example of an operational CES with sensor and other system parameter data collected over an extended period of time. A DBN structure was developed for SIPPRA health management based on the causal relationships within the case study system. This structure only contains the nodes

and the arcs within the model; in order to develop the CPTs needed to enable insight into the system's health, code was developed to take operational data and convert it into the conditional probabilities for different state transitions within the system. The conditional probability table (CPT) quantification code was then run and tested for validity.

A simulated operational data set from a sodium fast reactor (SFR) experiencing an accident was used as the source material for the DBN models generated and compared throughout this research [13]. This data set comes from Sandia National Laboratories (SNL) and is similar to the data used by Groth et al. in their work on dynamic risk-informed accident diagnosis procedures [10]. The data provides information on the thermonuclear, thermodynamic, and physical interactions occurring within the SFR as it faces varying degrees of SCRAM failure following a transient overpower event (TOP).

Further information about the data set including the operational context and the data structure is provided in Chapter 5.

4. The last activity was to perform the comparison and analysis of the different data stream discretization strategies identified in the second activity. Using metrics selected from the set generated in the first activity, metric values associated with different DBNs built based on the case study in the third activity are compared across models and discretization strategies. The results from this comparison indicate the impact that different discretization methods have on DBN model performance and provide insight into effective DBN model

design for SIPPRA health management of CESes.

1.4 Summaries of Contributions and Key Results

Completion of the four research activities generated the following results and contributions to the research objectives:

1. In studying the literature on model performance metrics, it was found that the categories of metrics used to evaluate traditional PRA models and PHM-based assessments for components and simple engineering systems do not provide a complete picture of model performance when applied to CES health monitoring models. To bridge this gap, a new list of model performance metrics was built using a structured taxonomy for identifying metrics and metric classes that should be considered when designing and selecting health monitoring models. An approach for identifying performance metrics was proposed and implemented based on the output of each phase outlined in the SIPPRA monitoring framework by Moradi and Groth [9]. The characterization, cross-validation, and verification of the resulting list of performance metrics support their use in comparing SIPPRA models.

   The primary contribution for Objective 1 is a proposed set of thirty-five performance metrics presented as part of an article accepted in *Reliability Engineering and System Safety (RESS)* for use to compare health-monitoring models of complex engineering systems [14].

2. A survey of the reliability engineering literature indicated that researchers pri-

marily select a time-based or a parameter state-based approach for generating data stream time slices depending on the nature of the study; however, there are other methods that have yet to be utilized that can take advantage of the dynamic changes in system parameter states to provide a clearer picture of system behavior. Combining both approaches in a model would generate meaningful system accident data that could increase system health assessment capabilities while also limiting the model's computational burden. To that end, a hybrid time-based approach that utilizes aspects from both time- and state-based discretization methods was defined to fill the gap.

The primary contributions for Objective 2 are the detailed processes for utilizing three techniques (time-based, state-based, and hybrid time-based) for discretizing continuous-time data streams into discrete-time slices published in conference papers for the 29th European Safety and Reliability Conference (ESREL)[15] and the 2021 Reliability and Maintainability Symposium (RAMS) [16].

3. A case study was developed to analyze the performance of SIPPRA model designs. Using simulated nuclear accident sequence data from a TOP event within a SFR, the network for a joint diagnostic- and prognostic-focused DBN model was constructed from the reactor's operational data and accident scenario conditions. The output of the model is a posterior estimate for the overall health of the system, the nature of the accident, and potential reactor outcomes. Simple model verification results and observations on the model

construction process indicate how such a DBN model helps provide system health prognostic and diagnostic capabilities for monitoring a CES's health.

The primary contribution for Objective 3 is the formation of a real-world case study published in an *Algorithms* article that can be used to further study the impact that different modeling design choices have on CES health-monitoring capabilities [17].

4. The results and analysis of comparing fifty-six DBN models built using the three previously defined discretization approaches across three performance metrics (assessment accuracy, model construction costs, and average information content) provided further insight into the impact that model design has on the performance of DBNs for SIPPRA health management. The range of the values provided by these metrics indicate that there is no single performance metric to use when considering the appropriate discretization approach. The state-based models studied had the shortest construction time, but were susceptible to missing accident sequences when collecting operational data. Time-based models had the most similar results to the underlying safety assessment, but were constrained by computational requirements to construct them. The novel hybrid time-based models introduced in this study provided comparable accuracy to the time-based models, while providing further information content. These findings create opportunities for trade-offs based on different risk model user preferences, needs, and requirements.

   The primary contribution for Objective 4 is further understanding into how

different modeling design choices have on CES health-monitoring capabilities and will be partly submitted in a conference paper for the 2022 Probabilistic Safety Assessment and Management Conference (PSAM 2022)[18] and as part of an article to RESS on the discretization comparison study and results [14].

1.5 Dissertation Overview

- Chapter 2 provides the necessary background information required for conducting the research in this dissertation. This chapter is a collection of short explanations of topics relevant to developing models for enabling system-level health management in CESes including: describing CESes and their characteristics; the functions of PHM, PRA, and SIPPRA for health monitoring and the challenges when applying those techniques to CES health management; and DBNs and their role in studying the reliability and risk management of CESes.

- Chapter 3 presents the results from the first research activity, defining system-level metrics that will be used to compare different time segmentation strategies. This chapter proposes a structured taxonomy for different model performance metrics and metric classes that should be considered when designing and selecting health monitoring models.

- Chapter 4 presents the results from the second research activity, identifying and defining methods for data stream discretization. Following a discussion of the preliminary literature review on discretization strategies, this chapter

presents different model design alternatives from which model performance comparisons can be made.

- Chapter 5 presents the results from the third research activity, developing a case study to effectively study the performance of SIPPRA model designs.

- Chapter 6 presents the results and analysis from the fourth research activity, performing the comparison and analysis of the different time discretization strategies using the case study described in Chapter 5.

- Chapter 7 provides a summary of the dissertation, its contributions, suggested work to further this line of study, and the potential impact of this research on improving the understanding of model design choices on health monitoring models for CESes.

Chapter 2: Fundamental Background Information

This chapter provides necessary background information required for conducting the research outlined in Chapter 1. This includes a detailed description of CESes, different methods of CES health management, and DBN structures and their applications in CES health management.

2.1 CESes

As represented in Figure 2.1, the defining feature of a CES is its composition of interdependent human, hardware, and software components. Each component is considered critical for maintaining system functionality. With such a broad definition, CESes include nuclear power plants, chemical processing facilities, and transportation infrastructure. These systems primarily differ from simple (comprised of only hardware components) or complicated (comprised of hardware and software components) engineering systems in that they rely on human involvement to maintain their functionality; as a result, the logic structure behind CESes is usually more intricate than either complicated or simple systems.

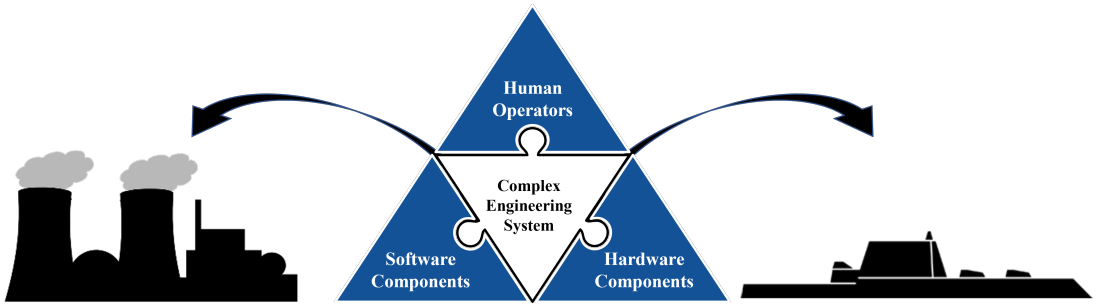


Figure 2.1: CESes are comprised of interconnected human, hardware, and software components. A failure of one of these components affects the performance and, ultimately, the reliability of the system.

2.1.1 CES Characteristics

Table 2.1 summarizes characteristic differences between CESes and other mechanical systems [16]. Collectively, these make it challenging to provide a structured approach for CES health management. One obstacle is evaluating CES system health. The failure of a CES component does not necessarily mean complete system failure; some of these new systems configurations may be considered operational, while others may ultimately lead to a system failure quicker than expected. Additionally, the presence of more varied components means CESes face a wider range of failure modes that could occur and degrade their functionality. Identifying expected failures can lead to improved measures for preventing or preparing against those outcomes, but assessing system health as the product of a single component's health is often oversimplifying the problem or wholly inaccurate.

Table 2.1: Differences between simple systems and CESes make it difficult to apply current health management practices to CESes. Modified from Lewis and Groth [16]

System Characteristics	Components and Simple Systems	CESes
Data Sources	Limited number of sources	Multiple sources
Flow of System Information Rate	Regular data intervals	Varying data intervals
Operational Environment	Static	Dynamic
Interdependencies	None/Assumed independent	Integrated dependencies
System Failure Modes	Limited	Multiple
System Uniqueness	Multiple versions	Unique
System Failure Consequence	System-level	Wide-ranging/ Catastrophic
Downtime Cost	Limited	Expensive
System Behavior Uncertainty	Usually low	Usually high

2.2 Different Methods for CES Health Management

2.2.1 PRA

Since a direct measurement of CES health is challenging to evaluate, a common approach for system-level CES health management is through monitoring the risk of system failure. The premise behind PRA as a form of health management is that methodically evaluating different forms of risk that the system is exposed to provides indirect insight into the health of a system by its ability to mitigate or prevent those risks [2]. This process takes the form of well-defined logic models where the risks are properly identified. Traditional PRA approaches to modeling complex engineering systems rely on well-defined logic models, such as FTs and ETs, where the risks are properly identified to evaluate the likelihood of accident scenarios, system failures and their consequences. Although traditional risk management models present off-

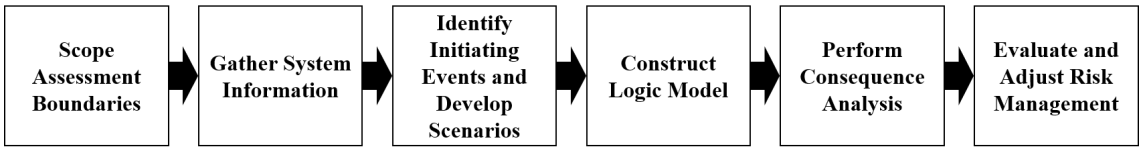


Figure 2.2: General process for PRA derived from Moradi and Groth [20]

line and static representations of system failures without considering the system's current operational state, the availability of operational data provides opportunities to develop dynamic models that can adjust their assessment of system risk based on system information [19]. Ultimately, this can provide more meaningful assessments for decision-makers.

The general process for conducting PRA is shown in Figure 2.2 [9]. First, system information is gathered following the definition of the boundaries for the risk assessment. That information is then used to identify initiating events of interest, develop scenarios to study, and then construct a logic model of the risk space. The results of a consequence analysis based on the contents of the logic model are then used to inform system health management decisions in the context of mitigating or preventing different risks to the system.

2.2.1.1 PRA Limitations in CES Health Management

PRA provides an opportunity to capture potential failures of a system and their likelihoods in a systematic and rigorous manner; however, its current structure is not ideal for on-line health management. It is difficult to validate risk models as they would be tested only on scenarios that have happened. An additional challenge for PRA models is quantitatively measuring their performance; that is, providing

an evaluation of models and their outputs. In their search for system safety research validation, Rae et al. [21] found that risk assessment performance was absent in the literature. Mosleh [22] acknowledged that they may not be verifiable with statistical evidence, PRA models can still provide credible numerical values. Furthermore, most PRA models are limited in their ability make on-line adjustments to model parameters based on new system data. FTs and ETs are constructed from a detailed process of developing scenarios and constructing the logic model; a proper inclusion of new data would require a renewed PRA effort for calculating the updated probability of system faults. Collectively, these problems pose a limitation in using PRA as the primary form of CES health management.

2.2.2 PHM

The introduction of system health monitoring and the transition from "fix when broken" prescriptive system management strategies to more "fix as needed" condition-based approaches requires greater insight into the current health state of the system [23]. PHM, closely aligned with the concepts of condition-based maintenance (CBM) and failure detection, identification, and recovery (FDIR), emphasizes health monitoring and system health evaluation through two kinds of analyses [8, 24]. *Diagnostics* evaluate the current health of the system. Assessments of this kind answer the question, "Is the system healthy and how is it functioning?" If there is a problem with the system, diagnostic practices can be used to identify root causes and pinpoint the failed component. *Prognostics*, on the other hand, provide insight

into future system health. Prognostic assessments answer the question, "Given a system's current operational state, what is its expected health in the future?" This requires knowing the current health of the system as well as its expected use in the future. Prognostics can help determine system degradation patterns and the expected remaining useful life (RUL) of the system. Based on this information, system owners can adjust system management policies to improve system availability, reduce maintenance costs, and better plan for maintenance events [25]. As a result, PHM is a popular approach for on-line assessments of manufacturing systems and hardware components[26, 27].

In their review of diagnostic and prognostic capabilities of manufacturing systems, Vogl et al. developed a flowchart for a standard PHM system process [20]. The process, illustrated in Figure 2.3, contains four distinct phases:

- Online Data Collection: Operational data is collected from the system via system sensors and monitors.
- Diagnostics: Collected system data is processed and critical data features are extracted. These features are used to determine the current health of the system.
- Prognostics: Extracted features are input into a model to estimate the RUL of the system. The model is structured on specific operational conditions.
- Evaluate and Adjust System Management: Current and future health states determined during the diagnostics and the prognostics phases are evaluated. Changes are then made to how the system is managed to address these findings.

There are two notable observations from this flowchart. First, PHM allows

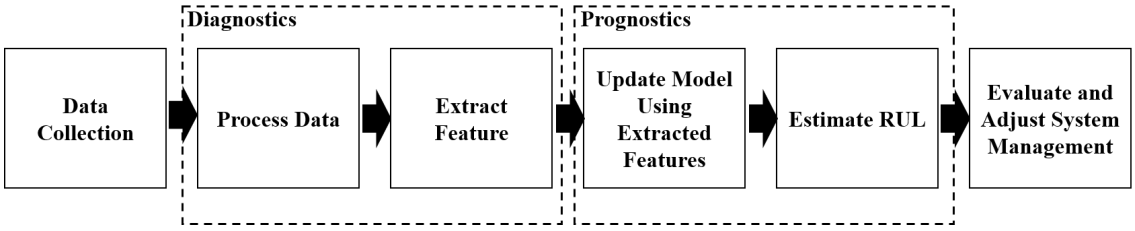


Figure 2.3: General process for PHM derived from Vogl et al. [20]

for multiple approaches for identifying and monitoring system health. The diagnostics phase is designed to determine whether a system is healthy based on available operational data; however, "healthy systems" may be defined differently based on the features extracted. A similar flexibility exists in the prognostics phase and the prognostic model used to determine the future health of the system or its RUL [26]. The second observation is that each step in the PHM process is dependent upon the previous steps. How operational data is processed impacts what features are extracted; likewise, the inputs (i.e., extracted features) into the prognostic model affects the RUL estimated for the system. This suggests that changes made to the PHM process may result in different evaluations of the system's health. Given the importance of system's health for updating and implementing maintenance and risk management decisions, it is important to understand how changes to the PHM process impact these assessments.

Advances in data collection and system sensors has led to implementing novel health monitoring techniques in an effort to apply PHM principles on CESes [17, 28, 29]. Weber and Jouffe [30] modeled the reliability of complex systems with an object-oriented approach by identifying relationships within the subsystems and expanding them to the system level. PHM strategies have progressively incorporated

more data from various sources. [31]. In 2020, Li, et al. [32] created a systematic methodology for defining and designing PHM for aircraft maintenance. While some methods rely on machine learning techniques to utilize available system data, others rely primarily on expert knowledge. Zio and Di Maio [33] approach dynamic failure scenarios with fuzzy on-line estimations of the RUL for nuclear plants. For these lines of research, however, the goal was to identify the future state of the system, rather than to diagnose the current system health.

2.2.2.1 PHM Limitations in CES Health Management

Previous PHM research has primarily focused on components and subsystems as there is more life data available for these smaller systems and it is easier to assume that their components behave independently. For these reasons, PHM techniques often rely on data generated from either a single sensor or sensor type. As mentioned earlier, monitoring CES health is more difficult as these systems have multiple integrated components functioning together and subsystems forming a complicated network of dependencies and common-cause failures. Data from a single sensor is not sufficient to provide an accurate depiction of system health; as a result, prognostic techniques for complex systems require the fusion of various data types and sources. Despite these challenges, the concept of PHM has promise for supporting CES operations as many of these systems are critical to maintain, costly, and potentially harmful if not functioning correctly. This is shown by Muller et al. [34] through their their prognostics model designed to support large industrial maintenance.

Different research efforts address the gap of rigorous system-level PHM for CESes by approaching the challenge from different perspectives. Moradi and Groth [9] propose a framework for integrating PHM advances with the system-level perspective provided by PRA. This relies on understanding the system's structure through the use of fault trees and other logic models. In his depiction of PHM for complex systems, Zio describes a "distributed intelligent dynamic maintenance management system" for predictive maintenance [35]. Such a system would utilize available data from sensors interacting and communicating through an Internet-of-Things. Another approach is the use of hyper-fidelity models to monitor system health. The "Industry 4.0" mindset [36], which focuses on increased mechanization and automation of health monitoring systems, has led to the design and development of "digital twins": precise virtual representations of a system that can replicate the system's physics, behavior, and logic structure [37]. These models take advantage of physical and virtual data and can be modified in real time to optimize system maintenance strategies. In a review of the state-of-the-art of digital twin technology, Tao et al. [37] found digital twin research was often applied to PHM for aircraft manufacturing systems, and wind farm turbines. Collectively, these research studies attempt to scale up PHM capabilities to meet CES health management needs.

2.2.3 SIPPRA

Another approach for addressing the current gap in CES health management capabilities is to systematically integrate aspects of PHM and PRA into the health

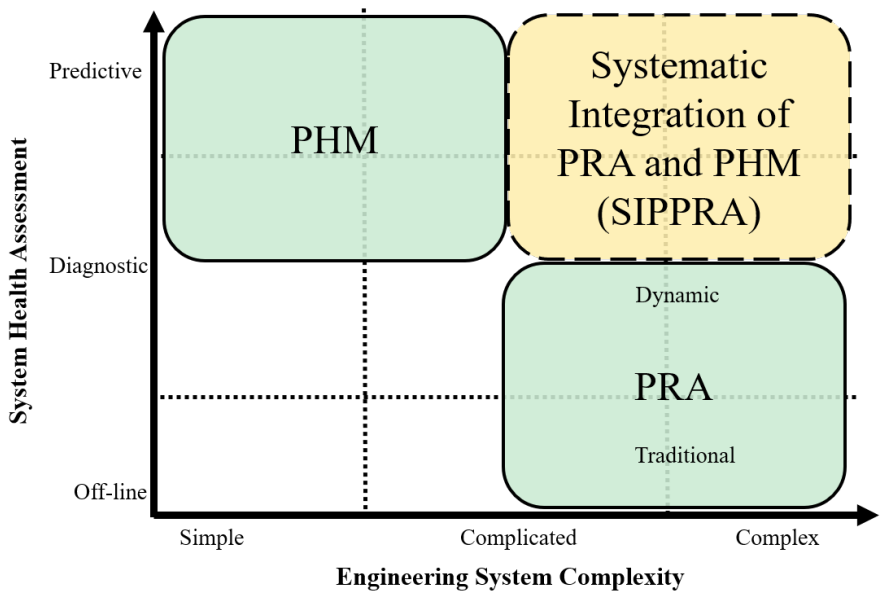


Figure 2.4: SIPPRA allows for PHM for more complicated systems and predictive PRA that lie outside of current health management practices.

assessment. Interests in scaling up PHM for larger systems and the introduction of dynamic and forecasting elements into PRA have led to the development of system-level models placed in the top-right region of Figure 2.4. SIPPRA ties these two fields together and provides a structured form for consistently utilizing available techniques and practices for monitoring, measuring, and evaluating system health across PHM and PRA. This has largely taken the form of PHM models providing input information for PRA models [10, 38, 39] or a PHM model taking the logic structure usually used in PRA models [40, 41]. Recognizing the need for a unified approach to combine PHM and PRA, Moradi and Groth [9] outlined a structured SIPPRA framework, shown in a simplified form in Figure 2.5, for monitoring complex engineering systems. In their approach, a dynamic risk assessment framework identifies the system-level faults before incorporating online system data to perform

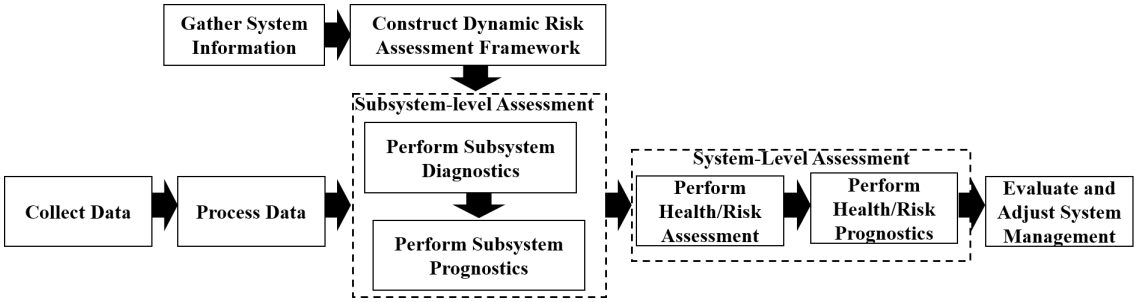


Figure 2.5: General process for SIPPRA derived from Moradi and Groth [9]

health evaluation. System health management decisions made using this structure take a holistic view of the system while utilizing available and relevant data.

2.2.3.1 SIPPRA Limitations in CES Health Management

The biggest challenge facing SIPPRA health management is its novelty and limited use. Although there are multiple research efforts underway to model CES health using a mix of PHM and PRA techniques, it has yet to be widely applied in industry settings to support system management. This means that there are many questions left unanswered regarding effective means for representing CESes, including how to appropriately incorporate system-level data into the health models. Although there are many techniques for assessing CES health through SIPPRA, the remainder of this research will focus on one potential modeling method: the Dynamic Bayesian Network.

2.3 DBNs

DBNs are an extension of BNs, directed acyclic graphs that describe conditional probability relationships between dependent nodes connected by arcs. BNs

hold the Markov property in which only the direct parents of a node have an impact on the state of that node [42]. In a literature review of BNs in fault diagnosis research, Cai et al. [12] indicates that for a given BN with X_n variables, the underlying probability that a certain scenario would occur, P, is based on Equation 2.1:

$$P(X_1, X_2, \ldots, X_n) = \prod_{j=1}^{n} P(X_j | parents(X_j)) \tag{2.1}$$

where $parents(X_j)$ is the set of nodes with arcs into the variable X_j. This allows BNs to model the probability of certain system conditions as a joint probability across the dependencies captured in the model. The type of BN dictates whether the marginal probabilities used in the network are static and describe constant relationships or dynamic, in which they vary over time. The latter models are referred to as DBNs and provide a more accurate relationship for complex systems with time-dependent attributes and parameters.

Like static BNs, DBNs share the same overall structural relationship between nodes over time; however, time-dependent relationships are now included in the model. DBNs are discrete-time models, meaning they work at specified points in time rather than a continuous timeframe [42]. These models are further classified by the number of time slices, also known as intra-slices or time steps, needed to fully describe the model's temporal relationship. The assumed DBN structure is a two-time slice Bayesian Network (TTBN) in which two time slices are needed to express the modeled system, although DBNs with more complex temporal relationships exist

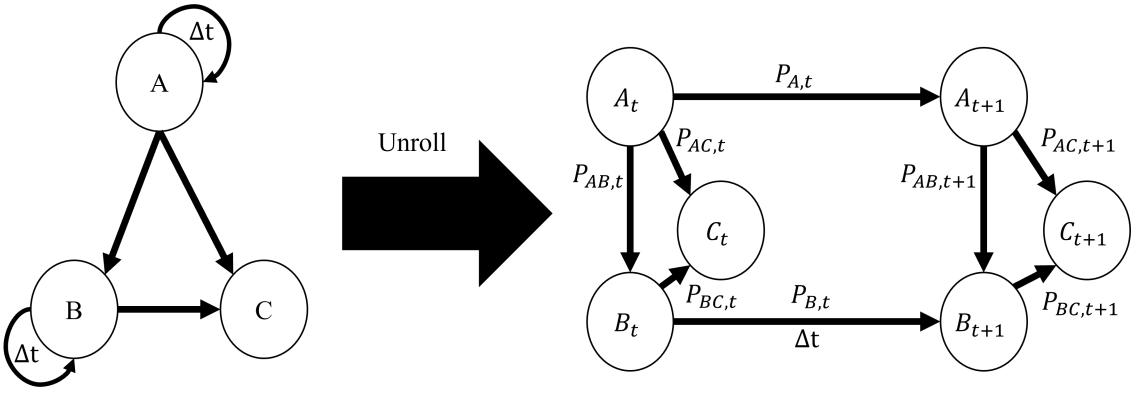


Figure 2.6: Rolled (left) and unrolled (right) two-time sliced DBN

[43]. Specific node relationships are more visible when the DBN is presented as an unrolled BN, as shown in the right image in Figure 2.6.

2.3.1 DBN Inference Capabilities

DBNs are effective in calculating inferences on the node states that are not otherwise easily observable. Using Bayes' Theorem (Eq. 2.2),

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.2}$$

the following inference techniques are possible with DBNs:

1. *Prediction*: Given information about the system's past as well as current system information, what is the expected state of a model node in the future?

2. *Filtering*: Given information about the system's past, what is the expected current state of a model node?

3. *Smoothing*: Given current system information and some information about

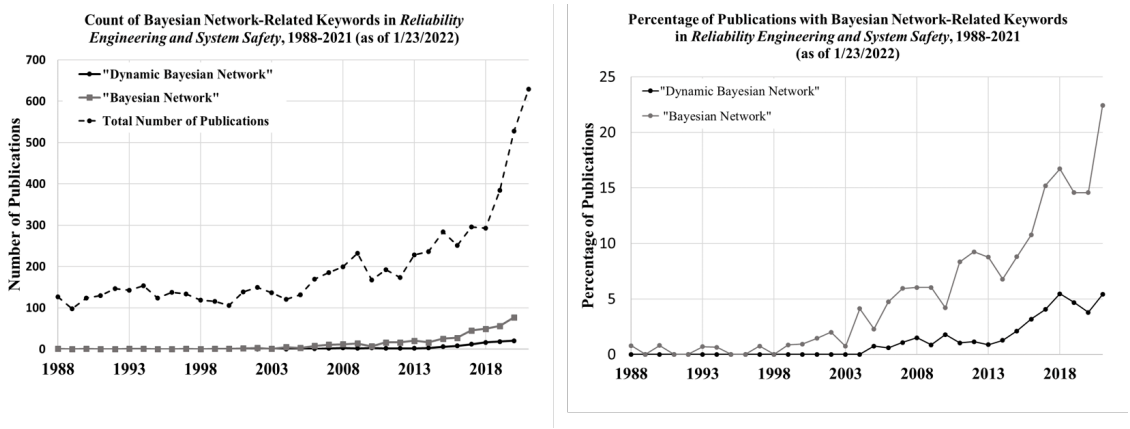


Figure 2.7: The use and frequency of BNs and DBNs in engineering literature has been steadily increasing. Source: *Reliability Engineering and System Safety* Publication Database, Elsevier, Jan 19 2022.

the system's past, what was the most likely state of a model node from an earlier time step?

These different methods allow DBNs to be used for a wide range of system monitoring and health management applications [44].

2.3.2 DBNs in CES Health Management

DBNs are increasingly used in prognostics modeling and risk assessments for CES health management for their graphical representations of complicated causal relationships and powerful inference capabilities [12]. Lewis and Groth [15] found in their literature search on the use of BNs in reliability research that the number of articles related to DBNs published per year has been steadily growing since 2012, as shown in the charts in Figure 2.7. These include studies related to structural engineering (e.g., [45, 46]), mechanical engineering (e.g., [27, 47], and risk and system safety (e.g., [11]). In these studies, the CPTs and initial value distribution used to

parameterize the networks are calculated from available data or determined through expert-based opinions. A DBN's logic structure and inference capabilities makes it a common alternative for causal-based system-level research. The growing interest in using DBNs to solve reliability problems places additional motivation to create models that are effective and efficient in their inference capabilities.

Risk-focused and reliability engineering studies have shown the versatility of DBNs for capturing system reliability and monitoring system health. Early research connected DBN formalisms to reliability block diagrams [48], dynamic fault trees [49], and Markov Chain models [50]. As part of their extensive literature review on the use of Bayesian networks for fault diagnostics, Cai et al. [12] found that DBNs were used to support specific areas of reliability engineering research, including process, structural, and manufacturing systems. Amin et al. [51] used DBNs to determine a dynamic availability assessment of safety critical systems, Wu [52] found that DBNs could be used to make safety decisions for tunnel constructions, and Rebello et al. [53] relied on Hidden Markov Models to monitor system functionality through DBNs. These researchers wanted to capture the dynamic qualities that would otherwise not be accessible to static models. There has also been some research into whether DBNs could be used for system health prognostics. Medjaher et al. [54] represented a small industrial system through DBNs to determine the expected prognostics of the system, and Zhao et al. [55] proposed the use of DBNs to monitor fault diagnostics and loss-of-coolant accident progression prediction in a High Temperature Gas Cooled Reactor Pebble-bed Module reactor. In each of these instances, an emphasis was placed on providing either system health prognostics or

diagnostics; there has been limited effort taken to combine this information into a single model.

DBNs have also been implemented in risk management studies. Kohda and Cui [11] proposed using DBNs to model and support a risk-based reconfiguration of safety plant monitoring systems. Khakzad [56, 57] showed the applicability of DBNs in capturing performance assessments of fire protection systems in chemical plants. When connected with other risk analysis tools, like FT or ETs, DBNs can provide meaningful and up-to-date system-level insight. This is evident from the risk management research coming from the nuclear sector [55, 58]. Groth et al. demonstrated this by showing how DBNs could be applied for risk management of CESes as part of a process for providing risk-informed diagnosis procedures[10]. However, there has not been a significant push to merge risk-informed DBNs with the system health of a CES to generate insight into the current health of the system.

Despite their usefulness in providing system-level assessments, DBNs face a few obstacles when modelling CES health. One noticeable challenge DBNs face the rapid growth in computational requirements as model size increases. Model growth occurs with the addition of new nodes, temporal arcs, and time steps. Models can expand to provide additional detail in describing a system, further predictive capabilities, or increased precision within a contained time frame; however, this requires additional resources for data storage and computational requirements. A model with an excessive calculation cost is not a practical tool to use in preparation for accident sequences, and DBNs can easily become too costly for operational use.

Another challenge of modelling a CES as a DBN is handling wide-ranging data

streams generated from various sources over source-specific rates. Focusing primarily on data sources with the slowest rates of information propagation results in the loss of key information from sources that have quicker data turnover. Conversely, creating a model that prioritizes rapidly generated information can potentially be unwieldy and computationally expensive. This challenge was recognized by Dean and Kanazawa [59]. In the first article on DBNs, they suggested that multiple models might be the solution to address two or more distinct time rates. Recently, other researchers have attempted to resolve this issue by creating hybrid continuous-discrete-time models, such as Iamsumang et al. [60], or even rejecting discrete model and pursuing a continuous approach, like Codetta-Raiteri and Portinale [61]; at this time, however, discrete-time models remain attractive as they provide efficient ways of assessing fault and probability issues outside of a continuous time-space.

2.4 Chapter Conclusion

DBNs represent complex time-dependent causal relationships through conditional probabilities and directed acyclic graph models. These models enable the forward and backward inference of system states, diagnosing current system health, and forecasting future system prognosis within the same modeling framework. As a result, there is interest in applying DBNs to model CES health, but there are open questions on how the decisions about DBN structures and data stream discretization can impact CES health management. These systems' tightly integrated human, hardware, and software components and dynamic operational environments

have previously been difficult to model using traditional PRA and PHM approaches, and SIPPRA techniques have mostly been applied to experiments and test cases. There is a need to further study DBN model designs of CESes for SIPPRA health management.

This work advances the understanding of how different data stream discretization techniques for selecting the data used to develop and update DBNs for SIPPRA health management can affect model performance. However, this first requires a set of clear metrics for evaluating system-level model performance.

Chapter 3: Metrics to Evaluate System-level Model Performance for SIPPRA Health Management

There is increasing interest in expanding the scope of PHM and PRA system modeling to include CESes [9]. Scaling-up PHM for larger systems and the introduction of dynamic and forecasting elements into PRA have led to research efforts from both the PHM and PRA communities to experiment with the feasibility of health monitoring in this space through SIPPRA [9, 19, 62]. Since SIPPRA research has been primarily exploratory, it is unclear how the performance and predictive capabilities of these models should be assessed and compared to other model designs.

As part of the growing effort to better understand how to integrate PHM-type techniques with traditional PRA practices to provide improved SIPPRA health monitoring for CESes, a structured taxonomy is proposed for generating different model performance metrics and metric classes to consider when designing and selecting health monitoring models. Metrics identified in the taxonomy were systematically developed based on the different steps of SIPPRA health management for CESes and rigorously verified using multiple forms of metric verification. Comparisons between performance metrics measured from two model designs provide insight into

which proposed model is more appropriate for use as a full-scale CES system health monitoring model. The result of this research extends model design decisions beyond assessment-based performance metrics, like accuracy or precision, to include other relevant information about CES health monitoring models, such as information about model inputs or structure. Expanding the definition of performance for health management models addresses a specific challenge to PHM identified by Zio [63] in his summary of directions for PHM advancement.

This chapter, accepted as a paper to *RESS*, first describes the categories of metrics used to evaluate traditional PRA models and prognostic and diagnostic assessments for components and simple engineering systems, as well as the limitations they face when monitoring CES health [64]. To reduce this gap, an approach for identifying performance metrics is proposed based on the output of each phase outlined in the SIPPRA framework by Moradi and Groth [9]. A proposed set of performance metrics is then introduced for use to compare CES health-monitoring models. This is then followed by an illustrative example of using selected metrics for evaluating a simple model design decision. After presenting research implications for using such a set, the chapter ends with a summary conclusion.

3.1 Characteristics of Model Performance Metrics

Model metrics are tools for converting model characteristics or parameters into usable information for decision-making. These decisions can range from selecting model designs to model implementation practices, and model revisions. Metrics

are not only tied to the model's objective (what the model is supposed to do), but also to its performance in accomplishing tasks that meet that objective. As such, they can provide useful insight into the model's overall functional effectiveness and efficiency.

Models designed for system-level health monitoring are structured to detect degradation or anomalous behavior that might result in future system failures [65, 66]. By tracking system behavior and notifying system operators when these operational state thresholds are crossed, these models reduce the likelihood of unforeseen system failure and, as a result, the risk of resulting consequences. The performance of these models could, therefore, be identified as the extent to which they reduced the risk of system failure. However, this is a particularly challenging value to assess, and previous attempts to measure this have relied on other performance metrics as a surrogate or an indicator of potential performance. Although there are multiple model features that could be considered for use as metrics, it is important to properly identify those that provide meaningful information to the decision-maker who must determine how a model should be designed, which version of a model should be used, or even whether to use one at all.

3.2 Performance Metric Classes for PRA Models

As mentioned in Section 2.2.1.1, one challenge for PRA models is quantitatively measuring model performance and providing evaluations of models and their outputs. It is clear from previous considerations of PRA performance [21, 22] that

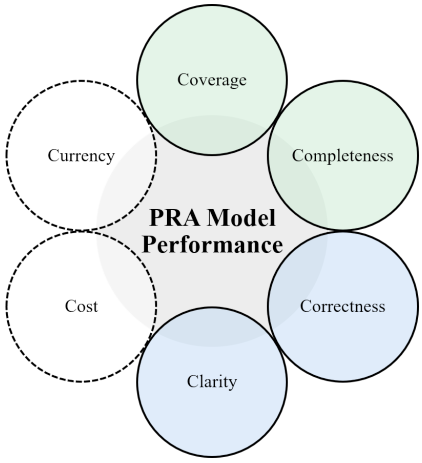


Figure 3.1: Metrics for evaluating static and off-line PRA model performance fall into six metric classes. Green classes address design choices, while the blue ones capture model output values. The white dashed categories should be considered if information collection is included as part of the model-building process.

a model performance metric should not just be limited to defining the specific risk value output, but could also help evaluate the design and data structures underlying the model used to perform the assessment. For that reason, comparative performance metrics for traditional PRA models should fall into the following six "C" evaluation classes represented in Figure 3.1.

- *Coverage* describes the scope of the PRA. This includes what system components, failures and accident sequences are modeled in the assessment. In terms of FTs and ETs, coverage captures the number of high-level system failures and initiating events considered, respectively. A coverage metric, like "Number of components included" or "Level of system abstraction," would help to provide a comparison for which context the PRA model would be applicable.
- In part, PRA *completeness* expresses the granularity of the risk model [22]. It may also provide insight into the depth of causality expressed in the model. Completeness ties in the uncertainties expressed within the model and its

parameters [67]. As such, metrics in this class include "Number of event classes" as well as "Range of risk output's confidence bounds."

- *Correctness* for a PRA model is the alignment of its risk output value with the result of the actual scenario occurring. Sometimes termed "conformability," this is approximately the accuracy of the model and is probably the first aspect of performance considered when comparing model designs.
- *Clarity* in a PRA model is necessary for explaining not only the model itself but also its results. Risk communication is a significant hurdle for PRA [2, 21], and it is important to be able to evaluate and understand what the insights given by the model are providing and why. Traceability is a significant factor in model clarity and helps to validate a model as being relevant for assessing system risk. Work by Johansen et al. [68] on the use of different risk metrics illustrates the importance of choosing the appropriate values. One metric in this class could include "Level of user training required."
- There are two additional performance metric categories to consider when the PRA model design includes system information collection. The *currency* of system information will help determine the age of the information used to inform the model design. Up-to-date information may provide more emphasis on accident scenarios that provide more risk now than before. The overall *cost* of this system data collection, in terms of time and resources, should also be considered as well.

The evaluation classes address different aspects of PRA model performance. While these categories may not equally contribute to a performance assessment,

combined, they create a fuller picture of model performance than a solely model-accuracy-based approach. Metrics that measure coverage and completeness classes address the model design process. Correctness and clarity metrics provide insight into the risk value output of the model, and, if applicable, the cost and currency metrics address system data collection. As traditional PRA models are static, most of the metric classes focus on designing the model rather than its implementation. This limits the extent to which PRA performance metrics can be used to make up-to-date decisions on previously developed risk models. It should also be noted that although the nature of PRA tends to lead towards more qualitative rather than quantitative performance metrics, these metrics can still provide valuable insight into the model.

The introduction of dynamic PRA into risk management provides opportunities for models to interact with operational data to provide up-to-date risk assessments [19]. Other metrics from another area of health monitoring, PHM, can provide some insight into what metrics may be useful in comparing these dynamic PRA models.

3.3 Performance Metric Classes for PHM Models

Previous efforts to identify PHM performance metrics have largely focused on evaluating a model's diagnostic and prognostic capabilities. Diagnostics models identify health by classifying processed operational data as indicative of either a "healthy" or "not healthy" system. The primary approach for assessing the effec-

tiveness of a diagnostic method is to compare the assigned classification value to the actual state of the system. Metrics that evaluate a model's performance of binary diagnostic classifications include commonly used metrics like sensitivity, specificity, and area under the receiver operating curve [8, 24, 69]. These measures of diagnostic performance are based on the model's ability to distinguish between and identify healthy system states from unhealthy ones. Binary classifications are commonplace, and these measures are used in a range of performance evaluations, including for sensors and other monitoring equipment [8, 69].

Prognostic-focused models use features extracted from the diagnostic phase as input to forecast the RUL of the system. Common evaluations on a model's prognostic performance focus on verifying and validating prognostic models based on how similar the prediction is to the actual lifetime of the system [70]. In their research on PHM metrics, Saxena et al. [24] grouped similar metrics, like accuracy, robustness, precision, and convergence, as measurements of algorithm performance. They found a number of different criteria, including the ease of algorithm certification, computational performance, cost-benefit-risk, and algorithm performance, that one could consider when comparing prognostics model performances.

In addition to prediction-accuracy driven metrics as the primary tool for evaluating model performance, there is also another set of metrics based on implementing the model itself. In their evaluation process for model-based prognostics tools, Atamuradov [71] classified criteria parameters into two overarching categories: general model context (required expert knowledge, model structure, and data characteristics) and tool efficiency (run time, accuracy, robustness, prediction horizon, and

learning time). Zeng et al. [72] created a multi-layer decision-making framework for assessing the prediction capability based on prognostics performance indicators that fell under the quality of the RUL prediction and the trustworthiness of the method. Zio [63] further argued that metrics like model output interpretability can be just as powerful for evaluating model's performance as the accuracy of the model itself.

The range of model performance categories identified in the literature highlights the multi-objective decision-making process required for selecting a specific PHM approach. A certain level of performance is balanced with the physical and computational restrictions of developing a PHM model. This process requires a careful analysis of the costs and benefits for each alternative capability [28, 69]. Additional factors to cover when determining appropriate system monitoring performance techniques include "time for problem mitigation," the "cost of mitigation," the "cost of failure," and the "uncertainty management capability" of the system [24]; these are very similar to the risk factors identified in PRA procedures. Each of these identified criteria are system-specific and require a baseline understanding of the system, its failures, and available resources for system monitoring and management.

3.4 Approach and Methodology

When comparing different methods for modeling system health, it is important to have a set of metrics that can be used to determine whether some modeling approaches provide the necessary system health information within the needed time

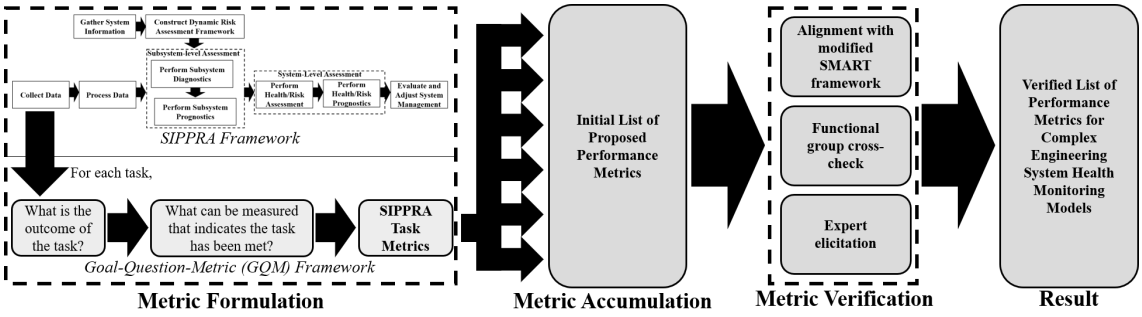


Figure 3.2: Visual representation of the generation and verification of a comprehensive list of SIPPRA model performance metrics for CESes.

period based on the right information. Although CESes can be extremely unique, having a set of applicable metrics that could be used to make comparisons across a range of modeling approaches would make selecting model design choices more systematic and aligned with the health monitoring requirements for a given system.

Evaluating system-level health monitoring models for complex engineering systems cannot rely solely on metrics previously used to compare traditional PRA and PHM models. A new set of performance metrics is needed to provide meaningful values that can be measured and used to enable improved decision-making in health model design and selection. To generate a comprehensive set of performance metrics for models built for SIPPRA health management, this work followed the approach outlined in Figure 3.2 and described in the remainder of this section.

3.4.1 Metric Formulation

The metrics identified in this research were generated using a modified version of the goal-question-metric (GQM) method for generating metrics [73]. In this approach, relevant metrics are determined by identifying a specific goal (in this

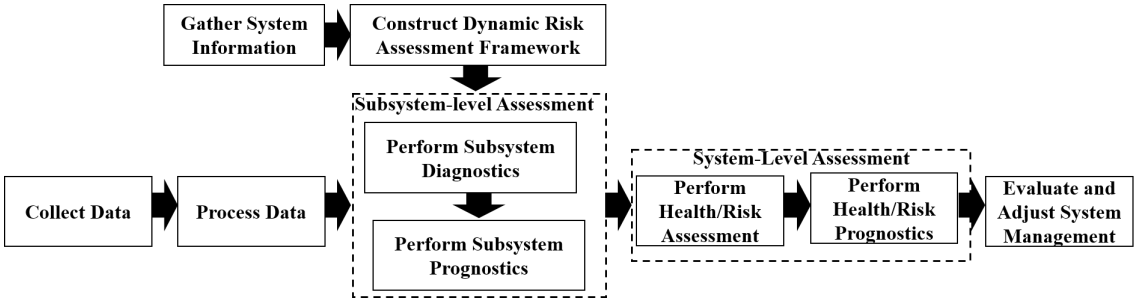


Figure 3.3: SIPPRA process modified from Moradi and Groth [9]

case, the successful performance of a health monitoring model) and what can be measured to indicate that the model is functional or as usable as expected. This work considers the performance of a system-level health-monitoring model by its ability to perform the different health-monitoring tasks identified in the SIPPRA framework by Moradi and Groth [9] in Figure 3.3. This framework was selected as the basis for metric formulation because it provides clear steps necessary for integrating PHM and PRA methods. As the primary output of each task aligns with an expected goal, metrics can be identified as characteristics from which insight about the model's performance in that particular task and overall performance can be gleaned.

3.4.2 Metric Accumulation

Metrics that are generated from following the process outlined above are then compiled to form an initial list of model performance metrics.

3.4.3 Metric Verification

In order to confirm that the set of proposed metrics is comprehensive as well as to verify the usefulness of the metrics themselves for capturing the performance of SIPPRA health monitoring models, a three-pronged approach was used for list verification.

3.4.3.1 Metric Alignment with Modified SMART Framework

A common framework for constructing metrics is the SMART framework whereby metrics are defined to be "specific, measurable, achievable, relevant, and time-based [74]." This research used a modified version of the SMART framework to verify the proposed performance metrics associated with system-level health monitoring models have the following characteristics:

- *Understandable*: metrics should have a clear definition and process for identifying and evaluating them
- *Well-aligned*: metrics should be an indicator of the model's performance
- *Achievable*: metrics should cover a well-defined range of values that are all possible to reach and measure
- *Robust*: metrics should be able to be evaluated in the same manner across time and functionally similar models
- *Timely*: metrics should be available or determined within a sufficient time period for the decision-maker

The changes made to the SMART framework emphasize the required information

quality a decision-maker would need to make to determine model design, selection, and usage decisions.

3.4.3.2 Functional Group Cross-check

Another method used to verify the process of identifying performance metrics for system-level health monitoring models is a cross-tabulation of the metrics to different performance classes. As SIPPRA assumes a systematic integration of PHM and PRA for health monitoring, assessing how the different metrics align with a particular metric class from either approach helps identify whether the proposed metrics sufficiently cover areas identified in PRA and PHM.

3.4.3.3 Expert Elicitation

The final form of metric verification is through structured discussions with seven experts from the PRA and PHM communities; in total, they possess over a combined 175 years of experience in their respective fields. Each expert received a preliminary list of proposed performance metrics that made it through the two previous verification methods and provided his or her thoughts and insight on the ones listed as well as other metrics to consider. The experts' feedback was considered when developing the final output of this process: a comprehensive list of verified metrics for complex engineering system health monitoring models.

3.5 Results: Metrics and their Definitions

This section presents the output of carrying out the process described in Section 3.4: a series of meaningful performance metrics for system-level health monitoring models. The model performance metrics are outlined for each SIPPRA phase; a summary table of the findings from this process are also provided in Table 3.1.

3.5.1 Gather System Information

The first task of designing system-level health monitoring models is to gather information about the system that will contribute to its proper representation. Data collected during this phase may include historical data such as system-level diagrams and maintenance reports. The product of this data collection is further insight and information into the system's failure modes and scenario accidents that may ultimately affect the health of the system.

Four metrics have been identified as relevant measurements of performance for this health monitoring phase. These metrics involve the nature of information gathering and the confidence in the completeness of the system information.

- *Age of system failure information*: the time the system failure information was last collected.
- *Completeness of system failure information*: the granularity of the system data with respect to specific system failures.
- *Cost-effectiveness of information gathering*: the amount of money required to improve one of the other metrics from this task.

Table 3.1: System-level health monitoring phases, their outputs, and relevant metrics

Health Monitoring Phase	**Phase Output**	**Relevant Metrics**
Gather System Failure Information	Informed understanding of system failure modes and accident scenarios	Age of system failure information Completeness of system failure information Cost-effectiveness of information gathering Coverage of system failure information
Construct Dynamic Risk Assessment Framework	Step-by-step guide for utilizing system-level health monitoring information	Ease of model modification for new failure modes Framework completeness Framework coverage of known failure modes Framework explainability Framework traceability Preliminary model construction costs
Collect System Data	Raw current data	Age of operational data Availability of data collection sources Data collection costs per sampling rate Data collection equipment requirements Data coverage Information content per sampling rate Number of inferred data sources Number of operational data sources Operational data redundancy Required data storage capacity per sampling rate Sampling rate frequency
Pre-process Data	Data prepared for subsystem-level assessments	Pre-processing equipment requirements Pre-processing explainability Pre-processing time per sampling Pre-processing traceability
Perform (Sub)system-Level Assessment	(Sub)system-level health assessments	Assessment accuracy Assessment algorithm sensitivity Assessment interpretability Assessment precision Assessment forecast window Response time to (sub)system-level anomalies System state requirements for assessments Time required for assessment
Evaluate and Adjust System Management	Informed system managers and modified (if necessary) system operations	Expertise required for model use Model output interpretability

- *Coverage of system failure information*: the scope of system failures for which data is collected.

3.5.2 Construct Dynamic Risk Assessment Framework

Using the gathered information about the system and its failures, a dynamic risk assessment framework can be developed that incorporates system-level assessments into decision-level support. To that end, the six performance metrics relevant to constructing the risk assessment framework emphasize the framework's structure and usability for decision-makers.

- *Ease of model modification for new failure modes*: the amount of resources (time, effort, material) required to expand the model to respond to new failure modes, either omitted from the previous model design or recently identified.

- *Framework completeness*: the granularity of the framework in classifying specific system failure outcomes.

- *Framework coverage of known failure modes*: the extent of the model's applicability to any known system failure, accident scenario, or degradation.

- *Framework explainability*: the extent of knowledge about the meaning behind the model's nodes and parameters.

- *Framework traceability*: the extent of knowledge behind the values applied to model nodes and parameters.

- *Preliminary model construction costs*: the amount of resources (time, effort, materials) required to construct the model that performs the risk assessments.

3.5.3 Collect Data

The previous two steps are carried out during the model design process. Once the model is operational, the health monitoring model provides assessments based on online data used in monitoring systems and off-line data collections, such as human-based inspections.

There are a number of metrics to evaluate the quantity and quality of the data collection process.

- *Age of operational data*: the time the operational data was collected.
- *Availability of data collection equipment*: the availability of the sensors and other data sources used to measure and capture the raw operational data.
- *Data collection costs per sampling rate*: the cost (time, effort, resources) of collecting data for a given sampling time.
- *Data collection equipment requirements*: the specific equipment and their requirements needed for data collection.
- *Data coverage*: the extent to which system data corresponds to specific elements of the model.
- *Information content per sampling rate*: the expected information entropy for the data collected at a given sampling time.
- *Number of inferred data sources*: how many data sources are based on infer-

ences from other data sources.

- *Number of operational data sources*: how many data sources are used as input into the model to provide information about the system.
- *Operational data redundancy*: how many data sources (sensors, etc) provide the same operational information.
- *Required data storage capacity per sampling rate*: how much data storage is required for the operational data collected at every sampling time.
- *Sampling rate frequency*: the rate data is collected to be used as input into the health model.

3.5.4 Pre-process Data

In most instances, the raw data that is collected is pre-processed before use in the subsystem-level health assessments. The metrics identified from this phase relate to the efficiency in and clarity of the conversion from available operational data to usable data.

- *Processing equipment requirements*: the needs (computational or otherwise) of the equipment processing the raw data.
- *Processing explainability*: the extent of knowledge behind the processing procedure used for the health model.
- *Processing time per sampling*: the length of time required to convert the raw data collected into usable information.
- *Processing traceability*: the extent of knowledge behind the data processing

values applied to model nodes and parameters.

3.5.5 Subsystem-level Assessment

In this phase, processed data is used to evaluate the health of specific subsystem levels. This is the level most similar to previous work on model performance metrics and the prognostics performance metrics research by Saxena et al [24]; as such, there are a significant number of applicable metrics to consider from this SIPPRA phase.

- *Assessment accuracy*: how close the model's assessment is to the ground truth.
- *Assessment algorithm sensitivity*: the extent to which the algorithm providing the health assessment is impacted by changes in operational inputs.
- *Assessment forecast window*: the period in which the health assessment is placed.
- *Assessment interpretability*: the complexity of the translation process between assessment and subsystem health.
- *Assessment precision*: the similarity of model assessments when applying the same operational data.
- *Response time to subsystem-level anomalies*: the time delay between accident or health degradation event and and event identification from the model.
- *System state requirements for assessments*: the operational characteristics required for the health monitoring model to make an assessment.
- *Time required for assessment*: the time delay between data input and assess-

ment.

3.5.6 System-level Assessment

This phase is similar to the process for the subsystem-level; as such it has similar performance metrics, but at the system level. Because system-level uncertainties are particularly important for complex system health management, those performance metrics associated with capturing the output uncertainties, like "Assessment precision," and "Assessment algorithm sensitivity" may be of particular interest.

3.5.7 Evaluate and Adjust System Management

The final phase of the system health monitoring process is to evaluate the assessments provided by the system and to adjust the system management. In some ways, this is the most critical aspect of the model as it leads to system and behavioral changes. Because the assessments from these models may come in different forms (binary classification, point estimate, distribution), it is important to be able to qualify the assessments received and their impact on future health decisions.

There are two metrics to consider for this phase:

- *Expertise required for model use*: the necessary level of user knowledge needed to properly use the model for system management.
- *Model output interpretability*: the complexity of applying model results to system management.

3.6 Illustrative Example of Utilizing System Health Assessment Metrics to Decide Between Models of Different Sampling Rates

The performance metrics for health monitoring models presented in Section 3.5 provide designers with different indicators as to how well or how useful a model is for capturing system-level health. This enables rigorous comparisons to be made for determining the optimal model structure and characteristics.

To show how the performance metrics generated using the methodology presented above could be used, a representative design problem is present wherein different model designs lead to varying values of performance depending on the metric. This example provides insight into the implementation of these metrics to make significant model design decisions.

3.6.1 Example background

For this hypothetical scenario, model designers are deciding what time sampling structure should be included for a Bayesian Network-based health monitoring model. The baseline design uses system data collected at a standard periodic sampling rate r and is similar to the model design structure presented by Lewis and Groth [15], but recent sensor additions have made it possible to utilize data collected at twice the frequency.

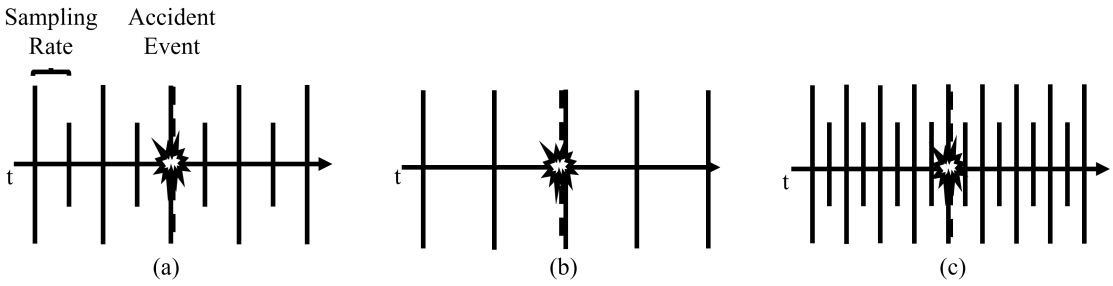


Figure 3.4: One possible decision performance metrics could support is whether to replace a health monitoring model that has the sampling rate shown in (a), with either one that has half the sampling rate (b), or one that has twice the sampling rate (c). A hypothetical accident event is presented to indicate the length of time to the next data sampling.

3.6.2 Illustrated example

A visualization of the different differing sampling rates for the baseline and proposed model designs presented in Figure 3.4 suggests that the shorter sampling frequency would be a better design choice given the shorter time period between an accident event and new data; however, there are other metrics to consider. The model designs are evaluated based on five metrics selected from the list provided in this chapter: sampling rate frequency, required data storage capacity (represented as the function $f(x)$), response time to anomalies, information content per sample (represented as the function $g(x)$), and framework explainability. The values of the proposed designs are compared relative to the values of the baseline model (i.e., a periodic discretization of sampling rate r).

Table 3.2: Example comparison process across three different models designs for selecting a sampling rate. Using model design (a) as a baseline, metrics for the proposed model designs (b) and (c) are first determined and then compare against each other. A model is selected based on the priorities of the model user and the resource limitations.

	Baseline Design (a)	Proposed Design (b)	Proposed Design (c)	Model Comparison
Sampling rate frequency	r	$(1/2)r$	$2r$	$b < a < c$
Required data storage capacity	$f(r)$	$f((1/2)r)$	$f(2r)$	$b < a < c$
Response time to anomalies	$1/r$	$2/r$	$1/(2r)$	$b < a < c$
Information content per sample	$g(r)$	$g((1/2)r)$	$g(2r)$	$c < a < b$
Framework Explainability	Baseline	Baseline	Baseline	$a = b = c$

3.6.3 Illustrated example results

The results of the modeling comparison are shown in Table 3.2. In this example, it is assumed that data storage capacity decreases with smaller sampling frequency, and the information content per sample decreases with greater sampling frequency.

In this simple comparison across models with different data sampling rates, different metrics resulted in different relationships across the model designs. The response time to system anomalies for the proposed model design (c) was shorter than model design (b), but that model design required greater data storage capacity and provided less information content per sample. The baseline model (a) fell consistently between the two proposed model designs. The metric "Framework explainability" had a consistent value across all three designs

3.7 Discussion

The results from Section 3.5 identify a list of performance metrics that may be useful for evaluating system-level health monitoring. This section serves to provide additional detail into how these metrics could be classified, quantified, compared, and used to identify appropriate model designs for CES system-level health monitoring.

3.7.1 Discussion of the Illustrated Example

The illustrative example in Section 3.6 makes it clear that different model designs would lead to distinct metric values. The values can then be used to identify differences between model performance. With this information, model designers can select the design that meets the need and priorities for monitoring the system. Those interested in prioritizing minimizing the required data storage capacity may pick a model with a smaller sampling rate frequency; i.e., more time between data collection as in design (b). On the other hand, if the safety-critical nature of the system would require a larger sampling rate frequency, then model design (c) may be more suitable to implement.

The five metrics selected for this example do not necessarily need to be the only metrics used to support this kind of model structure decision. In fact, for one of the metrics, "Framework explainability," the metric is not affected by the different model design choices; changing how the data is sampled does not impact the framework's structure. Conceptually understanding how the model design decision impacts its

function is a helpful first start in identifying what metrics should be considered for evaluation and assessments.

The evaluations of these models can be done in different ways; in a paper aimed for practitioners to identify proper prognostics tool selection, Atamuradov et al. [71] proposed a metrics selection matrix provided. Given the large amount of metrics available, that process is not recommended for an initial approach.

3.7.2 Classification of Metrics

The metric generation process used in Section 3.5 identified a group of performance metrics based on successfully completing each of the different steps present in the framework for constructing and using a system-level health monitoring model. Sections 3.7.2.1-3.7.2.2 offer methods for classifying these metrics based on different attributes.

3.7.2.1 Alignment with CES Metric Performance Classes

In addition to being categorized by different SIPPRA phases, the performance metrics for system level-health monitoring models can also be classified based on which metric class they provide information. Using the SIPPRA classes found in Figure 3.5, Table 3.3 maps the metrics to distinct metric classes.The process of constructing this table served as one of the verification methods used in building the list of performance metrics.

Table 3.3 provides interesting insight into what type of metrics are prioritized

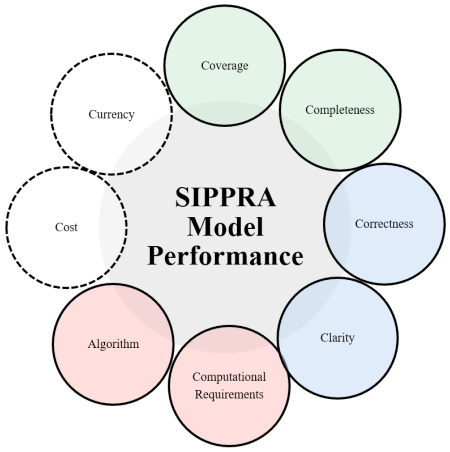


Figure 3.5: SIPPRA models have two performance metric classes beyond PRA models: "Algorithm" and "Computational Requirements."

throughout the SIPPRA process. There appears to be two groups of metric classes: those that are specific to a single or few health-monitoring tasks (cost, coverage, completeness, health modeling algorithm, computational and equipment requirements, and correctness), and those that pertain to all or most of the tasks (currency and clarity). As these tasks can broadly be aligned with traditional PHM and PRA activities, the combination of groups suggests the importance of these metrics in both scenarios. Currency of data and information is particularly important for monitoring the health of the system, and clarity of how and why model designs are structured as they are and the outputs of the model are necessary across the entirety of the process of designing and using system-level health monitoring models. The gaps in Table 3.3 do not mean that there are no performance metrics associated with this space; rather, the table serves to highlight the metrics that are most useful to consider.

Table 3.3: A cross-tabulation of the performance metrics across SIPPRA phases and PRA and PHM classes

Task	**Currency Metrics**	**Cost Metrics**	**Coverage Metrics**	**Completeness Metrics**	**Health Modeling Algorithm Metrics**	**Computational/ Equipment Requirement Metrics**	**Correctness Metrics**	**Clarity Metrics**
Gather System Failure Information	-Age of system failure information	-Cost-effectiveness of information gathering	-Coverage of system failure information	-Completeness of system failure information				
Construct Dynamic Risk Assessment Framework		-Preliminary model construction costs	-Framework coverage of known failure modes	-Framework completeness	-Ease of model modifications for new failure modes			-Framework explainability -Framework traceability
Collect System Data	-Age of operational data -Sampling rate frequency	-Data collection costs per sampling rate	-Data coverage -Number of inferred data sources -Number of operational data sources -Operational data redundancy			-Availability of data collection sources -Data collection equipment requirements -Required data storage capacity per sampling rate		-Information content per sampling rate
Pre-process data	-Pre-processing time per sampling rate					-Pre-processing equipment requirements		-Pre-processing explainability -Pre-processing traceability
Perform (sub)system-level assessment	-Response time to (sub)system-level anomalies -Time required for assessment		-System-state requirements for assessments		-Assessment forecast window -Algorithm sensitivity -Assessment precision		-Assessment Accuracy	-Assessment interpretability
Evaluate and Adjust System Management								-Expertise required for model use -Model output interpretability

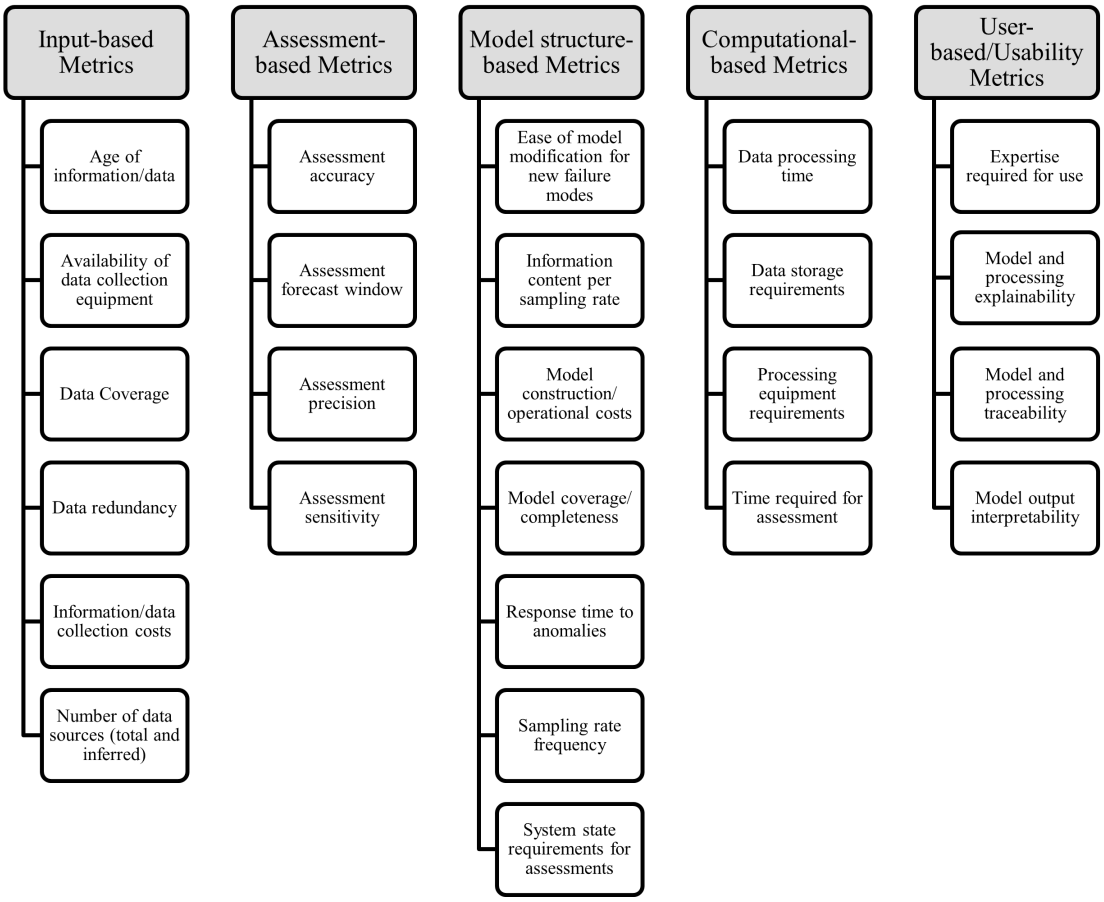


Figure 3.6: Performance metrics classified by model aspects.

3.7.2.2 Functional Classification

These metrics are not isolated measurements at different points along the lifecycle of the health-monitoring model; there are commonalities between metrics across the different phases. For instance, "framework explainability" and "pre-processing explainability" metrics could be consolidated into a general "model explainability" metric. In that sense, it may be better to classify metrics by the attributes to which they provide insight.

A proposed classification and taxonomic structure for CES system-level health

monitoring model performance metrics is presented in Figure 3.6. Health monitoring capabilities are distributed based on the following model attributes:

- Model input: What is the nature of the model inputs?
- Model output (assessments): What is the nature of the model outputs?
- Model structure: How is the model built relative to the system?
- Computational requirements: What computational requirements or limits are there for the model?
- User Involvement: How does the model user interact with the model?

These categories help direct CES model designers to specific model attributes that are relevant and should be prioritized. This is particularly relevant if computational or user limitations are the key inhibiting factors for modeling the operational system. A similar functional comparison for PHM models was considered by Saxena et al [24]; however, the groups identified in that research were based on performance objectives rather than model characteristics.

The re-framing of metrics from a phase-centric approach to a more functional one further blurs the lines between PRA and PHM metrics and solidifies the need for this group of SIPPRA health monitoring metrics. Both communities provide insight into the model input, output, structure and computational requirements. On the one hand, metrics derived from PHM tasks may provide insight for system health and monitoring, but this alone is not sufficient for this level of system health monitoring; there is the need for the system scenario analysis carried out by the risk side to provide further meaning to the health of the system. The objective of health monitoring is to enable change or modification to be able to improve the system. In

this case, they both provide necessary but incomplete functions.

Some metrics have multiple aspects to them. For example, data redundancy can be expressed as the use of multiple system sensors to monitoring the systems. However, it can also be used to refer to the data that is collected over the same time value. It is important to clarify the metric value definitions based on the realities of the the CES being monitored whenever possible.

3.7.3 Appraising Performance Metrics

An important feature of metrics is that they are measurable. Depending on the metric, however, how that measurement or value looks like can vary dramatically. Limited access to data traditionally leads PRA model metrics to be more qualitative [21], whereas PHM relies on quantitative metrics to compare the performance of different health-monitoring algorithms. Table 3.4 indicates which of the identified metrics can be expressed either qualitatively or quantitatively. As there are some metrics that can be expressed equally well with either quantitative or qualitative values, it is up to the model designers to determine what form they would like the metric to take. As there are multiple methods for evaluating these performance metrics, this chapter does not provide specific examples; however, the citations listed in Table 3.4 provide instances of relevant work in which this metrics have been considered and evaluated for a particular problem space.

Table 3.4: Quantitative and qualitative features for evaluating system-level performance metrics

Metric	**Expressed Quantitatively**	**Example Qualitative Ranges**	**Select Literature on Evaluating Metric**	
			Quantitative	Qualitative
Age of system failure information	✓	Outdated–Up-to-date	[75]	[75]
Completeness of system failure information	X	None–Complete	N/A	[76]
Cost-effectiveness of information gathering	✓	Inefficient–Economical	[2]	[2]
Coverage of system failure information	X	None–Complete	N/A	[76]
Ease of model notification for new failure modes	X	Rigid–Fully adaptable	N/A	[77]
Framework completeness	X	Binary assessment–Multiple outputs	N/A	[76]
Framework coverage of known failure modes	✓	Single failure–Comprehensive	[2]	[75]
Framework explainability	X	Black box–Fully explainable	N/A	[78]
Framework traceability	X	Black box–Fully explainable	N/A	[78]
Preliminary model construction costs	✓	Prohibitive–None/Minimal	[75]	[2]
Age of operational data	✓	Old–New	[75]	[75]
Availability of data collection sources	✓	Unavailable–Available	[75]	[2]
Data collection costs per sampling rate	✓	Expensive–None	[75]	[2]
Data collection equipment requirements	✓	Significant burden–None	[79]	[79]
Data coverage	X	Limited–Full	[2]	[75]
Information content per sampling rate	✓	Low–High	[80]	[69]
Number of inferred data sources	✓	None–Many	[79]	[79]
Number of operational data sources	✓	None–Many	[79]	[79]
Operational data redundancy	✓	None/Unique–Duplicative	[81]	[79]
Required data storage capacity per sampling rate	✓	Prohibitive–None/Minimal	[79]	[79]
Sampling rate frequency	✓	Very low –Very high	[80]	[79]
Pre-processing equipment requirements	✓	Significant burden–None	[79]	[69]
Pre-processing explainability	✓	Black box–Fully explainable	[78]	[78]
Pre-processing time per sampling	✓	Too slow–Instantaneous	[79]	[71]
Pre-processing traceability	X	Not traceable–Fully traceable	N/A	[82]
Assessment accuracy	✓	Not Accurate–Highly accurate	[24]	[79]
Assessment algorithm sensitivity	✓	Highly sensitive–Not sensitive	[83]	[79]
Assessment interpretability	✓	Not meaningful–Meaningful	[84]	[85]
Assessment precision	✓	Not precise–Very precise	[24]	[79]
Assessment forecast window	✓	Immediate–Long-term	[24]	[86]
Response time to (sub)system-level anomalies	✓	Instantaneous–Too slow	[24]	[71]
System state requirements for assessments	✓	Significant burden–None	[87]	[79]
Time required for assessment	✓	Slow–Fast	[87]	[71]
Expertise required for model use	✓	None–Specific training	[2]	[2]
Model output interpretability	X	Not meaningful–Meaningful	N/A	[84]

3.7.4 Metric Comparisons

The performance metrics described in this chapter are intended to be applicable for use across health monitoring model types and structures. As such, these metrics are not only beneficial for evaluating a model's current performance but also for comparing differences in health monitoring model designs. In developing a model, the designer must consider a range of design and structural choices before selecting the optimal alternative for use. Performance metrics provide a systematic and clear approach for identifying better design choices for specific health monitoring objectives.

There are two kinds of metric comparisons that should be considered when ranking model designs. The first case occurs when the design change impacts a limited number of model aspects. In that instance, the metrics that differ across models can be compared to one another. One approach for this would be to use one model as a baseline and to compare the "robustness" of the model; that is, how did the design change cause the metric in question to change relative to the baseline. This is a variation of the PHM metric "Sampling Rate Robustness" proposed by Saxena et al. [24], which was defined as the estimated effect on metric M from a change in the data set sampling frequency for a reference data set of length L. From a baseline reference frequency, they argued that the impact of that particular design choice, SRS, could be defined as Eq. 3.1:

$$SRS(\omega_{reference}, \omega) = \frac{1}{L}\sum_{l=1}^{L}\frac{min(M(l, \omega_{reference}, M(l, \omega))}{max(M(l, \omega_{reference}, M(l, \omega))} \tag{3.1}$$

Depending on the metric in question, the proposed design may be more or less preferable if the calculated SRS is greater than or less than 1, the value associated with no change to the model's performance. A generalized version of this equation, Eq. 3.2, provides a clear method for comparing the same metric's robustness across different model design alternatives D :

$$R(D_{reference}, D) = \frac{1}{L}\sum_{l=1}^{L}\frac{min(M(l, D_{reference}, M(l, D))}{max(M(l, D_{reference}, M(l, D)} \tag{3.2}$$

Comparisons across performance metrics to make design decisions, however, require a greater understanding of the system operator's priorities for system-level health management. If the objective is to minimize health management costs, designs that result in decreased cost-related performance metric (i.e., utilizing older data, reducing amount of data collection) values will be prioritized; likewise, if health accuracy is more critical, then system-level assessment performance metrics will be more important to consider. The functional breakout of metrics described in Section 3.7.2.1 provides a good approach for identifying relevant metrics to prioritize for a given model function.

3.8 Future Improvements to Performance Metrics Set for SIPPRA Models

Future research for identifying performance metrics for system-level health monitoring models falls under expanding of the list of performance metrics, further defining how to quantify the metrics, and the identification of specific groups of metrics that can be used to answer design questions. Work on any of these areas will help to provide model designers with improved insight for designing optimal models for a specified performance level and operational requirements.

This work plants a stake in the ground for the different performance metrics available for evaluating CES health models. Although the list in Table 3.1 provides a comprehensive picture of system-level health monitoring model performance, the list was derived using the SIPPRA framework outlined by Moradi and Groth [9]. A more detailed study using different approaches for SIPPRA can look into the different metrics and functional groups presented and determine whether other metrics should be considered when selecting criteria for comparing model designs.

Another way to improve this work is to consider other ways for quantifying these characteristics. One strength about the list of identified performance metrics is that each metric has a wide range of possible methods for evaluating them; this makes them applicable to a number of different model scenarios and system types. However, some metrics, such as those associated with model interpretability and explainability, are currently better described in qualitative terms rather than quan-

titatively. Defining a new metric that is calculable may prove to supply further rigor and objectivity to those metrics. One potential way to do this is through the use of a belief triple that provides different values about an individuals belief, disbelief, and uncertainty of a given design question.

A third step forward for future research would be to identify which groups or metrics are useful for answering specific model design and structure questions. Although this work has verified the list of proposed metrics, and the structured example presented in this chapter illustrates how these metrics could be used to make a design decision, the decision-making process has yet to be validated. That would require the implementation of these metrics to make and justify a model design decision. To do that though, certain metrics may be more useful for making those comparisons than other metrics; future work to consider performing includes further identifying those metric groups and classifications through a more detailed case study featuring specific model designs or a particular CES.

3.9 Chapter Conclusion

The availability of different performance metrics for models built for SIPPRA health management outlined in this chapter allow for risk and health models to be selectively designed and developed to monitor complex engineering systems. Health monitoring models can be time and resource-intensive to construct and maintain; identifying what is important for building effective and efficient models to monitor and approximate system prognostics and diagnostics is critical in maximizing their

utility.

Ultimately, as shown in the structured example in Section 3.6, the type of performance metrics used depends on the nature of the system and the operational requirements of the model. Systems with a greater reliance on human features may opt for faster response times for quicker human intervention, while hardware and software-heavy systems might find reducing required data storage capacity more desirable. Using different performance metrics identified and verified in this chapter, as well as scenario-specific quantification strategies, will lead to different design choices based on the balance between functional need and operational constraints.

Chapter 4: Techniques for Discretizing Operational Data Streams in Continuous-Time Accident Sequences

CES health models need to account for the large volume of available system data produced from multiple sources. The flow of data, however, can range from monthly or quarterly inspections to a near-constant stream of sensor information. This is particularly relevant for SIPPRA models that incorporate failure mode and risk analyses with live system data. DBNs simplify this requirement by segmenting data streams into discrete time periods to minimize data collection and processing requirements. However, there are multiple ways to discretize a continuous-time event sequence, and, unlike the discretization of continuous parameters and variable states, there is limited research on the definition and formulation of time steps in discrete-time models [59]. This chapter presents a range of techniques that can be used for discretizing continuous-time event sequences into discrete-time slices for use as input information for DBNs. The results of this work were presented in conference papers for the 29th European Safety and Reliability Conference (ESREL 2019)[15] and the 2021 Reliability and Maintainability Symposium (RAMS 2021) [16].

4.1 Operational Timeline for CESes

CES health evaluations are built upon the assessed health of system components, which in turn is derived from the measurements of system parameters. As such, CES health is not a single measurement, but the product of a complex interaction between multiple data streams. In the wake of an accident or operational changes, the shifts found in CES data streams reflect failures or degradation beyond baseline operational statuses that ultimately affect the system's overall health.

The generic operational timeline presented in Figure 4.1 depicts a simplified relationship of CES health with various types of CES data prior to and following an accident event. The start of the timeline (t_0) consists of baseline operational information about the current system's health and operator activities, O_0. As readings from sensors or other health monitoring activities come in, they provide information about the system's parameters. These readings can occur at different rates, thereby providing new information with which to update the system's health status. Over the course of a CES's lifetime, events can occur that fundamentally change certain aspects of the system. The impact of these events may also generate a delayed response from the parameter readings or operator activity. Immediate knowledge of these events or degradation states can mitigate accident consequences or change how the operator may respond to the accident scenario; however, discrete-time models like DBNs require a structured approach for for modeling time-dependencies within the system. If each parameter modeled has a distinct cycle for providing new information, then there is the potential for various approaches existing for defining

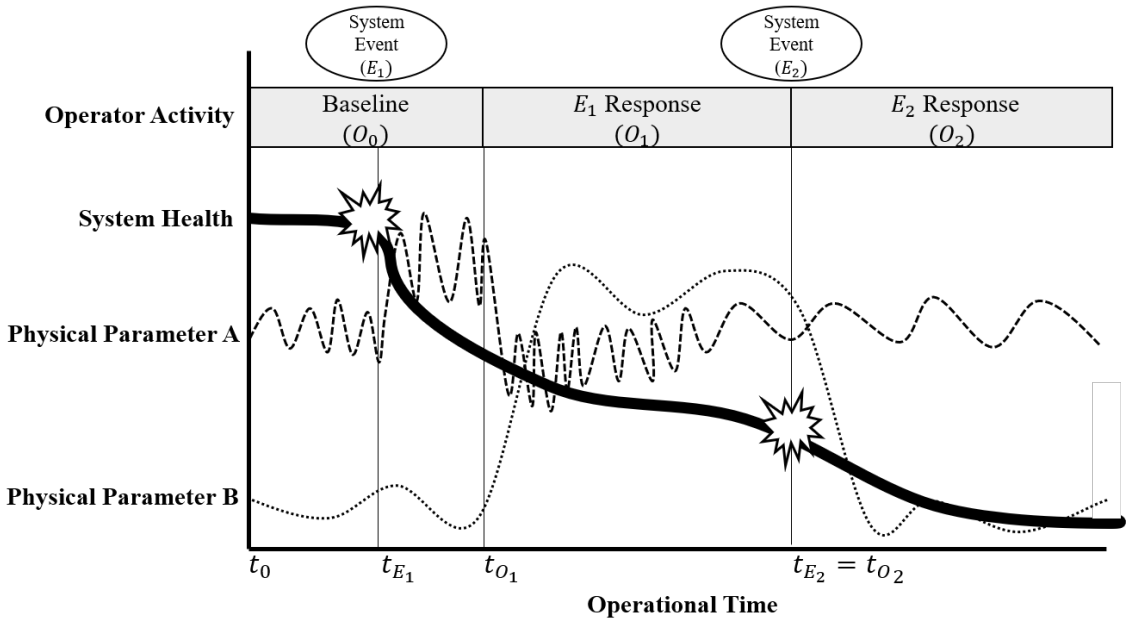


Figure 4.1: Simplified operational timeline for a generic complex system that visualizes the relationship between an event E_i, which impacts the system's physical parameters, and the system operator's activity state O_i.

discrete-time slices for a continuous-time sequence.

4.2 Time Segmentation: Analogous Research in Data Mining

The primary objective of CES data stream discretization is to model the flow of system data in such a way that the progression of system health due to changes in system parameters is effectively captured. With this information, a system's health could be evaluated based on collecting operational data. This problem has analogs in computer science data mining research. One common problem studied is the successful representation of time series data by mining databases over the use of large time segments. By utilizing segments rather than the full database, storage, use, and transmission of the information can become more efficient. As such, efforts towards

"time segmentation" are comparable to data stream discretization as both efforts attempt to use less data points to create as complete an image as possible. Even the description of the segmentation problem itself is similar: "Given a time series T, produce the best representation using only K segments such that the maximum error for any segment does not exceed some user-specified threshold." [88]

In their review of time segmentation algorithms, Keogh et al. [88] identified three primary methods providing a clear way for segmenting time while also maintaining the shape of the data:

- Sliding Windows: Using a starting point for the time segment, this approach approximates the data to the right of the starting point with increasingly longer segments until an error threshold is met.
- Top-Down: This approach partitions time series in a recursive manner until a threshold is met.
- Bottom-up: In a similar move to the top-down approach, this method merges data from the smallest approximation until a threshold is met.

Although a similar concept, there are a few differences between the time segmentation and the data stream discretization problems that the computer science and reliability engineering world are trying to face, respectively. First, time segmentation problems focus on replicating available data segments and data sets. This is different from CES health management, where the goal is to relate new operational data with previously identified system health behavior. The data stream discretization problem can be expressed as, "Given a time series T, produce the best representation using only K *data points* such that the maximum error for any seg-

ment does not exceed some user-specified threshold." Rather than trying to replicate the existing operational data, CES health models try to determine current system health. For that reason, "discretization" is a more appropriate term to be used in this context as the method transforms a continuous stream rather than a fixed data set. To that end, the top-down and bottom-up approaches by Keogh et al. are not useful algorithms to consider as the operational data set considered is continually expanding with new system data. The sliding windows algorithm, however, may be a useful tool to apply when considering a range of discretization lengths.

4.3 Analyzing DBN Discretization Methods in Reliability Engineering Literature

To determine how the reliability engineering community has previously addressed continuous-time discretization of data streams when constructing DBNs, reliability research publications were reviewed and categorized by the discretization method used. Within the *Reliability Engineering and System Safety (RESS)* publication database on Elsevier, 135 publications from 1988-2021 were identified as related to "dynamic Bayesian Networks." Of the 135 publications, 44 used DBNs as a primary components in their work and were considered "relevant" to this analysis, 30 were "loosely aligned" with DBNs but did not specifically used DBNs in the research, and 61 were considered "not relevant" to DBNs. Those that were classified as "not relevant" identified previous DBN research as fundamental to the published work, but did not contribute any DBN-related research.

4.3.1 DBN Discretization Methods Used in Reliability Research

4.3.1.1 Time-based Discretization

The primary discretization method for constructing DBNs use operational time as the discretizing factor. The earliest depictions of DBNs describe a generalized model with an unspecified time discretization; Dean and Kanazawa's [59] formulation of their model assumed that time was linear and that time steps were discrete and separated from one another by a constant "little Δ." This is similar to another approach by Cooper et al. [89], who used the same concept to create a model that predicted disease diagnoses based on temporal evidence. A generalized approach was also used in the research carried out by Kohda and Cui [11]; in their work, a time slice K was considered to be Δt away from its preceding time slice.

The timeline in Figure 4.2 shows how the operational timeline presented in Figure 4.1 would be partitioned using a generalized time discretization approach. Over a total length of time T, n time slices could be drawn over evenly-spaced intervals of length Δt at t_i, where i is the number of the time slice after t_0, the origin of the timeline. The location of these slices are independent of any events or system parameter fluctuations.

The n slices divide the operational timeline into n+1 different regions that theoretically represent the range from $[t_i, t_i + 1)$. The values of the parameters and other system states at the beginning of the time region become the inputs used in the DBN to represent system operations until the next time slice. Using the system's

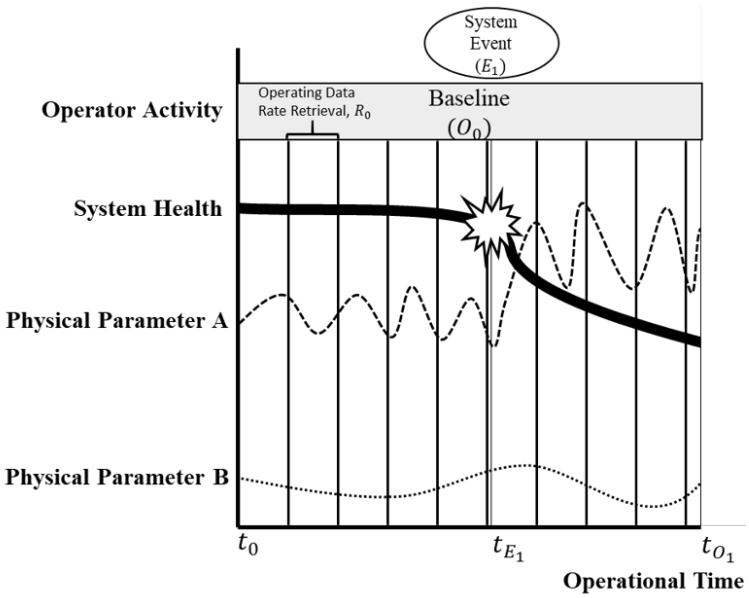


Figure 4.2: A time-based discretization overlaying the system operational timeline pictured in Figure 4.1. The time slices marked at t_i are separated by a distance Δt which reflects a regular data rate retrieval. This technique is independent of system-specific events or parameter changes.

status at the beginning of the model timeline as the model's initializing information, each region $R_i(X_1, X_2, \ldots, X_n)$ can be determined from the following relationship based on the model's conditional probability tables that define the system:

$$R_i(X_1, X_2, \ldots, X_n) = f(R_{(i-1)}(X_1, X_2, \ldots, X_n)) \tag{4.1}$$

where $f(R(X))$ is the formulation of the different conditional probabilities of that particular region. The space between intervals, Δt, is left unspecified and reflects the computational origin of this type of model. Often, it is indicative of the described model's granularity; i.e., for a TTBN, the fundamental time difference between the anterior and posterior time slice.

As DBNs became a more frequently used method to approach reliability and

PHM-related problems, it began to be appropriate to define the spacing interval used in the model. There is not a significant difference between constructing a generalized or periodic-based model; the difference is whether the Δt is known. In either case, the interval is repeated across the timeline.

Choosing a specific periodic time step for a DBN varies on the availability of computational resources and uncertainty in the system. As previously mentioned, generalized time steps reflect the granularity of the model; to that extent, the period length can be based on the rate of information released about a specific parameter. Typically, that time step is the lowest common rate of relevant information generation. Time steps can also be calculated to cover a pre-determined amount of time T. In that case, the space between each time slice is $\Delta t = T/k$, where k is the number of intervals to be included in the model. This matches a parameter discretization method referred to as the Equal Width Distribution, whose intervals are defined by Yang and Webb [90] as:

$$t_0 + \Delta t, t_0 + 2 * \Delta t, \ldots t_0 + (k-1) * \Delta t \tag{4.2}$$

4.3.1.2 State-based Discretization

The techniques described in Section 4.3.1.1 model a continuous operational timeline with discrete, evenly-spaced time steps. While such a method is commonly used in constructing DBNs, there are other approaches to discretizing time. These alternative methods identify significant parameter changes or threshold events

within the system and use them to assign the time slices.

State-based discretization techniques partition a system's operational timeline based on the attributes of its parameters. In a dynamic system, it is expected that nodes representing physical parameters may have a wide range of possible state values; depending on the user's needs, there is flexibility in defining when an interval cut is made. Figure 4.3 illustrates how a status-based discretization would partition the timeline; in this instance, the time slices would occur when the change in parameter A reaches a certain threshold value. In other instances, it may occur when the change in parameter values has been deemed significant. A similar approach can also be applied to discretize the operational timeline by events that impact the system. Since events have a distinct and irreversible impact on the system's state parameters, they provide useful information as markers in the operational timeline.

Using a state- or event-based discretization technique provides a discrete-time model with time slices that are closely aligned to time periods of shifting operational information and priorities; however, they are significantly more complicated to construct than their periodic counterparts. Instead of creating a timeline that has the desired number of time slices with a specified time length, the state-based discrete model are dependent on the current information provided by the system; as a result, the time slices are determined in near-real time. This type of discretization requires continuous access to system data to determine where the significant changes occur.

There are a few constraints that need to be met in order for DBNs constructed with these discretization approaches to be functional. First, the system operations

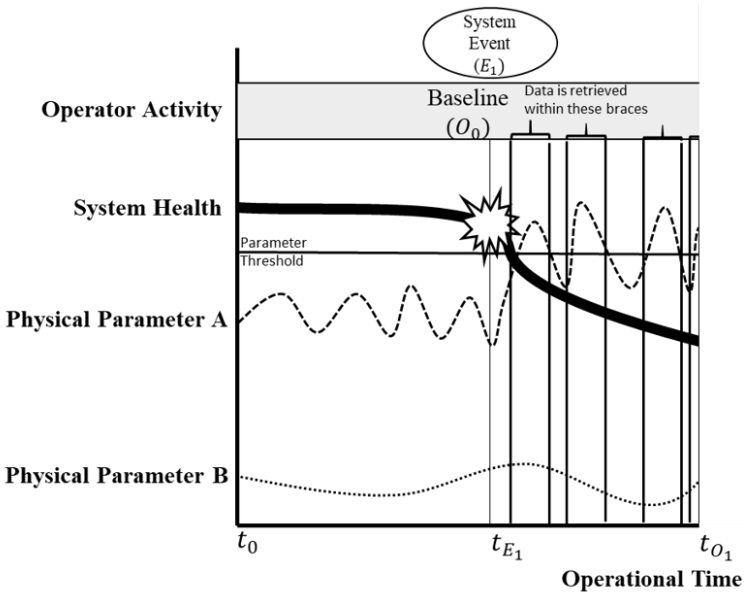


Figure 4.3: A state-based discretization method overlaying the system operational timeline. Since the parameter is affected by events that impact the system, these markers may be more useful in determining the health of the system at more turbulent moments.

cannot be repeatable; i.e.,

$$R_i(X_1, X_2, X_3, \ldots X_n) \neq R_j(X_1, X_2, X_3, \ldots X_n) \tag{4.3}$$

where R_i and R_j are different time regions. DBNs are required to be acyclic; however, there may be times when the values of the nodes may repeat themselves. It is understood that when this happens, the two configurations are not identical. More often than not, the preceding time slices differ, resulting in a different set of marginal probabilities. Additionally, the determination of the time slices must align with the requirements specified by the model. Fortunately, events and state changes would be identified during the regular pace of data generation for the system.

Although limited, there are a few papers that use non-time based methods to

discretize their model. Zhou et al. [91] created DBNs based on components that generated batches for aerospace pyromechanical device products. Bismut and Straub [92] used fatigue cycles. In each case, the time steps between the models was not consistent, but the implication is that the same loads were applied to the systems, making it possible to apply a consistent causal relationship. Some researchers, such as Zhao et al. [55] created simulated events that change the status of the studied system; the manufactured changes align with the time-based discretization used in the experiments but do not use the event itself as a means of breaking apart the operational timeline.

4.3.2 Analysis of Current DBN Discretization Practices in Reliability Research

Table 4.1 shows the distribution of the research by general topic area and discretization technique. The research literature in this table only used two methods to construct DBNs. For research in which accidents or events were planned to occur at a given time during the experiment, those models were classified as "time-based" because the event-related information was not used to create the time steps.

It is understandable why research has primarily used time-based discretization techniques for constructing DBNs. The benefit to using a standard division of operational time is that DBNs are simpler to construct when their CPTs are quantified using a pre-determined time interval. However, because they generate time steps that are independent of the status of the system, models can become redundant and

Table 4.1: Distribution of DBN discretization methods in *Reliability Engineering and System Safety* articles, 2005-2021.

Topic	Discretization Method	Number of Research Submissions
General Engineering	Time-based State-based Other	17 1 0
Petrochemical and Chemical	Time-based State-based Other	11 0 0
Manufacturing	Time-based State-based Other	2 0 0
Miscellaneous	Time-based State-based Other	12 1 0
Total	**Time-based** **State-based** **Other**	**42** **2** **0**

computationally expensive when the time steps create time periods in which the system does not change significantly. Alternatively, depending on the granularity of the model, events or state changes could be missed if the width of the times steps are significant enough.

4.3.3 Gaps in the Current Literature

A main conclusion from the analysis performed by Lewis and Groth [15] was that the discretization methods used in current reliability engineering studies face unique challenges. The independent nature of time-based discretization means that certain time intervals limit the amount of operational data considered for assessing system health. On the other hand, a state-based discretization approach depends entirely on a system's current operational status; if a parameter threshold is not reached, the model will not capture valuable system information. This prevents

significant system health forecasting in the event of pending system degradation.

Previous efforts to model CES health using DBNs have relied on consistent sensor information sampling rates across the system's operational timeline. Since the intervals were determined prior to their implementation, the models were independent of the system's operational context. As such, significant changes in a system's operating state resulting from accident events or dynamic operational environments can render prior data collection rates ineffective. Overly frequent sampling may result in redundant data collection and processing, while infrequent rates lose the operator valuable response time and insight to address any system malfunctions. It is desirable, therefore, to have a method of adjusting the model's discrete updating rate to respond to dynamic shifts in the complex engineering system's operational status.

4.4 Developing a Hybrid Time-based Discretization Model

The analysis on the discretization methods outlined in Sections 4.3.1.1 and 4.3.1.2 indicate that relying on either a solely system-independent or fully system-dependent method to discretize an operational timeline has varying challenges with respect to computational costs and redundancy of the model. A system-independent time discretization method is easy to create and initialize; however, it does not respond to the unique features of a specific system. Alternatively, a system-dependent approach identifies critical events and parameter changes of interest; however, doing so would be resource-intensive and require a significant amount of a priori data to

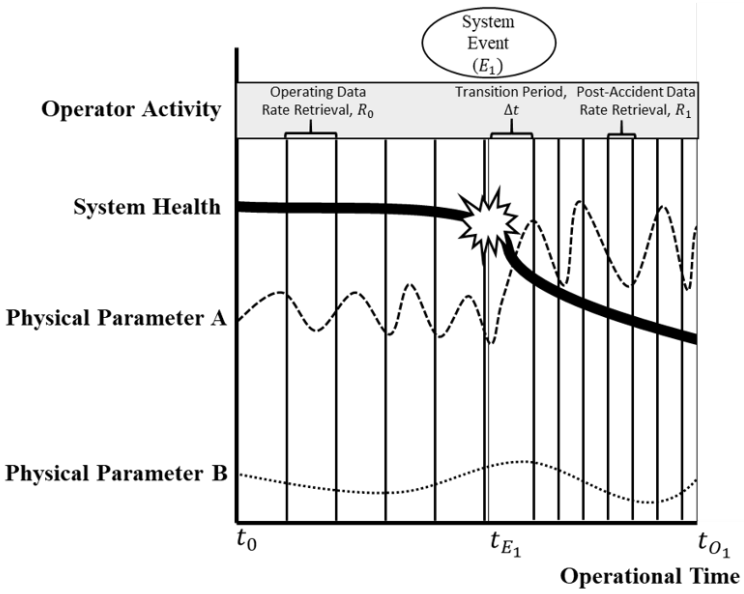


Figure 4.4: Example of proposed hybrid time-based data measurements for a system experiencing an accident. First, system events are identified; then, a periodic time step is used to monitor the changes in the system until another event is identified.

understand system state fluctuations.

Integrating these two techniques in a hybrid discretization approach, would produce a DBN that has the beneficial aspects from both discretization methods. One method of hybrid discretization would be equivalent to non-uniform spacing between events. Immediately following an event, new information provides more insight into the system's health; as such, there may be a need for a model to have more time steps. Once the system's parameters have stabilized to the event, fewer time steps are needed to understand the system's status. This approach was used by Groth et al. [10], who partitioned the operational timeline of a nuclear plant accident sequence over varying lengths based on how much time had elapsed since the accident.

Another hybrid implementation of the two discretization approaches, shown

in Figure 4.4, uses two different time scales. A primary time scale is used as a baseline for system measurement until an event occurs. Following the specified event, the model uses a secondary time scale for system measurement. The width of the time step used following an event depends on the desired granularity; smaller time intervals generate a greater amount of information to conduct prognostic and system health-related inferences, while longer time intervals provide fewer updates.

The remainder of this chapter is the structured proposal and demonstration of the approach mentioned above to sample data from a near-continuous stream of system information at an appropriate rate during an accident scenario. The result of this method will be to construct causal-based models, like DBNs, with CES operational data that better reflects the current health status of the system.

4.4.1 Changes to a CES Operational Timeline During an Accident Sequences

Assume that a CES C has an operational timeline O_T , in which there exists two time periods, as shown in Figure 4.5: a normal operational time period O_0 and an accident operational time period O_1. The transition from O_0 to O_1 occurs following accident A, but the timing of the accident is unknown. It is also assumed that following the accident, C maintains the same structure and rate of information availability.

Within CES C, there exists a set S of system data sources that could be used to measure, quantify, and monitor the system's health. Elements within this set could

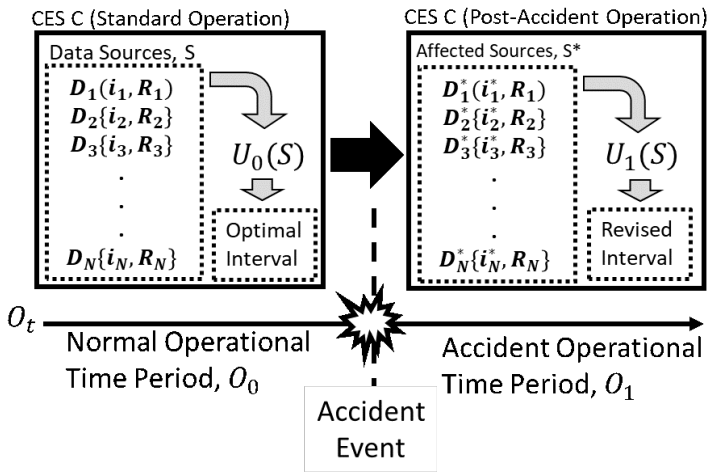


Figure 4.5: Following an accident, a CES needs a new interval for appropriate system health management

be sensors, health monitors, or even system performance evaluations. Each data source D has an information input value i with a minimum updating interval of rate R. The data sources, their values, and their information rates are inputs into a utility function U_O, that describes the model's overall value. By understanding the model utility function, one can optimize the function to maximize model performance and minimize performance costs by adjusting data collection rates. The optimal collection rate determined should then be used in a normal operational environment.

During an accident sequence, however, the value of system data shifts. The causal relationships held during the O_0 phase may no longer be valid, requiring new system relationships to be identified. In order to reach a similar level of confidence in the health assessments evaluated during the O_1 phase, an increased sampling rate is needed decreased amount of time may be needed to better understand the progression of affected system operations and system health following the accident. Following these steps results in the discretization structure in Figure 4.4.

4.4.2 Hybrid Time-based Discretization Framework

The framework for the hybrid time-based approach for data stream discretization, represented visually in Figure 4.6, contains five steps for developing a responsive data collection strategy that provides relevant system information to make system health assessments.

In Step I, available system data sources and their rates for generating new information are identified. Causal relationships that occur within the system are identified in Step II. These two steps help construct a DBN causal model structure that represents the system as well as prepares for Step III: determining the data updating rate interval under normal operating conditions. The measurement interval is conditional on the available data sources, the quality and age of information, and the resources available to process the data into meaningful prognostics work. It is assumed that the system is initially in a normal operational phase and predominantly stays under those conditions. This is reasonable given the relatively low frequency of accident occurrences for a standard system.

Steps IV and V reflect that CESes operate in dynamic operational environments. The fourth step of this procedure considers what accidents may occur to the system, and the parameter thresholds that would indicate whether a system has transitioned into abnormal operating conditions. Certain information may be more indicative of system health after an accidents. The last step is to identify data sampling rates that can better identify system health following specific accidents. Depending on the accident, different values may be identified and considered.

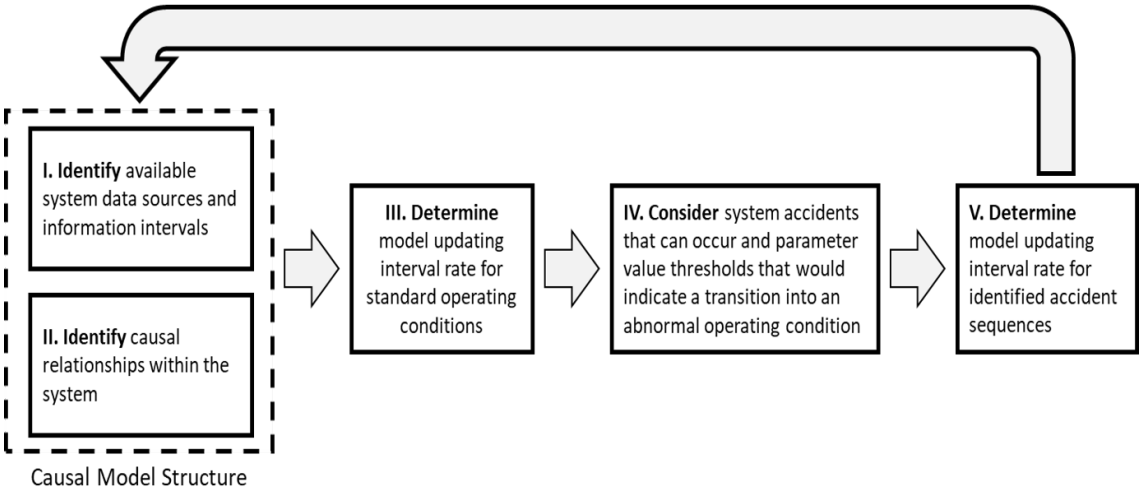


Figure 4.6: Proposed procedure for determining CES operational data interval rates in dynamic environments

The result of this method is the formation of a DBN model that emphasizes a better understanding of the system health for a degraded CES. Data collected at these specified intervals will be input into the DBN as new evidence and will propagate new system and component health estimates.

4.5 Applying the Hybrid Time-based Approach to a Simplified Accident Scenario

A simple toy problem is used to demonstrate how the hybrid time-based sampling method proposed in Section 4.4.2 is applied to a CES operational timeline. The structure of the problem is not based on an actual system but serves as an appropriately complex example for understanding the differences in using this technique over a general time-based discretization.

Demonstration of Step I

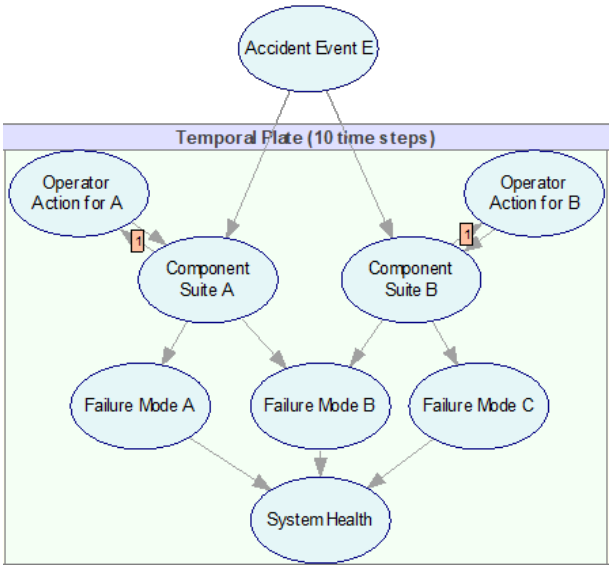


Figure 4.7: DBN for simplified toy problem

In this example scenario, there is a CES C with two suites of components, A and B, that is at risk of an accident event E. Each component suite has distinct failure modes and share a common failure mode as well. Operators to this system are monitoring readings from the suites and make system adjustments accordingly. Component Suite A provide suite info every second, while B provides new info every five seconds. The variability of system information data suggests multiple data processing rates are viable for constructing a DBN for SIPPRA health management.

Demonstration of Step II

The DBN structured in Figure 4.7 is based off the information outlined from Step I and represents the causal relations between the health of CES C and its system parameters and operator actions.

Demonstration of Step III

In this study, the normal operating time step would be five seconds, the rate of data generation for Component Suite B. This was chosen because new data would

be available for each relevant system parameters at each time step. This is not the fastest possible rate; Suite A generates new data every second. However, the longer interval reduces unnecessary data collection and storing and model requirements for a system operating under standard conditions.

Demonstration of Step IV

As mentioned earlier, Event E may occur and provide degradation if the components are not functioning as intended.

Demonstration of Step V

For the defined accident sequence, Event E, it is important to identify a data interval that is as small as possible since unaddressed system degradation can rapidly lead to further system and health risks. The smallest possible measurement rate is one second, the rate of the generated data from the model. This is a rate that is five times faster than the standard operating data measurement rate and would provide more system information faster for the new operational state. Therefore, a DBN constructed to model the system health of this reactor would begin with a time step of five seconds; following event E, the time step would shorten to one second.

4.6 Analysis of the Hybrid Time-based Discretization Demonstration

4.6.1 Demonstration Results

Table 4.2 illustrates how the proposed hybrid-time sampling method improves upon the data collection for a single-rate method. The table compares the amount of time steps, i.e., data points, collected from the system for three different data

collection strategies: two single-rate approaches with information intervals every one or five seconds, or the hybrid method described in Section 4 (five second intervals before accident; one second intervals afterwards).

The one second interval has the fastest data rate and records the largest amount of system data. If a TOP accident occurred 150 seconds after beginning system measurements, 150 time steps would have been recorded. This would provide a significant amount of system data for improved system understanding; however, the system is assumed to be under normal operating conditions, so much of that information is unnecessary and requires a strategy for excessive data storage and management. After the accident, however, it is more important to have access to system data pertaining to the new operational environment. Twenty minutes following the accident, the one second interval approach would have collected 1,200 new points of system information; it only takes 100 seconds to acquire 100 new points. This rapid data sampling is useful in new operational environments, but less so in the baseline state.

On the other hand, a five second interval provides much less system information over the same time period. While this is better under standard conditions (only 30 data pulls occurred, compared to 150 over the one second interval method), this means that less information is also available after the accident. Where the faster sampling method collected 1,200 measurements in the 20 minutes following the accident, this slower rate only had 240 opportunities for updating the model with new information. Less operational data means a reduced ability to effectively model system health in a changing environment.

Table 4.2: Illustrative comparison of time-based method with hybrid time-based method in simple example

Data Sampling Plan	Single Interval Approach		Proposed Approach
Normal operational rate	1	5	5
Time steps before accident	150	30	30
Time steps 20 minutes following accident	1200	240	1200
Time required for 100 data points (s)	100	500	100

The single discretization approach prevents system prognostics from monitoring system health effectively for both before and after accident events; the proposed multi-rate approach addresses these challenges. The new hybrid time method utilizes longer data intervals for standard conditions. This makes sense as that environment space is more stable; a CES operating under specified conditions should behave in a known way. On the other hand, new operational states require more information for better understanding system health trajectories. As such, the new discretization approach emphasizes more data propagation and shorter data intervals during accident sequences.

4.6.2 Implications of Hybrid Time-based Discretization on CES Health Management

In previous health management efforts, there has been a focus on systems that face consistent degradation patterns in a singular operational environment. These assumptions allow for the use of known rates and models to make accurate health predictions. CESes operate in dynamic environments where they may face a sudden degradation event that greatly weakens the system. The approach proposed in this chapter allows for a more nuanced look at accidents, throwing out multiple

Table 4.3: Qualitative comparison of data stream discretization techniques (H - High, M - Medium, L - Low, N - None).

	Discretization Technique		
	Time-based	State-based	Multi-Interval
Time Intervals	Evenly-spaced	Model-dependent	Evenly-spaced Sets
Timeline Computational Requirements	L	H	M
Timeline Complexity	L	H	M
Timeline System Information Requirements	N	H	M
Potential for Redundant Calculations	H	N	M

assumptions about the operational state of a CES.

Table 4.3 compares the hybrid time-based discretization approach against a time- and state-based one. Discrete-time models rely on a brief window of information to make their inference calculations. If there is an instance in which everything is relatively stable, then a DBN constructed using the hybrid time-based discretization can have more regular and longer time intervals; when there is a lot of variability in the system parameters as a result of external events, then the hybrid time-based discretization technique can provide the time steps and additional information needed to make critical inferences right after an event.

The hybrid time-based method recognizes that CESes operate in different environments; acknowledging this elevates health monitoring to address multiple operational states. Previous research efforts have addressed failures from a single operating environment. By responding to changes in operational states, maintenance plans are adjusted based on relevant system data, allowing for effective and efficient CES health management.

4.7 Chapter Conclusion

An analysis of DBN-based reliability engineering research found that although there are multiple data stream discretization methods defined, reliability engineers have predominantly followed a time-based approach when constructing DBN time steps: defining the time step length, defining the number of time steps and the period of model run-time, or leaving the time step undefined. It is unclear, however whether these approaches are the most effective and efficient for CESes that produce a wide volume of data at varying rates. Other approaches leverage system-specific knowledge based on system parameters or distinct events that change the time frame of system monitoring, data collection, or operator activity. Relying solely on a state-based techniques can limit one's understanding of a system as events can be random and result in large increments of underutilized system data.

This chapter proposed a framework for developing a hybrid time-based data stream discretization method that incorporates major status and event changes with a periodic discretization. The results from the simplified CES example indicate that its use would provide more flexibility in gathering relevant information. At the same time, it would still obtain consistent system information while potentially reducing computational time. This approach allows models to provide more flexibility to address critical system variations while maintaining a simple method for defining time steps.

Chapter 5: Development of a Case Study for Comparing the Performance of DBN-based SIPPRA Health Management Models

Chapters 3 and 4 presented SIPPRA model performance metrics and data stream discretization techniques as CES-agnostic; however, having a working case study of an average CES helps to better understand the impact that such discretizations have on the performance of DBNs for SIPPRA health management. For this case study, a DBN model was structured based on a real-world operational scenario. By adjusting the data stream discretization strategy used to parameterize the health monitoring model, the CPTs for that DBN would also change. This chapter outlines the development and verification of the DBN node structure used for the comparison studies presented in Chapter 6. Using simulated accident sequence data from a model sodium fast nuclear reactor as a case study, a DBN structure is designed, quantified, and verified based on evidence associated with a transient overpower event. The results indicate that a joint prognostic and diagnostic model that is responsive to new system evidence can be generated from operating data to represent CES health. This underlying model structure, therefore, can serve as a valid form of comparing the impact different data stream discretizations have on the

performance of DBNs constructed for SIPPRA health management. This work was presented by Lewis and Groth in a journal article published in a special edition of *Algorithms* in March 2020 dedicated to applications of Bayesian Networks [17].

5.1 Case Study Development Methodology

Four key actions are taken to develop this case study:

1. Define the underlying scenario and context of the case study
2. Collect and analyze the data relevant to the case study
3. Build a SIPPRA DBN structure for the case study
4. Verify the constructed DBN structure captures expected scenario outcomes

This chapter will address the results of these actions in a fairly high overview; further details regarding the accident dataset and the overall DBN construction process are covered in Appendices A and B, respectively.

5.2 Case Study Scenario

The case study outlined in this chapter is a simplified version of the scenario studied by Jankovsky et al. [93] that captures the modeling and monitoring of a SFR in a transient overpower (TOP) event. SFRs can be considered a typical CES in that they feature the primary characters inherent for CES outlined in Chapter 2; namely, they are composed of human, hardware and software components and generate a large amount of operational data from a number of data sources at varying rates.

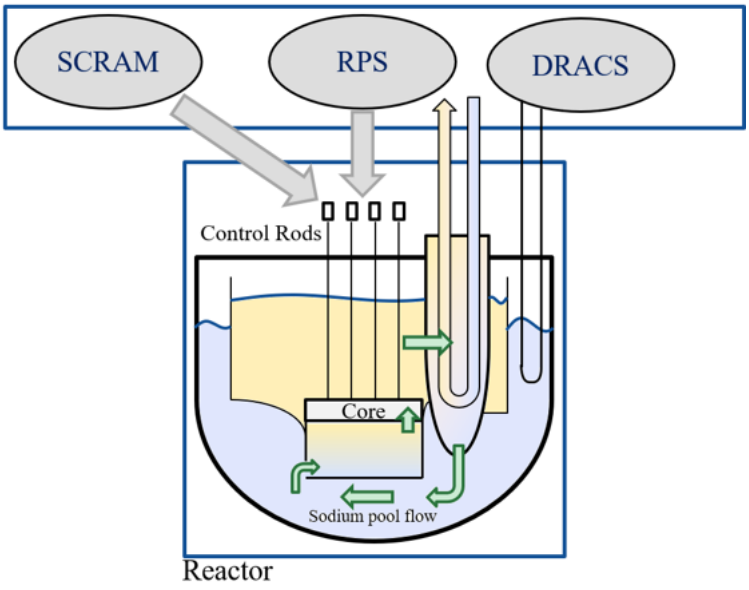


Figure 5.1: This case study models a SFR, consisting of the reactor, SCRAM, a reactor protection system (RPS), and a direct reactor auxiliary cooling system (DRACS), that experiences a transient overpower (TOP).

As SFRs rely on fast-neutron activity, the need for other equipment is minimized, making them useful models for simplifying complicated nuclear processes. As shown in Figure 5.1, in addition to a nuclear core which consists of four distinct channels, the system in the case study has a balance of plant consisting of a SCRAM and reactor protection system (RPS) and a direct auxiliary cooling system (DRACS); however, for the purposes of this case study, the focus will be on the reactor core itself. Although there are multiple components to a sodium fast reactor that provide a significant amount of system information through sensors and operational reports, the case study focuses on a limited number of data sources; namely, the main indicators of the automatic SCRAM process for shutting down the reactor.

The primary accident event described through the DBN model in this case study is a TOP event. Previously studied by Jankovsky et al. [93], such an event

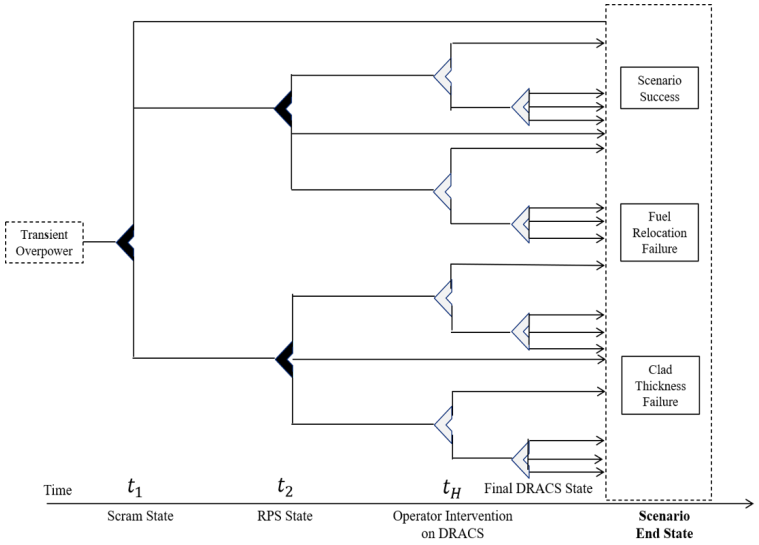


Figure 5.2: General progression of SFR TOP accident event leading to a successful scenario, fuel relocation failure, or clad thickness failure. Further discussion of the event tree is presented in Appendix A

can be caused by external factors, e.g., an earthquake, that results in a sudden surge of power generation in the reactor. When such an event occurs, the reactor's automatic SCRAM mechanism is expected to respond to operational changes by inserting control rods into the reactor to greatly reduce power generation; common indicators for the automatic SCRAM mechanism include changes to net reactivity, cold pool temperature, and other fuel feedback values [93]. Depending on the cause of the accident, however, SCRAM and RPS functions may be impacted, limiting their ability to prevent core reactions from further escalating. If this were to occur, the reactor would face a significant risk of fuel relocation and clad melting, resulting in a partial or full nuclear meltdown. A visual representation of the simplified event description is presented in Figure 5.2.

5.3 Case Study Data

The accident data used in this case study is modified from the study by Jankovsky et al. [93]. In their work, a dynamic event tree (DET) was used to construct a series of accident event scenarios that addressed potential failure points when responding to a TOP event. Based on the event scenario specifications determined through the ADAPT software, simulation models focusing on different aspects of the nuclear reactor were used to produce different parameters necessary for monitoring overall system health. The models were run to simulate data readings throughout the reactor and BOP for a full day after the TOP event (86,400 simulation seconds). The scenario was considered finished when either: the cladding fraction of the core channels reached an average of 90% (representing a clad melting failure), the temperature of the cold pool had reached a significantly high temperature resulting in a fuel relocation, or the reactor had survived the simulated day without reaching those other thresholds. In those instances, it is assumed that operators would have had enough time to address any problems with the system's processes.

To simplify the accident scenario further for the case study presented in this chapter, a TOP event has already occurred and the initial SCRAM and TRIP actions have already been performed. This results in a modified event tree compared to the one used by Jankovsky et al. [93], further explained in Appendix A.

5.3.1 Case Study Datasets

The research by Jankovsky et al. [93] produced three datasets that contained operational data different system parameters. They provided information about the reactor channels, overall reaction values, and information about the balance of plant and auxiliary systems. These datasets were created from the following two models:

- **SAS4A/SASSYS-1:** This simulation tool provides information about the nuclear reactions occurring within the reactor channels. Data provided from this model include inlet and outlet temperature, and inlet and outlet flow. SAS4A also generates nuclear core activities including power generation and reaction coefficient values. This data provides insight into the current power generated from the reactor and other information about the physical nuclear reactions taking place. SAS4A data generation rate varies, from providing new information every 0.1s at the onset of the simulation, to increasing up to every 100s further along the accident simulation run-time.

- **PRIMAR4:** This simulation tool generates values for the overall piping and thermodynamics of the reactor, the BOP and other auxiliary systems. This includes information about the temperature and pressures of compressible volumes and pools around the reactor, measurements of the pumps and the different elements of the balance of plant and cooling systems. The data collection rate for this code is every 9.09s.

An operational timeline for the SFR can be comprised from these models of the

different parameters necessary for monitoring overall system health.

5.3.2 ADAPT Tree Data

The ADAPT software provided the framework to generate the accident scenarios. Using a set of reactor coefficients, simulations would run until either a DET branching condition was met or a failure threshold had been reached. At branching conditions, ADAPT generates different instances based on the possible branch states. Those instances would then run until either the simulation stopped or a new branching condition. Each branch segment contains the operational data for that instance as well as general branch information including the branch's starting and ending times, initial branching conditions, and the probability of the branching occurring.

5.3.3 Data Pre-processing

The original study carried by Jankovsky et al. consisted of 2,052 scenarios separated into 2,920 branch segments. The following process, shown in a visual representation in Figure 5.3, was used to construct the complete operation timeline for each scenario identified through the DET and pre-process the scenario data:

1. Construct complete timelines by combining branch segments' starting and ending times

2. Separate scenarios DET outcome (Success, Clad Failure, Fuel Relocation)

3. Filter scenario set by removing scenarios from list that concluded before 9.09

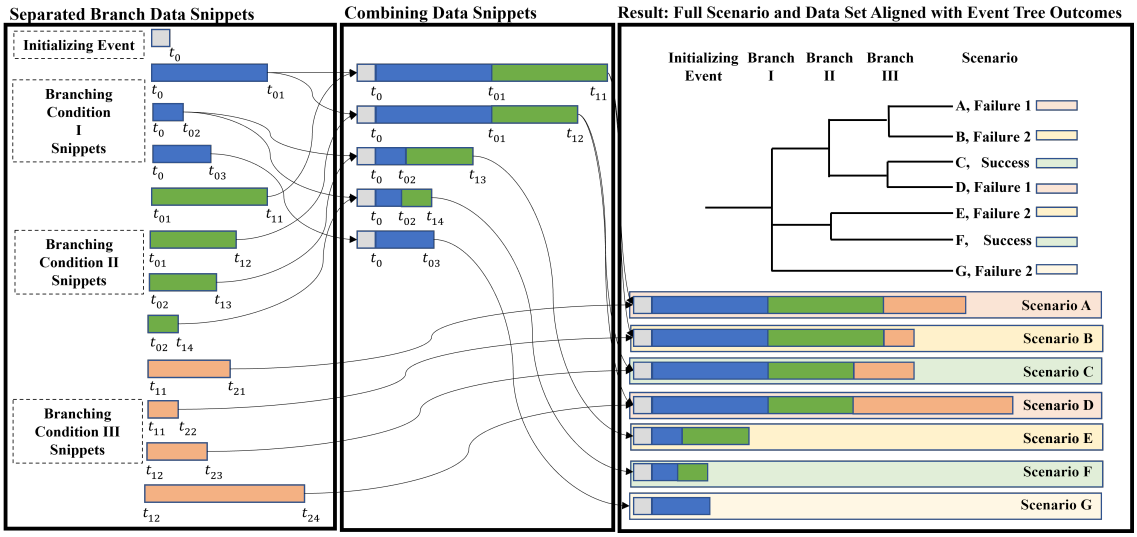


Figure 5.3: A rigorous data processing approach was used to combine the different branch data snippets into full accident sequences reflected in the DET.

seconds (the first measurement of the PRIMAR4 data)

The last step ensures that data is available for each scenario to capture the dynamic causal relationships within the system. This is a reasonable step to take as these are scenarios within which there is limited time for action, making the scenario outcome unavoidable. This left a total of 1,920 scenarios for use as data sources for parameterizing DBN CPTs.

5.4 Building a DBN for SIPPRA Health Management

As previously mentioned in Chapter 2, a primary characteristic of DBNs is that evidence concerning the state of one node can lead to an updated estimate of the value of another node within the model through logical inferences. As DBN nodes have the potential to represent a wide range of features within a system, from individual sensors to entire subsystems, the ability to perform inferences on

nodes across a DBN model makes it a powerful tool for understanding the current a system's current health condition. Information about the system can act as evidence in one section of the model and propagate to other parts of the model. This allows DBNs to provide more insight into the system than non-causal models with the same amount of evidence.

DBN models designed to monitor and provide CES health information following a major accident should incorporate operational system information as well as the conditions of potential accidents. As shown in Figure 5.4, DBN models representing this type of CES scenario can be structured using six distinct data regions. Each section of the model has its own node types, data availability, and purpose for managing CES health. These six regions are:

- **Accident State:** The nodes within this region represent the different accident events covered by the DBN that the CES might encounter. Typically, CESes operate at normal or baseline conditions until one of these events occur; after an accident, the system operates under different circumstances. Accidents covered by these models can be external to the system (i.e. an earthquake or a power outage) or internal (sabotage). Depending on the potential accidents that may impact the CES, different accident nodes are needed to reflect different states that may not be mutually exclusive and occur at the same time.

- **System Component Health:** This region of the model describes the current operational states of CES components. Depending on the particular opera-

tional scenario represented in the model, these states may be dynamically changing over the course of scenario or static. As such, this is the only information region that includes both static and dynamic nodes. The state of the dynamic component nodes are often affected by operator involvement during the process of CES health management.

- **System Information/Sensor Data:** CESes generate a sizable amount of data. This data can take the form of sensor readings, analytical measurements, and status and maintenance reports, and as either continuous or discrete measurements. Since data sources are frequently updated with new system information, these are the "dynamic" nodes of the DBN. The classification of data into discrete bins is dependent upon the nature of the data; however, a common bin distribution would be for "normal operating conditions", "above operating conditions", and "below operating conditions." This region is predominantly where additional model evidence is added to the DBN, as extra information can be used to make informed decisions about the nodes in the other regions.

- **Human Involvement:** This region contains information about the necessary actions and interventions that the operator would take on the system. CES operators receive system information and sensor data; with that knowledge, they make decisions to adjust system component performance. This creates a causal loop (since system component states affect the system data they provide) that is not possible in a static BN, but is by using DBNs. For

that reason, this region contains nodes that are temporally dependent on any operator actions previously taken place.

- **System Prognostics:** This region of the model provides insight into potential failure modes that a CES might fail from given a particular accident. These are typically distinct from other prognostics techniques which might indicate a remaining useful life of the system; rather than indicating whether a system will be healthy or faulty at a given point in time, these nodes indicate what will be the resulting failure of the system given the current system information and data. Examples may include "metal cladding failure" or "short-circuit" and are often expressed as a binary option (i.e., "Yes/True" and "No/False"). Each failure event state should be considered as a separate node.

- **System Health Diagnostics:** Based on the system prognostics estimate captured in the "System Prognostics" region, CES health can be assessed by whether or not the system will fail from another failure mode other than expected failures at the end of its life cycle. Unlike the other nodes, this region can be fully captured in a single node with a number of mutually exclusive states; depending on the CES's structure, this approach can also be used on the subsystem level as well. An easy way of expressing this is through a simple OR gate-style node for overall system health. Example of states may include "Healthy," "Faulty," or "Inoperable."

A DBN constructed using the framework in Figure 5.4 uses information about the accident state as well as system or sensor data to provide information about the

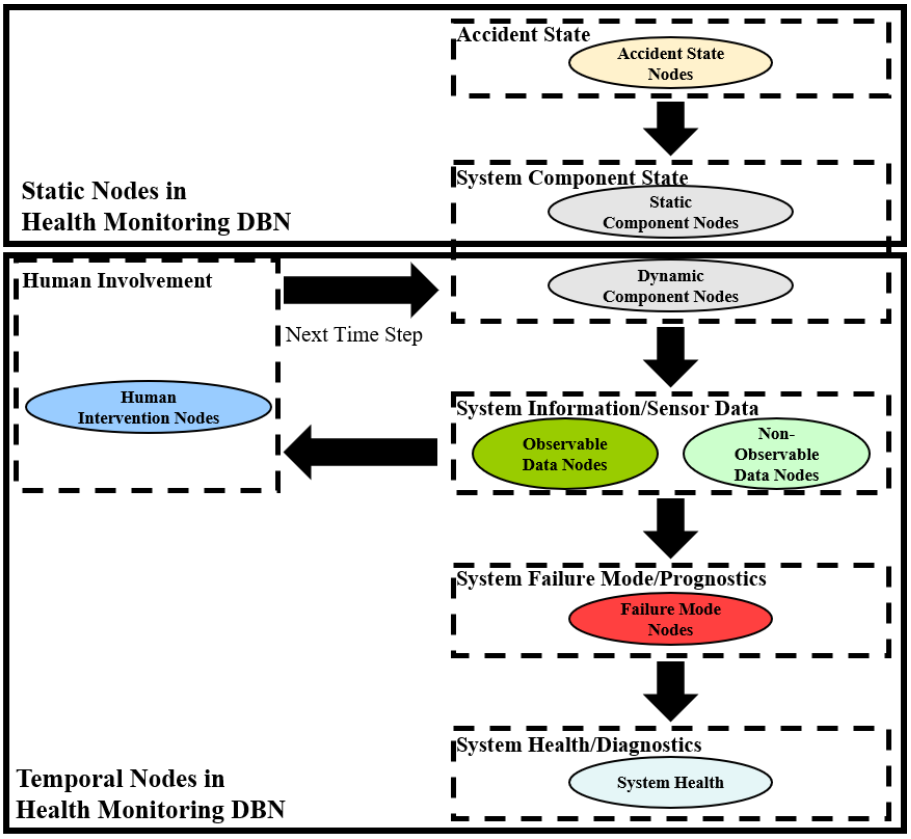


Figure 5.4: DBN models for monitoring CES health following an accident event can classify their nodes into six information regions. Arrows drawn between the information regions reflect the directed relationships across information regions.

current system diagnostics. Understanding the current system health in conjunction with the system measurements can be used for system prognostics to identify potential causes of system failure. Because of the relationship arcs connecting the six different regions within a DBN, a model structured in this manner can be used to provide both diagnostics and prognostics. From the information provided by the sensor and knowledge about the accident sequences, insights can be gleaned about the current health of the system. Through the time-dependent relationships within the system, information can propagate backwards through the model. This adjusts the current understanding of system health, particularly concerning human

involvement and intervention into the system.

In addition to diagnostic capabilities, this framework introduces a predictive end state for system health. The benefit of including this in a temporal network is that the probability of certain prognostic updates can fluctuate, resulting in dynamic prediction of system failures. As such, Information provided about the current system can then be used to calculate the future outcomes that the system might face.

Although this proposed framework is intended to provide diagnostic and prognostic capabilities for CES health management, it can also serve as a structure for an operator decision support tool. Supplying current evidence about the system into the model would provide insight into future system health; varying potential operator actions would result in different model outcomes. An operator would be able to see those potential outcomes and make a more informed decision about which operational action to take. In that instance, additional network nodes representing the operator's decisions would be placed within the "Human Involvement" information region.

This structure of system data and model evidence into these distinct information regions is scalable to address different accidents, data types, and prognostics failure modes. It is also compressible; a purely prognostics-focused model can have the accident or failure state nodes act as root nodes, while a solely diagnostics model would have a singular failure mode in the system prognostics information region: "System Failure."

5.4.1 Case Study Network Structure

Using the case study data described in Section 5.3, a DBN model was constructed to cover the primary elements of the SFR relevant to TOP-induced SCRAM failures. Previous work by Groth et al. [10] found that the following parameters provided meaningful information for evaluating reactor health during a TOP event: net reactivity, coolant reactivity feedback, radial expansion reactivity feedback, doppler reactivity feedback, and cold pool temperature. Designing the DBN model shown in Figure 5.5 to help operators identify current system health status and potential failure modes following a TOP required nodes from the component state, system and sensor information, human involvement, system diagnostics, and system prognostics information regions. The temporal loops included in the model add temporal causality to constrain outcomes to follow logical relationships (e.g., clad thickness only deteriorates, the operator will not become undecided once he or she has made a decision to intervene on the DRACS, and the state of the DRACS will not revert back to nominal once it has been either enhanced or degraded). This is distinct from the other nodes which have static conditional probabilities (i.e., a prediction of the current SCRAM state is not dependent upon the SCRAM state prediction from a previous measurement).

5.4.2 Case Study Conditional Probability Tables

This model's CPTs are trained with operational data provided from scenarios that resulted in three distinct outcomes: failure due to clad melting, failure due

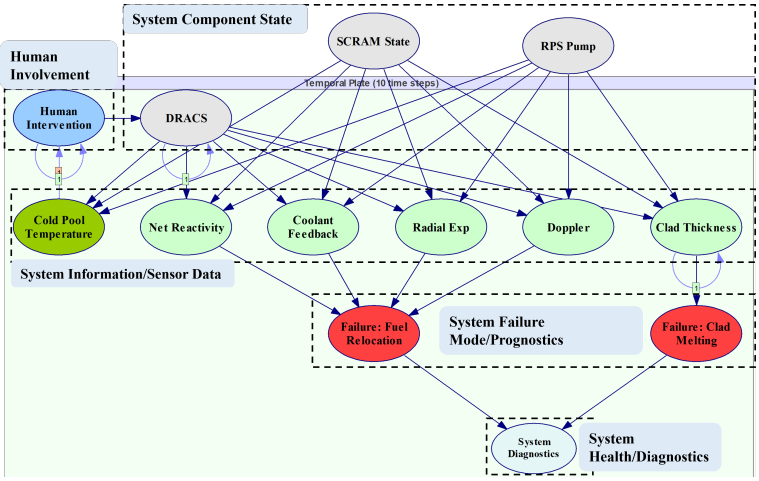


Figure 5.5: DBN node structure and relationship graph for the SFR TOP case study. Dashed boxes represent the different node regions for a diagnostics and prognostics model for CES. Node arcs capture causal relationships within the same time step, with the exception of the dynamic arcs labeled with a boxed "1." Those indicate a relationship with the previous time step. Dark green represents observable parameters, while light-green nodes are un-observable or inferred parameters.

to thermal relocation, and a successful model outcome. The model's objective is identifying the current health state of the reactor as well as the likelihood of a certain outcome based on current data from the system's sensors. Data received from the system will be used as evidence for an improved determination of the state of the reactor's SCRAM and trip mechanism. The DBN model is constructed using the GeNIE software [94]; CPT elements are calculated using the Python programming language [95].

As CESes generate a multitude of data, there are large amounts of readily available data that can be used to inform the model's quantification of the conditional probability tables. The information provided for this model was carried out in multiple simulations over different time period measurements. As previously mentioned, the nuclear reactor data from the SAS4A-SASSYS-1 code is collected

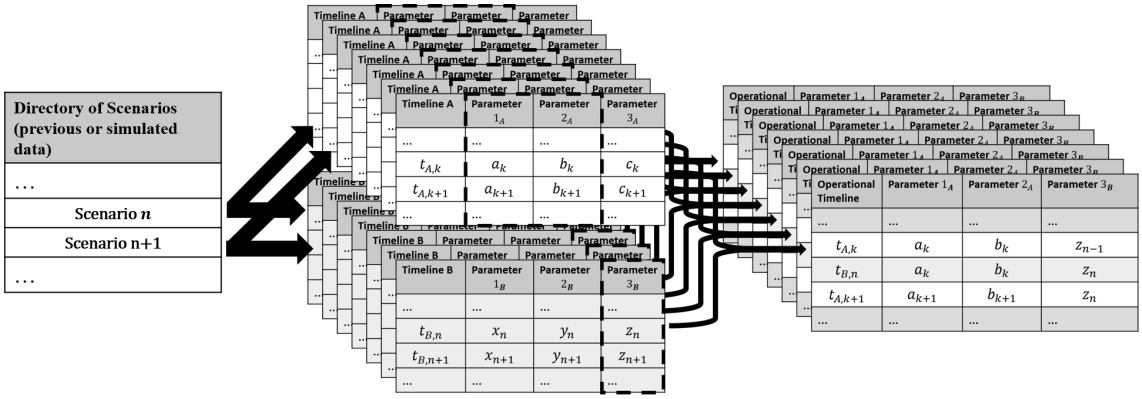


Figure 5.6: Data derived from the simulations are generated at varying time frequencies and are compiled into a single operational timeline. Given multiple accident scenarios, there are many possible operational timelines to parameterize DBN CPTs.

more rapidly at the beginning of the accident simulation, at a rate of 0.1 simulation seconds, and slows down to a collection frequency of 100 seconds, while new information from the PRIMAR4 code is provided approximately every 9 seconds. This is similar to real-world scenarios in which measurements and sensor readings occur over different frequencies. As such, operators are dealing with information with different levels of currency. In order to capture as much relevant information as possible, an operational timeline was created to consolidate data generated from the two simulation codes into one sequence of events. As illustrated in Figure 5.6, relevant information was identified from both data sets. The available data was then sorted based on the simulation time at which the data was received. When new data was acquired from a sensor, that entry would replace the measurement from the previous timing; however, the "current" information from other system sensors would remain as new data had not yet been provided.

A sample set of CPTs reflecting the progression of reactor health was quantified using the simulated data generated from SAS4A and PRIMAR4. The CPTs

Table 5.1: Model parameters and discretized bin threshold values

Model Parameter	**Low Threshold**	**High Threshold**
Net Reactivity (%)	-0.778	0.02
Coolant Reactivity Feedback (%)	-0.007	0.06
Radial Expansion Reactivity Feedback (%)	-0.077	-0.02
Doppler Feedback Reactivity (%)	-0.18	-0.04
Cold Pool Temperature (K)	753	

provide insight into the transformation of different nodes across the model over the simulation time and describe the causal relationships within the nodes. For this DBN structure, there are three types of CPT that are reflective of different structures in the nodes: static CPTs for the static nodes, and initialization and temporal CPTs for the dynamic nodes. For these CPTs, the elements in the table can be determined by a frequentist approach by counting the number of instances a child node state occurred with the identified parent node states, or

$$P(ChildState|ParentState) = \frac{\Sigma ChildState}{\Sigma ParentState}. \tag{5.1}$$

DBN nodes are designed to contain discrete states; for this model, the simulation data provided was separated into ranges based on reasonable expectations for "Low," "Medium," or "High" values. This case study used on expert judgment based on observed parameter values to determine the boundaries of the middle bin. The ranges for the case study nodes are shown in Table 5.1.

Dynamic nodes within the DBN require a starting distribution to initiate temporal relationships. This study assumed that at the beginning of the event, the initial core clad thickness was at "100%," the operator is "undecided" about intervening on the DRACS, and the operational state of the DRACS is "nominal."

Based on the previous model and data received, the CPTs for the previously described model were quantified using the 1,920 scenarios that reflected a TOP event occurring. Where there was evidence, a frequentist approach of determining probabilities was used; however, when data was not available, appropriate approximations were used to complete the table that minimized influencing the posterior estimates to a greater extent than the available information. Table 5.2 is a portion of one of the quantified CPTs based on a time step of 9 seconds. In each instance, most data was classified in the same bin as the previous measurement; any deviation would therefore be considered a rare event and worth noting. Although certain relationships might not occur in an actual accident scenario, those relationships are still expressed in the CPTs.

5.5 Case Study DBN Structure Verification

To show that the proposed DBN is effective at assessing an SFR's health following an accident event and potential future failure outcomes, hypothetical cold pool temperature data is input into the model that may be indicative of a SCRAM failure following a TOP. This data serves as evidence that will impact the posterior estimates for the system prognostics, diagnostics, and accident state. For this initial evaluation, designed as a verification of the DBN node structure only, a time-based

Table 5.2: Portion of the "Radial" node CPT. Columns with round estimates are instances of expert-based judgement.

SCRAM/Trip	SCRAM Failure,Trip Success						SCRAM, Trip Failure					
RPS Pump	Operational			Not Operational			Operational			Not Operational		
DRACS	Enh.	Nom	Deg.	Enh.	Nom	Deg.	Enh.	Nom	Deg.	Enh.	Nom	Deg.
Low	0	0	0	0	0	0	0.0537	0.0508	0.0530	0.0537	0.0508	0.0530
Medium	1.0	0.7891	1.0	1	0.0245	1	0.7946	0.7507	0.7825	0.7946	0.7507	0.7825
High	0	0.2109	0	0	0.9755	0	0.1517	0.1985	0.1644	0.1517	0.1985	0.1644

Table 5.3: Prior and posterior probabilities of SCRAM, prognostics, and system diagnostics with evidence of cold pool temperature below 753K

Model Parameter	**Prior**	**Evidence**	**Posterior**
SCRAM,Trip Success	$1 - \Sigma P(fail.) \approx 1$	Cold Pool	$1 - \Sigma P(fail.) \approx 1$
SCRAM Success, Trip Failure	$1.4 * 10^{-}9$	Temperature (1)	$1.40 * 10^{-}9$
SCRAM Failure, Trip Success	$1.4 * 10^{-}9$	=Below 753K	$1.36 * 10^{-}9$
SCRAM,Trip Failure	$2.9 * 10^{-}7$		$2.54 * 10^{-}7$

discretization with a time-step of 1200 seconds (20 minutes) is used.

Table 5.3 shows the prior and posterior SCRAM state probabilities based on evidence that the cold pool temperature was found to be below the threshold value. As expected, the prior probabilities of the SCRAM state and the cold pool temperature CPTs along with the limited amount of information provided little change to the prior; this model is still predominantly assessing that the SCRAM process is working as intended. Although minute, the posterior estimates for the state of the SCRAM and trip mechanisms are changing based on the new information. However, that information alone is not enough to convince the model that the SCRAM mechanism failed. There are many reasons that the cold pool temperature may be below the threshold value before human intervention is required; given the significantly low probability that the SCRAM and trip mechanism fails, the DBN is estimating that there is something else that could explain the discrepancy. This is also seen in the system prognostics at this particular point in time, as shown in Table 5.4. With this information, the failure outcome is very likely due to fuel relocation rather than clad failure.

This assessment of the reactor's prognostics changes, however, when new information is received. Assume now that the following sensor readings indicate that

the cold pool temperature of the reactor is now higher than the 753 threshold. This combination of evidence, significantly alters the estimate of whether the SCRAM mechanism worked, as seen in Table 5.5. The posterior estimates indicate that it is now far more likely that the SCRAM and trip failed. The model responds to a small amount of information to raise a concern that an accident has indeed occurred.

The addition of new data also changed the current prognostics outlook of the system, as seen in Table 5.6. The previous prognostics seen in Table 5.4 showed a negligible reactor failure from clad melting, and a nonexistent risk from fuel relocation; however, that assessment was based on the assumption that the SCRAM and trip mechanism were successful. Since the new evidence introduced into the model changed the posterior estimate of the SCRAM state to have failed in some manner, there is a greater likelihood that the reactor will fail by one of those failure modes. The updated prognostics now suggest that given the current data received from the system sensors, there is a 6.19% chance that the system, if conditions remain the same, would result in a failure by fuel relocation. In addition to changing the assessment of the reactor's prognostics, the influx of new system data and sensor information should impact the estimate of the reactor's health. Table 5.7 provides point estimates on the reactor system's diagnostics. At the beginning of the experiment (Time 0), there is no indication that the system would be faulty as the initial

Table 5.4: Prognostics outcome for reactor with evidence of cold pool temperature below 753K

Prognostic Outcome	**Failure: Fuel Relocation**	**Failure: Clad Melting**
Will Occur	≈ 1	$4.48E0 - 9$
Will Not Occur	$2.77E - 07$	≈ 1

Table 5.5: Prior and posterior probabilities of SCRAM diagnostics based on cold pool temperature measurement below 753K, followed by temperature measurement above 753K, and then another reading below 753K

Model Parameter	**Prior**	**Evidence**	**Posterior**
SCRAM, Trip Success	$1 - \Sigma P(failure) \approx 1$		0
SCRAM Success, Trip Failure	$1.4 * 10^{-}9$	Cold Pool Temperature (1) = Below 753K	0
SCRAM Failure, Trip Success	$1.4 * 10^{-}9$	Cold Pool Temperature (2) = Above 753K	0.0009
SCRAM, Trip Failure	$2.9 * 10^{-}7$	Cold Pool Temperature (3) = Below 753K	0.9991

Table 5.6: Prognostic outcome for reactor with cold pool temperature measurement below 753K, followed by temperature measurement above 753K, and then another reading below 753K

Prognostic Outcome	**Failure: Fuel Relocation**	**Failure: Clad Melting**
Will Occur	0.0510	0.0619
Will Not Occur	0.949	0.9381

state distribution is consistent with that of the operating baseline; as a result, it is deemed a fully healthy system facing a severe overpower event. When the high temperature reading in at Time 1, there is now a possibility that the SCRAM mechanism has failed; as a result, the reactor's health is marginally diminished. When the additional temperature reading is received at Time 2, and it becomes evident to the model that a failure in the SCRAM mechanism has occurred, the system's health diagnostic assessment is further degraded. The collection of this result, as well as the prognostic assessments and estimate in SCRAM failure would result in a more educated process to find and address the issue, minimizing any potentially harmful outcomes.

Table 5.7: Progression of system health following example accident sequence

System Diagnostic	Time 3 (Initial)	Time 23	Time 43	Time 63
Healthy	0.9490	0.9482	0.9389	0.9111
Not Healthy	0.05104	0.0518	0.0611	0.0889

5.6 Discussion of Case Study Verification Results

The results from the verification process indicate that the proposed DBN structure provides a system-level diagnostic and prognostic capability for the reactor accident sequences it was designed to monitor. Using available information from multiple number of system sensors, a clearer image of current and future system health was estimated for a complex system. The strength of the model lies in its inference abilities, as it provides a responsive posterior probability for both specific system outcomes and current health and accident states. This type of modeling is important to consider when monitoring CES health as it provides a visually appealing method of presenting the causal relationships found in these systems and subsystems. CESes are heavily integrated platforms that would otherwise not have their time-dependent causal relationships as explicitly captured with other models.

One of the common challenges associated with applying DBNs to real systems is the CPT quantification process. Depending on the number of state bins and the amount of parents for each node, the size of the tables can vary greatly, increasing the time and power required to process the probabilities. For this case study, most failure scenarios led to the same parent/child node relationships and some parent/child combinations were not met. Limiting the number of accident scenarios, minimizing the amount of states per node, and relying on expert-based relationships may reduce

the computational requirements; however, an increase in the number of scenarios and state bins make the model applicable to a wider range of accidents and failure modes and increases the granularity of the model, respectively. Further analysis is needed to identify the proper amount of granularity and model coverage for each specific CES.

This case study relied on simulated nuclear system and thermodynamics data to model the different event sequences because of limited availability of real operational data to academic institutions. However, this model structure can be a platform for evaluating on-line observational data. Historical data can be used to parameterize the DBN's CPTs; then, that information can provide opportunities for new insight into improving the future diagnostic and prognostic health management of the system.

The current structure of the DBN model is designed for a continuously operating CES that can experience an accident at any given moment. Given the long operational lifetime of these systems relative to start up and wind down time periods, this is a reasonable assumption; however, accidents can just as easily occur at the onset of operation or operation build-up. To consider these time periods when constructing a model, data or expert-based opinions of the system relationships are needed. Such a model may end up entirely distinct from one of a similar CES in its operational phase.

5.7 Chapter Conclusion

This chapter described the development and verification of a DBN structure defined for a joint diagnostic and prognostic model for monitoring complex engineering system health. By breaking apart the model nodes into the six distinct information regions, access to sensor data and system information allow for different assessments for accident scenarios, prognostics, and diagnostics for systems and subsystems. Through the SFR TOP case study, expert-based judgment and data-driven techniques were used to quantify the DBN's CPTs and strengthen the model. The model responded to the hypothetical accident data supplied as evidence by indicating an increased chance of SCRAM and trip mechanism failure and overall system failure, and a decrease in overall system health. Such an ability suggests that this model can be used to prepare CES operators for rare-event accident scenarios. Given its potential as a health monitoring model, the DBN structure can be applied as a case study for comparisons across different model designs.

Chapter 6: Comparison of DBN SIPPRA Health Models Parameterized via Different Data Stream Discretization Methods

This chapter integrates results from the previous three chapters by implementing performance metrics to better understand how different data stream discretization strategies affect the performance of health monitoring models. Using the case study scenario outlined in Chapter 5 as a specific example for a complex engineering system, this study conducted a structured comparison of model alternatives. The results indicate that different model design choices not only affect the health value outputs, but also lead to significant variations in usability. Understanding these differences will lead to different design selections under different operational conditions and restrictions.

6.1 Model Design and Discretization Methods Compared

The work in the chapter compares the performance of different DBNs designed to model the accident scenario described in Chapter 5; that is, an SFR experiencing a TOP and subsequent system decay. A total of fifty-six different DBN models are constructed using the different discretization strategies outlined in Chapter 4. This includes time-based, state-based, and hybrid time-based discretization meth-

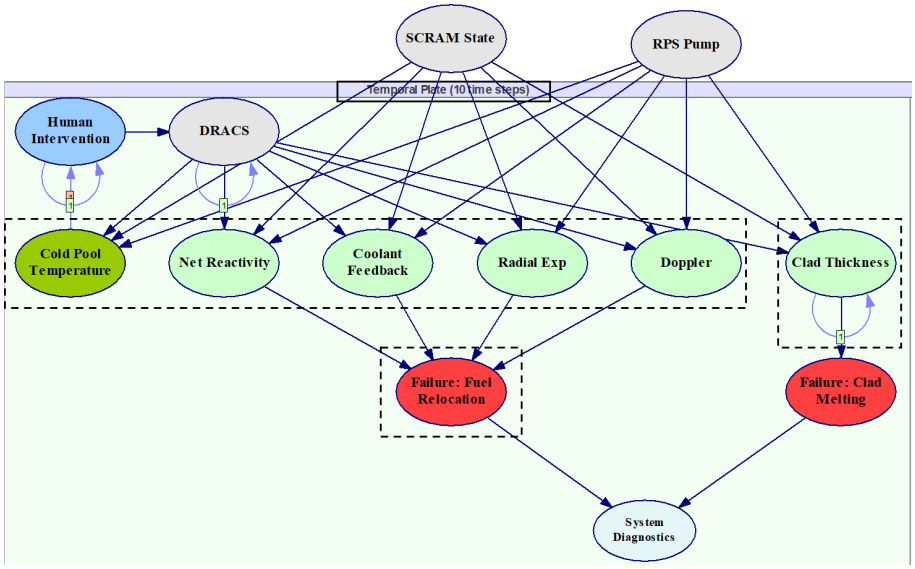


Figure 6.1: The DBNs compared in this case study use the same network structure.

Table 6.1: Summary description of discretization values used in model comparison

Discretization	Discretization Description					Number of Cases
Time-based	Data collected every	9s	60s	120s	1200s	4
State-based	Data collected when reactivity greater than	-$0.1	$0	$0.02	$0.2	4
Hybrid Time-based	Data collected every X sec until reactivity threshold; then, every Y sec					48

ods. These models all have the same node structure shown in Figure 6.1; however, each discretization method generates different CPTs that describe the underlying conditional probabilities of the system, as separate sets of data are considered when constructing the tables. This produces distinct models to consider as viable alternatives for monitoring system health. This section further describes how each discretization process is applied in this study; Table 6.1 provides a summary description of the discretization approach used in the models compared in this study.

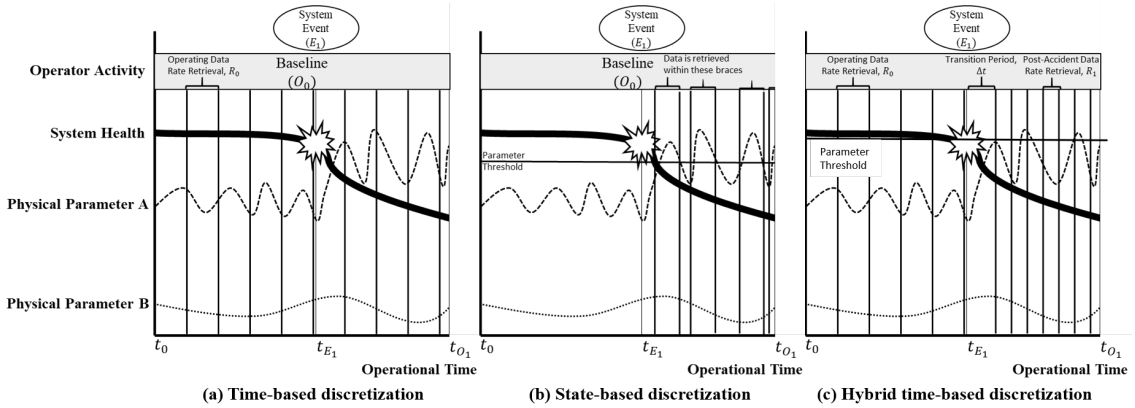


Figure 6.2: The CPTs in the DBN compared in this study are generated from data derived by a) time-based, b) state-based, and c) hybrid time-based data stream discretizations.

6.1.1 Constructing DBNs with Time-based Discretization

DBNs constructed with a time-based discretization approach are built on data collected over a specified period of time, as shown in Figure 6.2a. Four different data collection frequencies are evaluated in this comparison: 9, 60, 120, and 1,200 seconds. As this case study covers a period of 86,400 seconds, these rates translate to DBN models with 9,500, 1,440, 720, and 72 time-steps, respectively. These values were selected to provide a range of feasible monitoring time periods, with the 9 second rate equivalent to the rate in which the PRIMAR4 simulation code generates temperature data. These models were constructed using the process outlined in Chapter 5 and Appendix B.

6.1.2 Constructing DBNs with State-based Discretization

DBNs constructed with a state-based discretization approach are structured on data pertaining to a certain operational state; this is shown in Figure 6.2b.

For this case study, the reactor's net reactivity value was used as the trigger for data collection. Data is collected only when the net reactivity is evaluated over a specified threshold in a given accident scenario. Net reactivity was selected as the triggering variable because that parameter indicates whether a nuclear reaction is moving towards additional power increases.

Four net reactivity values were chosen to compare as thresholds for collecting system data: -$0.1, $0, $0.02, and $0.2. These values relate to the binning used to discretize the associated net reactivity node ($0.02), capture baseline operations ($0), or provide extreme bounding scopes (-$0.1,$0.2). To build the CPTs for these models, data is evaluated over the smallest available interval for each accident scenario. If the value of the net reactivity is evaluated as greater than the specified threshold at a given measurement, then all of the system data associated with that time is included in constructing the relevant CPTs.

6.1.3 Constructing DBNs with Hybrid Time-based Discretization

Similar to those built with a time-based discretization, the CPTs for DBNs developed using a hybrid time discretization approach are built from data collected over a specified interval; however, once a threshold state is reached on a triggering variable, data is then collected at a different rate (as explained in Chapter 4.4.2). This type of model is built as a hybrid of the previous two models, shown in Figure 6.2c.

For this study, different combinations of time-based discretization values are

paired with a net reactivity threshold as the limit to switch from one data collection rate to another. This results in a total of forty-eight distinct models (combinations where the two rates are the same are not compared as they are equivalent to the single time-based discretization described above). Two different situations were considered when defining the threshold state: when the initial time steps are larger than the subsequent ones, and when the initial time steps are smaller than the next steps. The first describes an instance of increasing the data uptake from the system; for those models, the second time steps begin when net reactivity is greater than the specified threshold. The second situation relaxes data uptake. There, the second time steps start when net reactivity is less than the specified threshold.

6.2 Performance Metrics Used to Compare Model Designs

For this study, relevant performance metrics were selected and then framed based on the specifications of the case study. The metrics used to compare the different model designs were selected from the list generated in Chapter 3. After reducing the list to consider metrics relevant for inspection, the following metrics were identified as providing different ranges of performance:

- Assessment Accuracy (Alignment of risk assessment)
- Preliminary Model Construction Costs (CPT development time)
- Information Content per Sampling Rate (Average information content)

As this work studies how different discretization methods impact model per-

formance, these metrics are model characteristics that are affected by changes to data quantity. The remainder of this section outlines how each metric is measured in this study.

6.2.1 Assessment Accuracy: Alignment of Risk Assessment

The first metric used to compare the different discretization approaches is assessment accuracy; in this study, that means how well the model's prior estimate of system health matches the underlying system safety of the accident scenario. This is a common approach to evaluating model performance; if a monitoring model is unable to provide an appropriately reflective health assessment, it is limited in its ability to be used as a health management tool.

For this case study, "Assessment Accuracy" is related to the output values of the "System Health" node. This alignment estimate is determined by calculating the joint prior probability for the "System Health Diagnostics" node derived from the model's CPTs. The prior measurements for the last model step (86,400s or equivalent) are then compared in magnitude and by percent error to the DET assessment, calculated by the summation of failure probabilities, for the health of the system. The closer the assessment is to the baseline estimate ($2.77 * 10^-7$), the more aligned the model is to the DET assessment. In terms of percent error, those values should be as close to zero as possible.

6.2.2 Model Construction Costs: CPT Development Time

The next metric studied is the model construction cost; in particular, the time required to develop the CPTs for the DBN models. Understanding the length of time required to develop a model prior to use is important when considering appropriate model designs to pursue.

This metric is evaluated as the summation of time taken to construct network CPTs that vary in response to the different discretization methods. These CPTs describe the causal relationships for the four unobservable parameters (net reactivity, coolant feedback, radial expansion, and doppler), observable parameter (cold pool temperature), dynamic clad thickness, and fuel relocation failure. For this metric, models that take a shorter amount of time to construct are preferable to those that take longer to develop.

6.2.3 Information Content per Sampling Rate: Average Conditional Entropy

The last metric compared in the study is the average information content of each model. At the beginning of an accident scenario, there are many unknowns beyond the probabilities of occurrence that are assigned to the potential accident timelines. As new system information becomes available from different data sources over time, there is greater certainty about the nature of the current accident sequence as well as its outcome. This new knowledge can ultimately lead to better preparation

and risk management for expected outcomes.

For this case study, "Information Content" is related to the values measured in the "Cold Pool Temperature" node and the associated values for the two failure mode nodes ("Failure Mode: Fuel Relocation" and "Failure Mode: Clad Thickness"). Information content for each measurement from the "Cold Pool Temperature" node is quantified using information theory principles. Equation 6.1 shows how the information content for a collection of scenario outcomes X based on the previous knowledge about Y data measurements can be expressed as the sum of the conditional entropies of potential operational sequences that would generate those same measurements:

$$H(X|Y) = -\sum_{y \epsilon Y} Pr(y)(\sum_{x \epsilon X} Pr(x|y) log(\frac{1}{Pr(x|y)}))) \tag{6.1}$$

The total information entropy is then averaged to better approximate the information content for a given set of cold pool temperature measurements. As entropy describes the amount of overall uncertainty or information required to identify a current scenario from all possible events, lower values for this metric are preferable (e.g., a value of 0 indicates complete certainty of the outcome) to larger values.

6.3 Comparison Results

This section presents the results from evaluating the performance metrics described above for DBN models built using different data stream discretization strategies. For a cleaner discussion and analysis, this section will feature either sample

Table 6.2: Sample DBN model prior safety estimates (vs. DET baseline safety estimate of $2.77 * 10^{-}7$)

	Time-based		State-based		Hybrid Time-based	
	120s	1200s	Net Reac. >= 0	Net Reac. >= 0.02	1200→120 @ Net Reac. <= 0.02	120→1200 @ Net Reac. >= 0.02
Prior Risk	2.59E-07	2.68E-07	5.16E-08	8.00E-08	8.65E-08	2.47E-07
% Difference	-6.36%	-3.21%	-81.4%	-71.1%	-68.8%	-10.73%

values or summarizing figures; the associated metrics for each model compared can be found in Appendix D. The summary figure common across the performance metrics is a heat map of metrics values. Shown in Figure 6.3, these maps can be divided into the four regions for the discretization approach used: the lower-left section (hybrid time-based discretization where the first time step rate rate is less than the secondary rate), the diagonal (standard time-based discretization), the upper right part (hybrid time-based discretization where the first time step rate is greater than the secondary rate), and the separate right-side column for state-based discretization. Model designs with more preferable values appear closer to dark green, while those with less desirable values are a darker shade of red.

6.3.1 Results of Risk Assessment Alignment Study

Table 6.2 shows a sample of estimated priors from example models for the different discretization approaches and their similarity with the underlying DET's baseline estimate of $2.77 * 10^{-}7$. The values lie roughly within an order of magnitude to the baseline estimate. The models that collect more data (1200s time step vs. 120s time step, and reactivity threshold greater than 0.2 vs. greater than 0) appear to produce more conservative safety estimates with greater percent error from the baseline estimate. This trend is further expressed in Figure 6.4, which plots the

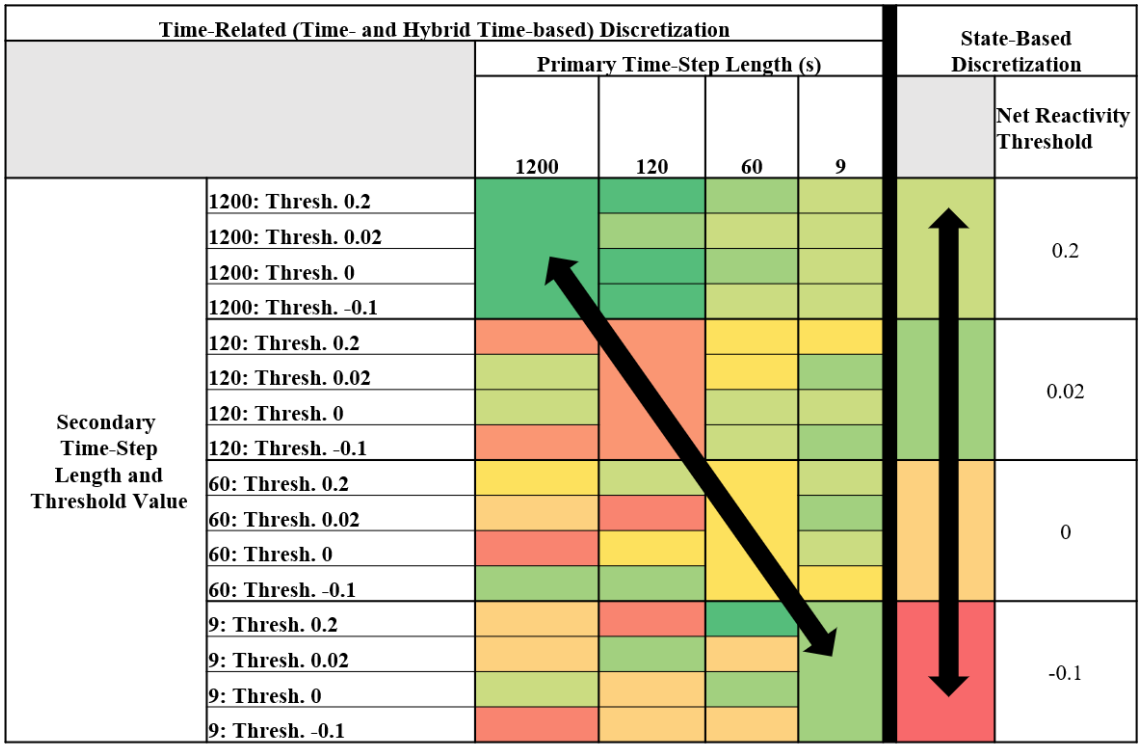


Figure 6.3: Heat maps like this one summarize the results from the performance metrics studies. Green indicates a preferable metric measurement, while red squares indicates less preferable ones. The cells along the diagonal arrow represent models built using a time-based approach, while the cells under the vertical arrow capture the results of models constructed with the state-based discretization.

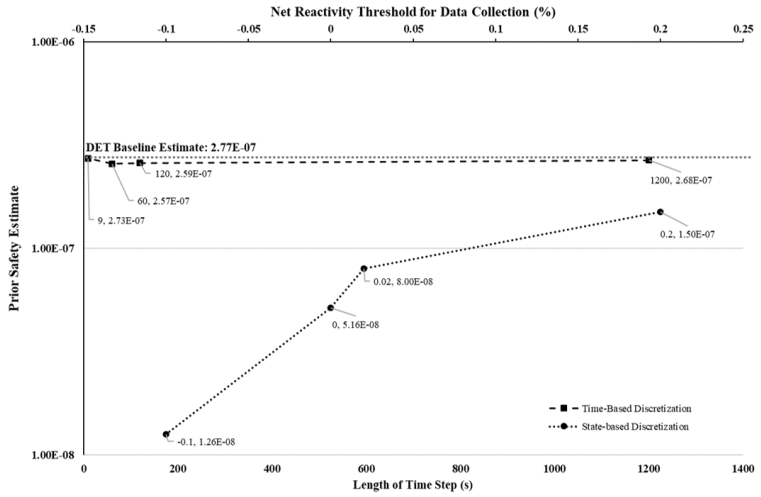


Figure 6.4: Prior safety estimates for DBN models constructed using a time- and state-based discretization approach compared to the baseline DET estimate. Time-based values (dashed line) align with the lower axis, while state-based values (dotted line) align with the upper axis.

calculated safety assessment for each state- and time-based values (the DET value is included as reference). The exception to this is the model built with 9s time steps, which has the value most similar to the baseline estimate. Even though both time- and state-based discretization strategies have a similar trajectory, the state-based discretization cover a wider range of values.

The percent errors for the hybrid discretization are compared alongside the time- and state-based discretization results in the heat map in Figure 6.7. The percent difference for the diagonal region is consistently better than the other two regions, but gets progressively larger with smaller time step lengths. The upper-right region is slightly worse than its diagonal counterparts, but improves with lower threshold states. On the other hand, the models represented in the lower-left region are significantly further off from the baseline DET estimate but worsen with lower threshold states.

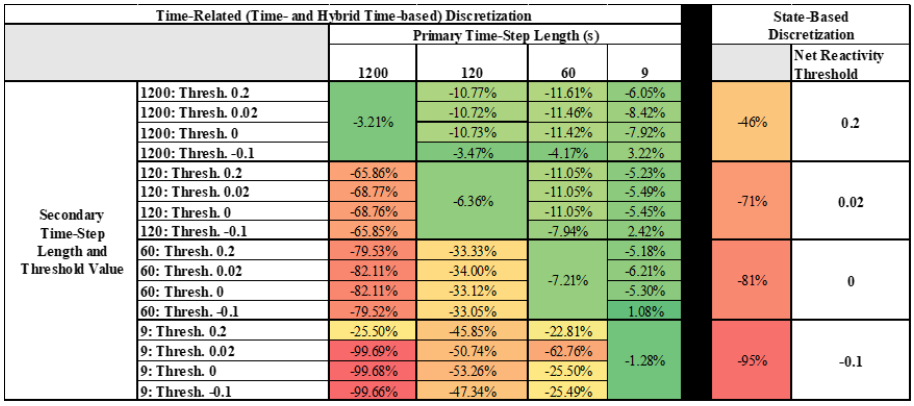


Figure 6.5: Heat map comparison of percent error of safety estimates across models and discretization strategies.

Table 6.3: Sample development time for CPTs. The remainder of the values can be found in Appendix D.

	Time-based		State-based		Hybrid Time-based	
	120s	1200s	Net Reac. >= 0	Net Reac. >= 0.02	1200→120 @ Net Reac. <= 0.02	120→1200 @ Net Reac. >= 0.02
Non-Observable Parameters	383.3	3,420.1	838.2	838.2	4,068.6	1,223.0
Observable Parameter	2,035.6	19,590.1	9,958.6	9,958.6	2,610.3	2,229.9
Fail: Fuel Relocation	58.7	28.7	15.1	15.1	16.4	17.1
Dyn. Clad Thickness	1.0	10.2	21.8	21.8	31.9	338.4
Total (s)	**2,478.6**	**23,032.9**	**10,833.7**	**10,833.7**	**6,727.2**	**3,808.4**

6.3.2 Results of CPT Development Time Study

Table 6.3 presents the amount of time it took to develop the CPTs for the example models described previously in Table 6.2. Overall, the CPTs that required the most amount of time to construct described the causal relationships for the non-observable and observable parameters. This is in large part to these variables changing over time, while the other nodes are constant over accident scenarios.

As expected, the models with CPTs constructed with more data, either by shortening the time step length or lowering the threshold value, took longer to build that those with longer time steps or higher threshold values. The construction

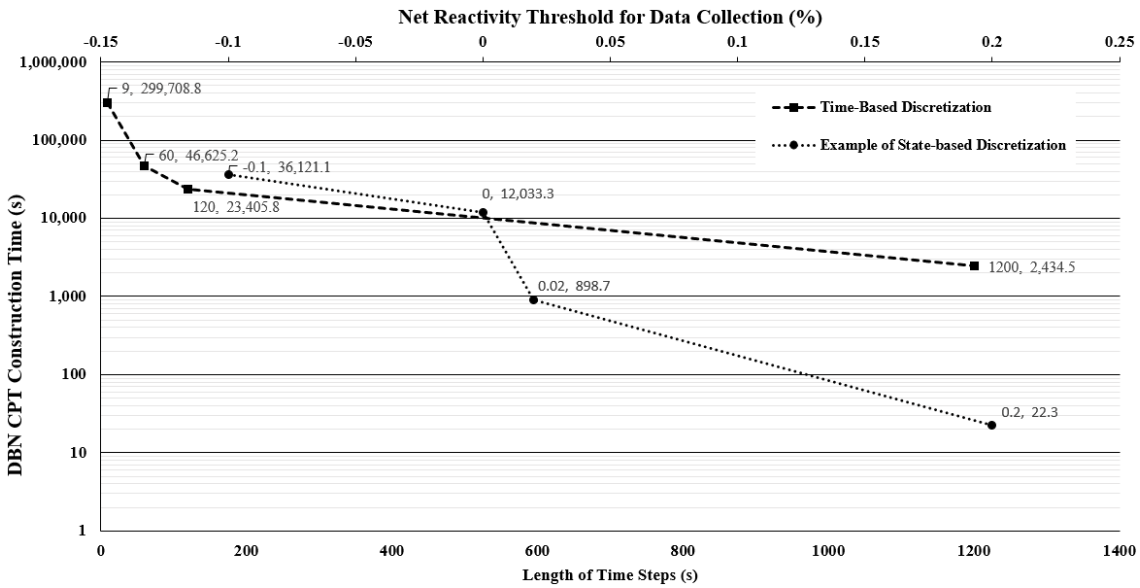


Figure 6.6: Comparison of total CPT construction time based on the length of time steps and threshold values.

times for the four time-based and state-based models are plotted in Figure 6.6 and compare the increases in computational time requirements with the increases in available data for either discretization strategy (either through shorter time-steps or lower thresholds). CPT construction times associated with the DBNs built from a time-based discretization follow a power curve. While the state-based models also require more time to develop CPTs at lower thresholds, the increase in time is not easily modeled through a curve. This can easily be seen by the sharp jump in computational time between the model measuring data at \$0.02 to \$0 threshold. The model construction time for these two discretization times appears to intersect somewhere between \$0-\$0.02 reactivity threshold and, using the power curve to determine boundaries similar to the time for the state-based discretization strategies, somewhere between 240 to 3500 s.

CPT construction times for the hybrid time-based models presented in Table

Time-Related (Time- and Hybrid Time-based) Discretization						State-Based Discretization	
		Primary Time-Step Length (s)					
		1200	120	60	9		Net Reactivity Threshold
Secondary Time-Step Length and Threshold Value	1200: Thresh. 0.2	2,478.6	3,545.1	4,791.9	17,645.1	58.30	0.2
	1200: Thresh. 0.02		3,808.4	4,831.1	17,147.0		
	1200: Thresh. 0		3,735.4	4,796.1	17,179.5		
	1200: Thresh. -0.1		18,820.6	36,783.1	229,696.3		
	120: Thresh. 0.2	2,780.0	23,032.9	26,513.9	38,295.7	813.20	0.02
	120: Thresh. 0.02	6,727.2		26,202.7	38,234.9		
	120: Thresh. 0	12,791.7		26,797.3	39,515.8		
	120: Thresh. -0.1	24,134.4		44,420.4	236,564.0		
	60: Thresh. 0.2	2,632.1	26,159.5	46,053.5	61,993.4	11,117.20	0
	60: Thresh. 0.02	10,884.9	46,278.3		62,689.6		
	60: Thresh. 0	24,947.4	48,627.0		62,689.4		
	60: Thresh. -0.1	47,889.3	49,458.4		249,578.4		
	9: Thresh. 0.2	2,725.0	26,930.4	53,125.0	299,708.8	35,319.80	-0.1
	9: Thresh. 0.02	62,493.2	295,423.2	281,304.2			
	9: Thresh. 0	155,729.9	304,475.5	312,017.7			
	9: Thresh. -0.1	312,802.7	313,412.6	318,069.3			

Figure 6.7: Heat map comparison of total CPT construction time across models and discretization strategies.

6.3 lie between the construction values for the two discretization rates when used in a time-based discretization. The remaining computational times for the hybrid strategies are captured and compared to the times from the other models in the heat map in Figure 6.7. For the most part, hybrid-time discretization construction times lie between the values of the two time-based methods used, presented along the diagonal of the heat map. That is not always the case, however; for some models, like the one built with a primary time step length of 120 seconds that transitions over into a new rate of 60 seconds following a reactivity measurement above \$0.02, the computational time required for developing the hybrid CPTs were longer than that for the time-based model built with a time-based discretization of 60s time steps (46,278.3 vs 46,053.5 seconds).

6.3.3 Results of Conditional Entropy Study

The charts in Figures 6.8 and 6.9 show the progression of average conditional entropy, or information content, for the models built using the state- and time-based

discretization strategies across the different model time steps. Each model begins with an entropy of 4.32; this is derived from the failure probabilities from the DET branches. The figures show that additional system information can affect the value of information for the particular scenario.

Generally speaking, the average conditional entropy decreases over time across all discretization methods studied, with greater decreases more likely to occur towards the earlier time steps for each model. In instances where the time-steps overlap (i.e., data would have been collected at the same time), the average conditional entropy is greater for models with more time steps. The difference between entropies at the same point in time however, appears to be reduced over smaller distances than larger ones. This is further seen, when at approximately 70,000 seconds into the simulation for the time-based discretizations, differences in the conditional entropy across the models eventually decrease, leading to roughly consistent entropy values from then on.

The heat map presented in Figure 6.10 captures the averages of each model's average information content. In general, those values were larger for models with larger time steps and more inclusive thresholds, validating the observations made before. However the values for the hybrid time-based models were either comparable to their time-based counterparts or were significantly lower than either value.

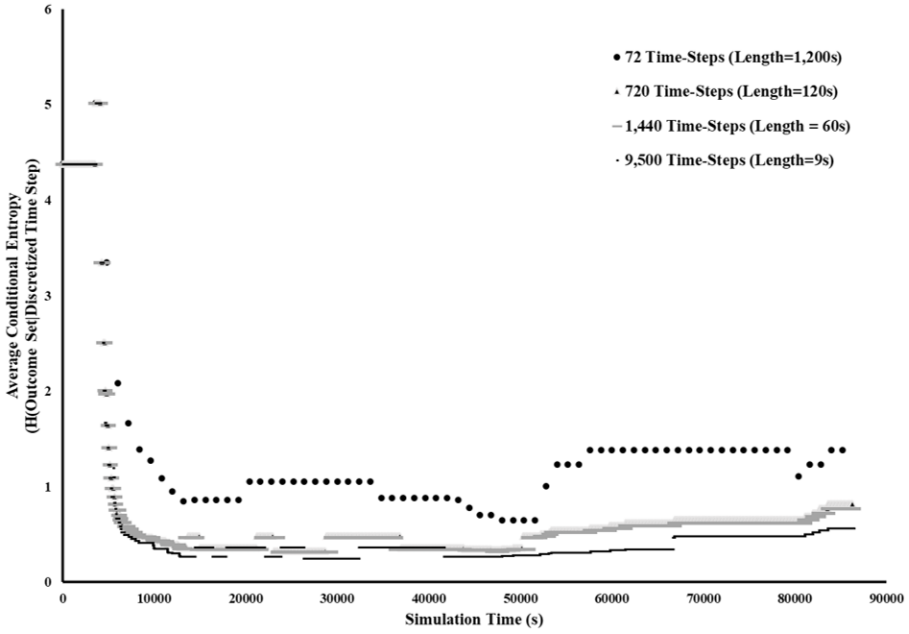


Figure 6.8: Progression of information content in the form of conditional entropy across simulated time for models built with time-based discretization.

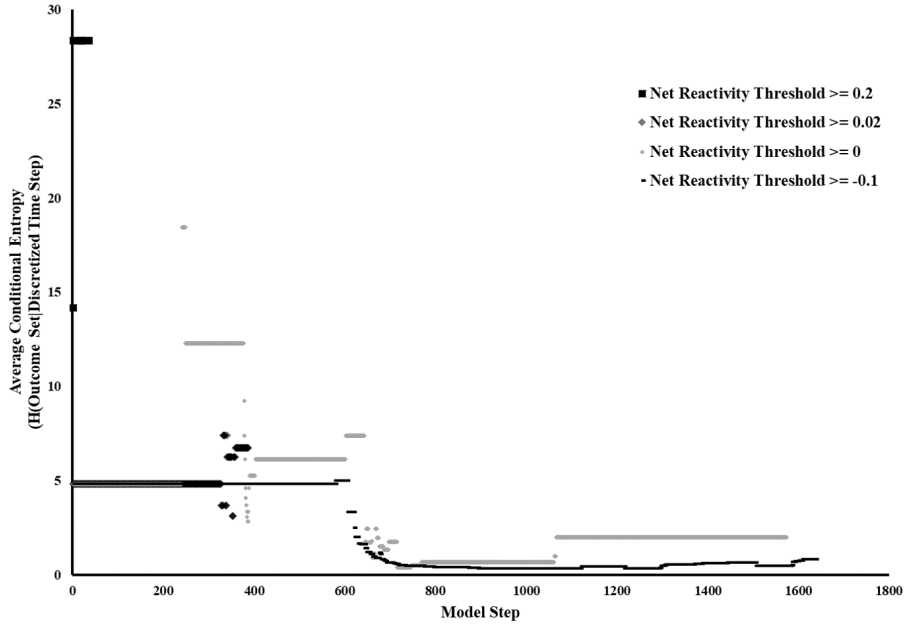


Figure 6.9: Progression of information content in the form of conditional entropy across time steps for models built with state-based discretization.

Time-Related (Time- and Hybrid Time-based) Discretization							State-Based Discretization	
		Primary Time-Step Length (s)						
		1200	120	60	9			Net Reactivity Threshold
Secondary Time-Step Length and Threshold Value	1200: Thresh. 0.2	1.347	1.347	1.347	1.347		27.935	0.2
	1200: Thresh. 0.02		1.304	0.520	0.737			
	1200: Thresh. 0		1.303	0.381	0.686			
	1200: Thresh. -0.1		0.724	0.702	0.590			
	120: Thresh. 0.2	1.347	0.726	0.726	0.726		5.035	0.02
	120: Thresh. 0.02	0.726		0.160	0.327			
	120: Thresh. 0	0.726		0.165	0.686			
	120: Thresh. -0.1	0.726		0.702	0.590			
	60: Thresh. 0.2	1.347	0.726	0.703	0.703		3.758	0.3
	60: Thresh. 0.02	0.703	0.703		0.300			
	60: Thresh. 0	0.703	0.703		0.686			
	60: Thresh. -0.1	0.703	0.703		0.590			
	9: Thresh. 0.2	1.347	0.726	0.703	0.592		2.146	-0.1
	9: Thresh. 0.02	0.592	0.592	0.592				
	9: Thresh. 0	0.592	0.592	0.592				
	9: Thresh. -0.1	0.592	0.592	0.592				

Figure 6.10: Heat map comparison of mean values of average conditional entropy across models and discretization strategies.

6.4 Analysis from the Metrics Comparison Studies

The structure of the metrics studies allowed for an initial evaluation of the difference between modeled system safety and the "ground truth" system safety captured by the DET. For the most part, the models provide roughly the same level of performance with respect to prior assessment accuracy, with time-based models providing slightly more similar results than either the state-based or hybrid time-based models. From this metric alone, the discretization strategies appear comparable in model performance; however, the results from the other metrics studies indicate that there are substantial differences in the performances of DBN SIPPRA health monitoring models based on the discretization approach used to derive model CPTs. The rest of this section expands upon more findings from this study with respect to the different discretization strategies.

6.4.1 Analysis of State-based Discretization Model Performance

The DBN literature search in Chapter 4 found that examples of time-based and state-based discretization methods were being used to develop DBNs for research. When applied to constructing DBNs for SIPPRA, both approaches seem to offer a way to reduce the overwhelming amount of CES data to consider when developing CPTs. Where the data is reduced, however, varies significantly. While adjusting time-based discretizations changes how many measurements are taken across all potential scenarios equally, a change in the threshold for state-based discretization alters the number of scenarios considered for as usable system information. If the measurement threshold would not be reached during a potential scenario, that scenario is not considered in building out the underlying conditional probabilities of that model.

The elimination of certain scenarios during model construction distinguishes the metrics results for the models built with state-based discretization from those built with the time-based discretizaton. First, the range of prior assessment values is considerably larger for state-based models as only similar data are considered for use in constructing the CPTs; adjusting the threshold value changes what data are deemed "relevant." With respect to computational time requirements, DBNs constructed with state-based discretization could not be plotted along a similar power curve like the time-based discretization. Rather, it is the amount of system data above the threshold value that indicates the time required for CPT construction; for this accident space, there are far more instances across more scenarios where net

reactivity was measured between \$0 and \$0.02 than \$0.02 and \$0.2. This explains the large increase in computational time when the threshold was lowered from \$0.02 to \$0. Lastly, DBNs discretized with a state-based approach had the widest range of average entropy values. Although lowering the number of time steps for these models tended to lower average entropy, and therefore reduce the uncertainty, of the accident scenario's identity for any specific point in time, the information content values associated with these models were greater than either time-based or hybrid-based. One reason for this is that net reactivity can be associated with values of cold pool temperature. As such, the threshold selected for the net reactivity also impacts the range of different cold pool temperatures available for constructing model CPTs.

Another effect of eliminating any data from certain scenarios is the transformation of CPTs across models and discretization values. Table 6.4 shows the same portion of a CPT across different time-steps and threshold values considered for this studies. As the threshold and length of time steps get lower, the CPTs begin to approach a similar value; this is to be expected as with the smallest possible steps and no threshold for collecting data, both approaches would capture the same data. Moving away from that point, however is when the CPTs vary drastically. With a reactivity threshold value placed at \$0.2, system data collected for that model would suggest that a scenario in which DRACS could be enhanced or degraded is not possible. With this albeit unrealistic threshold value, model designers are left to figure out an appropriate uninformed relationship to place in the empty spaces of the CPTs. As the threshold is lowered, however, evidence is made available about those scenarios, and the CPT can be filled in using available system data. This

Table 6.4: Portion of "Radial" node CPT over different state- (upper table) and time-based (lower table) discretizations ("SCRAM" node: "SCRAM Failure, Trip Success"; "RPS Pump" node: "Operational")

React. Thresh.	**0.2**			**0.02**			**0**			**-0.1**		
DRACS	**Enh.**	**Nom.**	**Deg.**	**Enh.**	**Nom.**	**Deg.**	**Enh.**	**Nom.**	**Deg.**	**Enh.**	**Nom.**	**Deg.**
Low	No Evid.	0.306	No Evid.	0.209	0.079	0.208	0.068	0.018	0.095	0.084	0.002	0.083
Middle	No Evid.	0.575	No Evid.	0.791	0.371	0.792	0.932	0.184	0.905	0.916	0.061	0.917
High	No Evid.	0.119	No Evid.	0	0.550	0	0	0.797	0	0	0.937	0

Time Step	**1200s**			**120s**			**60s**			**9s**		
DRACS	**Enh.**	**Nom.**	**Deg.**	**Enh.**	**Nom.**	**Deg.**	**Enh.**	**Nom.**	**Deg.**	**Enh.**	**Nom.**	**Deg.**
Low	0	0	0	0.0004	4.3E-06	0.0004	0.001	6.2E-06	0.001	0.001	1.3E-05	0.001
Middle	1	0.011	1	0.9996	0.01	0.9996	0.999	0.010	0.999	0.999	0.010	0.999
High	0	0.989	0	0	0.990	0	0	0.990	0	0	0.990	0

contrasts from the time-based discretization models, where even at the largest time step studied, the time-based discretization had access to available data for those scenarios.

For these reasons, constructing a DBN health monitoring model using a state-based discretization is not a recommended approach. Although they were often faster to construct than their time-based counterparts, DBNs constructed with state-based discretization have too much uncertainty and variability associated with the amount of data above or below different threshold values to consistently predict their performance across the different metrics studied. Eliminating scenarios that do not meet a threshold also presents significant challenges in ensuring that the health monitoring model has appropriate scenario coverage; that is, the model is applicable for different scenarios of system operation. If the model is unusable in certain situations, i.e. when there is a SCRAM failure but not high net reactivity, then it will be not helpful in predicting the system's progression of system health. This problem is only exacerbated if sensors that are used to determine whether a threshold has been reached are inaccurate or broken.

6.4.2 Analysis of Time-based Discretization Model Performance

Although models built with the time-based discretization approach were shown to have the most similar safety assessments relative to the baseline estimates, the other results from the multi-dimensional performance study indicate that models built with the time-based discretization also face limitations of their own. The placement of CPT construction time on a power curve greatly restricts the ability for the model to capture on-line time. For example, in some instances, the SAS4A data set also used to develop this case study, provided data about the reactor simulation at a rate of 0.1 seconds. Using the modeled power curve as an estimate for predicting computational time, the amount of time require to construct a 8,640,000 time step model would be approximately 24.5 million seconds, or about 284 days. For modeling a CES with even more components and failure modes, this would be an overwhelming amount of time and computational requirements. There were even challenges in calculating CPTs for models with larger time steps; even building a model with a realistic monitoring of every two minutes took a considerable amount of time to construct. Time-based discretization models are also constrained by the length of time that they cover; for instance, given the limited capability for GeNIE to tackle models greater than 3,000 time steps, the models with the 9.5 second had to be split up over subsequent models. This space requirement is a major concern for time based models over long forecasting periods; reducing the time of interest to focus on more upcoming events and scenarios may be beneficial for improving the performance of these models.

As shown in Table 6.4, the CPTs for time-based models quickly converge; this is a product of the data from this study, as most of the accident scenarios have relatively constant data over the length of the simulation time. This also helps to explain the stabilizing average information content per model as the simulation progresses. However, as these CPTs become relatively similar, the only noticeable difference becomes the amount of time steps present to represent the 86,400s time period. As the model CPTs reflect a degrading system, more time steps indicate a greater likelihood of system failure. This explains why the time-based discretization models with more time steps have lower safety assessments than those with fewer. Furthermore, with fewer time steps, the beginning of the simulation time (where most of the data volatility occurs), is weighted more heavily against the more constant data of the success scenarios; this helps capture why, in this instance, the system safety assessment of the models utilizing larger time steps are approaching the same estimate as the time-based model that had a data rate measurement equivalent to the data generation rate. It should be noted that in more volatile scenarios, larger time-step values could overstep available information that indicated a SCRAM failure event had occurred. Without that information, the model would provide an incorrect assessment. Furthermore, increasing the number of time steps for time-based models tended to lower average entropy for any specific point in time. With only a set number of scenarios addressed in this case study, providing more information about the branching from "High" to "Low" cold pool temperature restricts the range of possibilities that could occur. This allows the user of these models to limit his or her attention to the possible scenarios based on the available information. Smaller

time steps capture more data variations and data trends earlier, which, when incorporated into a CPT, help to create DBNs that are better aligned with the scenario; however, this results in increased computational requirements.

6.4.3 Analysis of Hybrid Time-based Discretization Model Performance

The hybrid time-based discretization approach was introduced to address some of the challenges faced by the previous two discretization strategies, The aim of this approach is to reduce the computational costs of the time-based discretization strategies by emphasizing scenarios relevant to the model user while minimizing, but not eliminating the scenarios that do not meet the specified interests.

The metrics results from the hybrid models indicate a discretization approach that provides comparable performance while reducing computational requirements. Table 6.5 shows how the CPTs for a hybrid time-based discretization compare to the same CPT for the two related time-based discretization scenarios. Depending on the threshold, some columns of the table may align more to one time-step length or another as the threshold value restricts data from certain scenarios. This is similar to the state-based discretization approach, which is built from data of select scenarios; however, unlike that discretization approach, all scenarios are considered in building the CPTs. This is shown in the computational time required to build a hybrid time-based model's CPTs. In most instances studied, the computational time for these models lie between the computational time for the two measurement

Table 6.5: Comparison of "Radial" node CPTs for time-based discretization and sample hybrid-time discretizations

Select Portion of Radial CPT	Operational Context	1	2	3	4	5	6	7
Time-Based Disc.: 120s time steps	Low	0.0004	4.3E-06	0.0004	0.0004	0.006	0.0002	0.006
	Medium	0.9996	0.010	0.9996	0.002	0.845	0.028	0.831
	High	0	0.990	0	0.998	0.149	0.971	0.162
Time-Based Disc.: 60s time steps	Low	0.001	5.6E-06	0.001	0.0002	0.006	0.0002	0.006
	Medium	0.999	0.010	0.999	0.002	0.845	0.028	0.831
	High	0	0.990	0	0.998	0.148	0.971	0.162
Hybrid Time-Based Disc.: 120s until net reactivity >0.02, then 60s time steps	Low	0.001	9.3E-06	0.001	0.0002	0.006	0.0004	0.006
	Medium	0.999	0.017	0.999	0.001	0.845	0.055	0.831
	High	0	0.983	0	0.999	0.148	0.945	0.162
Hybrid Time-Based Disc.: 120s until net reactivity >0s, then 60s time steps	Low	0.0005	5.6E-06	0.0005	0.0002	0.006	0.0002	0.006
	Medium	0.9995	0.01	0.9995	0.001	0.845	0.028	0.831
	High	0	0.990	0	0.999	0.148	0.971	0.162
Hybrid Time-Based Disc.: 60s until net reactivity <0, then 120s time steps	Low	0.0004	4.3E-06	0.0004	0.0005	0.006	0.0002	0.006
	Medium	0.9996	0.010	0.9996	0.003	0.845	0.028	0.831
	High	0	0.990	0	0.997	0.149	0.971	0.162

rates as they remove a number of excess measurements from scenarios that are of lower interest. However, it should be noted that as the number of scenarios meet the specified threshold, the additional time required to check scenario data causes these models to become equivalent, or even become greater than, the time required for a model constructed using single time-based discretization with the smaller time steps.

The performance of the hybrid time-based models vary based on the time-step lengths used as well as the threshold value assigned to switch from one rate to another. This can be seen in the stark difference in the models' system safety estimates. Here is another instance in which the discretization of the operational data is affecting model performance. For models whose primary time-step length is smaller than the secondary rate, more emphasis is placed on data after the threshold value has been met. In this situation, where an accident has already occurred, this switch gives data further away from the accident more weight in the CPTs. On the other hand, time step rates that are smaller immediately following an accident

prioritize data closer to an accident that can offer a better picture of what is going on. These rates can be relaxed once more normal values have been met. This is also shown in the average conditional entropies for these models, in which the two scenarios present different amounts of knowledge about the current situation.

The values generated during the information content study highlight the distinction of the hybrid time-based discretization approach from either the time-based or state methods to further separate accident scenarios based on whether a threshold state is met. Once a threshold is reached, those sequences are now using data collected over a secondary rate that is specifically unlike the primary rate. This allows more granularity and greater certainty in knowing which operational scenario is currently experienced by the system. However, this split is greatly impacted by the specific threshold chosen. A threshold in which either all systems respond to at the same time or don't respond to at all will leave the system with the same information content as a time-based model built using either the secondary or primary rate, respectively; this appears to have been the case for the models in the lower left region. Structuring the threshold so as to provide a gradual spread of scenarios over time may provide more scenario insight.

6.4.4 Comparison across Model Performances

Ultimately, the results from the study shows that in this scenario, models built with a hybrid time-based discretization method provide a useful compromise between the operationally dependent but often incomplete state-based models, and

Table 6.6: Metric summary comparisons

	Time-based	State-based	Hybrid Time-based
Risk Alignment with Underlying DET Assessment	Comparable (More Accurate)	Comparable (Less Accurate)	Comparable (In-Between)
Description of CPT Development Time	Defined power curve	Disjointed step function	Bounded between Time-based values
Information Content: Avg. Conditional Entropy	Decreasing with more time steps	Highest	Either comparable to time-based or lower

the all-inclusive but time-consuming time-based models. If model selection was solely based on time or assessment accuracy, the time-based models constructed with 1200s or 9s time steps would be the top choice, respectively. However, because both are limited in providing meaningful knowledge about the accident scenario currently experienced, the hybrid time-based model that starts at 120 second time steps and transitions over to 60 following a reactivity threshold of $0 might also be another choice to consider. These decisions require understanding the model user's needs and subsequent consequences for system failure.

6.5 Implications of Study Results and Analysis

6.5.1 Applying Discretization Strategies to Other CES Health Management Scenarios

Table 6.6 summarizes the broad findings of applying the three performance metrics on DBN models constructed using each of the different data-stream discretization approaches. The differences in metric values across the three discretization strategies highlight the variations in model performance that arise when DBN CPTs are parameterized using data collected over different time windows and system characteristics. These findings serve as an initial step towards better understanding

the impact of decisions made by dynamic risk model developers when determining what time discretization to use for a particular operational scenario.

Ultimately, the range of values provided by these metrics indicate that the performance of SIPPRA health monitoring models is multi-dimensional, and cannot be narrowly constrained to a single metric. This is important when considering an appropriate discretization approach for developing, as there exists opportunities for trade-offs based on different risk model user preferences, needs, and requirements. For example, in the SFR case study, larger time steps may result in shorter computational time to develop the CPT, but this comes at a loss of information per model step. Likewise, smaller time steps and more relaxed thresholds provide more information about the current scenario, but require significantly more time to construct the model. A hybrid time-based model may address reduce some of these limitations while providing more certainty about the trajectory of the current accident sequence, but it is still often bounded in performance between time-based models constructed using either rate. Considering these trade-offs, as well as additional ones from other performance metrics mentioned in Chapter 3, will provide better understanding on how DBN discretization strategies impact SIPPRA model performance and allow risk model developers clearer insight for designing improved system health assessment models.

It should be noted that although these results are valid for this particular scenario and CES, inherently, conclusions cannot be separated from the purpose behind building a model and the assumptions that went into constructing it. This SFR TOP scenario has a number of unique features that may have contributed to

these results. First, the scenario outlined in this case study is the aftermath of a external disaster that has damaged the system; as a result, the focus of this scenario is not the prevention of a disaster (that has already happened), but rather a better understanding of whether the system will be able to return to normal operations. To that end, the time period covered for this accident sequence is skewed far beyond most operational changes would occur to the system. As a result, the volatility of the parameters lessens over time, making inspection beyond a certain point unnecessary. This is seen in the relatively constant CPTs constructed over time. Despite the additional information, the data was still incorporated into the CPTs at the same rate (as in, doubling the time steps over the period of time would just double the count of data to consider).

Understanding CES operational scenario nuances is important when considering discretization strategies for a health monitoring model design, particularly in the case for hybrid time-based discretization. As previously mentioned, models built to assess system health within the context of the scenario in this study are intended to reflect the health of a system that has already experienced damage. Given that insight, the hybrid-time structure best suited for this study is one that collects more system data early on, gradually loosening restrictions once a certain threshold has been reached. Other CES operational data may appear differently than the accident data used in this study, however. For example, the scenario of interest may be the lead-up to a potential system failure based on component degradation or human intervention. In that instance, system parameter values begin as baseline values but become more abnormal over time. There, it is reasonable to increase measur-

ing rates once an abnormal threshold is met, as the aim there is to identify the likelihood of system failure as early as possible. To determine which discretization approach would be best suited for that CES scenario would require a similar study to the one carried out here that takes into consideration the operational nuances and requirements of the CES of interest.

This prognostics modeling architecture is well-suited for models that represent CESes with known distinct failure modes that take time to develop. These prognostic models provide insight into potential future system failures; however, time is required to collect the necessary data to support the identification of specific future scenarios or current system health. If system failure follows immediately after an event, this type of model analysis is limited in its usefulness. Failure modes should also be known or expected for this type of analysis. To construct model CPTs, data should either available or able to be simulated; however, that requires in-depth knowledge of the system. This model analysis is also beneficial when the approach to managing and mitigating CES failure modes are wildly differently. By identifying specific failure modes that are more likely to occur than other under certain operational conditions, operators would be able to prioritize addressing those types of failures over less likely ones.

6.5.2 Applying Study Methodological Process to Other SIPPRA Model Design Decisions

The results from this study provide further insight into how discretization strategies affect different aspects of model performance, and also serve as a validation for the use of the methodological process applied in this study to investigate aspects of CES health monitoring model design decisions. Effectively discretizing data streams is just one open question in the area of SIPPRA and CES health management; there are many others that would greatly benefit from a similarly structured comparison study. These potential research areas may be focused, like studying the impact of different data binning discretization practices on DBN health assessments, or broad, like comparing different approaches to health monitoring. Tackling these research questions would require a similar approach: identifying the different model designs for the comparison, selecting the performance metrics used to compare the model designs, and then applying them on a specific CES health monitoring scenario and analyzing the results of the comparison. The continual process of studying the impact of different SIPPRA approaches on model performance would support a richer understanding of CES health and provide better approaches for effectively monitoring and managing them.

6.6 Chapter Conclusion

This chapter presents the results of comparing fifty-six DBN-based SIPPRA health models for a sodium fast reactor experiencing a transient overpower built using different discretization techniques outlined in Chapter 4 and compared across different performance metrics from Chapter 3. Although the risk assessments for each model are comparable to one another, the computational time and information content for each model vary drastically. This indicates that the modeling decisions one makes in the formation of health monitoring models has an impact on their performance. Although the state-based discretization models offers a fast solution and general approximation of system safety, their approach of removing possible scenarios from the risk analysis and reliance on system-generated data reduce the coverage of the model and threaten usability issues during accident scenarios when the state threshold is not reached. Time-based discretization models provide the greatest accuracy in this study, but face a significant computational burden in model development. Hybrid time-based discretization offers a compromise between computational time, information content, and alignment value. By capturing the relevant scenarios, this approach offers an alignment of scenarios most similar to current practices. Ultimately, the results of the study show that other performance metrics are needed outside of considering assessment accuracy in determining appropriate discretization parameters for optimal performance. This study helps to provide better understanding on how DBN time-step discretization impact performance through the variations of these metrics; prioritizing certain metrics over others will allow

risk model developers to design useful tools to provide risk managers clearer insight into potential accident scenarios and help to develop improved risk management strategies for CESes.

Chapter 7: Summary, Contributions, and Suggested Work

7.1 Summary of Research Conclusions and Contributions

This research expands upon the current understanding of the impact that time segmentation of continuous time-series data has on DBN system-level health assessments using SIPPRA framework. The work is structured across four research activities to meet distinct research objectives. Conclusions and technical contributions are made at the overall research level and also at the objective and supplementary levels below that. Figure 7.1 provides a graphical representation of the technical contributions for this research and will be referred to throughout this section.

Given the novelty of SIPPRA and the limited understanding of system health

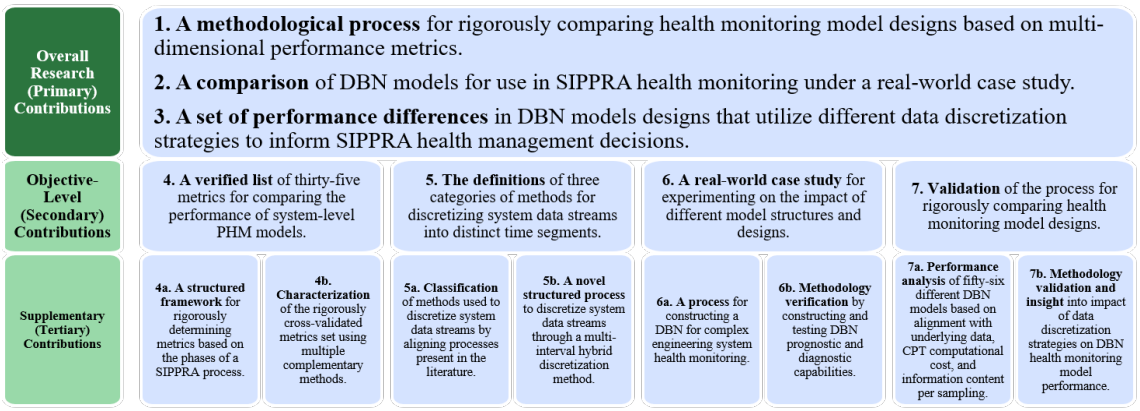


Figure 7.1: Technical contributions from this research separated into overall research (primary), objective-level (secondary), or supplementary (tertiary) contributions.

management for CESes, the first contribution from this research is the development of a methodological process for rigorously comparing CES health monitoring model designs based on multi-dimensional performance metrics (TC 1). Before the process could be applied to the underlying research question about the impact of time discretization strategies on the performance of DBN-based health monitoring models, three elements were required: a set of performance metrics to evaluate SIPPRA methods, a list of methods used to discretize continuous time-series data for DBNs, and a real-world case study for analyzing the impact of different design choices on health monitoring models.

7.1.1 Defined Set of Performance Metrics for SIPPRA Models

This work produced a set of thirty-five metrics that could be used to compare the performance of different system-level health monitoring models as a multi-dimensional concept (TC 4). Current metrics used to evaluate model performance in PRA and PHM techniques are not sufficient for system-level health models that utilize SIPPRA techniques. Developing a rigorous process to identify performance metrics as indicators of a successful completion of SIPPRA tasks (TC 4a) ensured that the metrics set was comprehensive and verifiable. These performance metrics are designed to be evaluated for a specific a system or operational environment, enabling meaningful and justifiable comparisons across model designs. These metrics were then classified based on their functionality and cross-validated as a viable set using multiple complementary methods (TC 4b). The results of applying a selec-

tion of these metrics to a simple model design decision highlight their usefulness in providing measurable and structured means to compare different models.

7.1.2 Defined Data Stream Discretization Strategies

This work structured three categories of methods for discretizing system data streams into distinct time segments (TC 5). A review of the recent reliability literature indicated that researchers have relied on the use of only two discretization methods for discretizing DBNs: time-based and state-based (TC 5a). However, these approaches are shown to not always respond appropriately to changes in a complex engineering system timeline. Between the capability gaps of these two discretization lies a third approach: a multi-interval hybrid discretization that adjusts its sampling frequency based on operational and environment changes. This work presents and verifies the framework to develop a model using this discretization process (TC 5b) through a simplified model of a CES undergoing an accident sequence. The results of the toy problem indicate that using a multi-interval discretization method allows for greater flexibility in the data that is used for the model, and therefore its alignment with expected operational scenarios.

7.1.3 Developed CES Case Study

This work produced a real-world case study of the operational after-effects of a SFR experiencing a transient overpower; this serves as a structured means for studying the impact of different DBN structures and designs meant to capture CES

health (TC 6). Structured processes were defined for converting simulated operational nuclear data into the DBN's node structure and CPTs. This work introduced a framework to use for constructing DBNs for CES health monitoring based on connecting operational environments, component health, and human interventions, to system failures and prognostics (TC 6a). An equally structured method was also designed for this work to develop DBN CPTs based on case study data generated across different measuring periods. Model verification indicated that the proposed DBN structure is appropriate for design experimentation as the model responds to new system data with either increased or decreased likelihood of system failure (TC 6b).

7.1.4 Integrated Previous Results to Validate Model Comparisons

Using the developed performance metric set, data-stream discretization classifications, and SFR case study, this work validated the methodological process for comparing SIPPRA-based DBN health monitoring model performances through a real-world case study (TC 2, TC 7). The performance of fifty-six different DBN models were analyzed based on three performance metrics: assessment accuracy, CPT computational cost, and average information content per sampling (TC 7a). Results from the comparison validated the methodology used in this research and helped provide insight into the impact of data-stream discretization on DBN health monitoring model performance (TC 7b). As the comparison was made across DBNs with CPTs built using different discretization strategies, it specifically identifies the

contrasting model performances based on those differences (TC 3).

The results of this study indicate that the selection of the time discretization strategy impacts different aspects of model performance. In terms of overall safety assessments, all three provided comparable values, with time-based values providing the most similar results to the underlying estimate from the DET. Those methods, however, were particularly susceptible to a wide range of CPT development time, making some theoretically possible time steps unsuitable for potential operational use. Relying on state-based discretizations produced DBNs that did not provide coverage for possible scenarios that did not generate system data which met the measurement threshold requirement. The novel hybrid time-based discretization proposed in this research not only provided comparable accuracy at a fraction of the computational time required for time-based models, but also provided further information content for relevant scenarios. This indicates that the hybrid-interval method is best suited for CES health monitoring models constructed with a SIPPRA as it can respond accordingly to the operational needs of the user.

7.2 Work Products

7.2.1 Models and Programming Codes

DBN models consist of a node-arc network structure and the CPTs that indicate the probability of node state transitions; as such, a substantial portion of the products from this research are the DBN model structures and associated code for CPT quantification. Two different model structures were made made: a generic

DBN framework for modeling CES PHM and a quantified DBN derived from that framework that is specific to the SFR case study from Chapter 5.

The list of models considered as products from this research are:

- DBN Framework for Generic CES PHM.
- DBN structure built for SFR case study described in [17].

In order to quantify the CPTs within the DBNs, code was developed to convert the case study's operational data into usable information for the model. Since the discretization technique used in a DBN determines what information is collected for the CPTs, separate codes were developed for each discretization method. While the codes are structured for data based on the case study, they are modifiable and work with other input data from this or other CES operational scenarios.

The list of annotated programming codes for CPT quantification are as follows:

- Time-based discretization.
- State-based discretization.
- Hybrid time-based discretization.

This code is available to use with other DBN-related SIPPRA research within the Systems Risk and Reliability Analysis (SyRRA) lab; dissemination of the codes is possible upon contact and further discussion with Austin Lewis at adlewis@umd.edu.

7.2.2 Publications

7.2.2.1 Journal Papers

- Austin D. Lewis and Katrina M. Groth. "A dynamic Bayesian network structure for joint diagnostics and prognostics of complex engineering systems". *Algorithms* 13 (Mar. 2020). Special Issue Bayesian Networks: Inference Algorithms, Applications, and Software Tools, pp. 64+. DOI: 10.3390/a13030064. Invited paper & cover article.

- Austin Lewis and Katrina Groth. "Metrics for evaluating the performance of complex engineering system health monitoring models" (). Accepted in Reliability Engineering & System Safety

- Austin Lewis and Katrina Groth. "Comparison of performance of DBN models for SIPPRA-based health monitoring based on different data stream discretization methods" (). In progress

7.2.2.2 Conference Papers

- Austin Lewis and Katrina Groth. "A review of methods for discretizing continuous-time accident sequences". *Proceedings of the 29th European Safety and Reliability Conference* (2019), pp. 754–761

- Austin Lewis and Katrina Groth. "A multi-interval method for discretizing continuous-time event sequences". *2021 Annual Reliability and Maintainabil-*

ity Symposium (2021), pp. 1–7

- Austin Lewis and Katrina Groth. "Impact of different time discretization methods on dynamic Bayesian network-based dynamic probabilistic safety assessments". *Proceedings of the 2021 International Topical Meeting on Probabilistic Safety Assessment and Analysis* (2021), pp. 410–419

- Austin Lewis and Katrina Groth. "Impact of complex engineering system data stream discretization techniques on the performance of Dynamic Bayesian Network-Based Health Assessments" (). In progress.

7.2.2.3 Presentations

- 2019 European Safety and Reliability Conference (ESREL 2021). Hannover, Germany. September 22-26. Conference Paper Presentation and Poster.

- 2021 Reliability and Maintainability Symposium (RAMS 2021). Orlando, FL (presented remotely). May 24-27. Conference Paper Presentation.

- 2021 International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA 2021). Columbus, OH (presented remotely). November 7-12. Conference Paper Presentation.

- 2021 Society for Risk Analysis Annual Meeting (SRA 2021). Washington, DC (presented remotely). December 5-9. Conference Paper Abstract Presentation.

- 2022 Probabilistic Safety Assessment & Management Conference (PSAM 2022) (anticipated). Honolulu, HI. June 26-July 1. Conference Paper.

7.3 Opportunities for Future Work

As an initial investigation into the impact that applying different time-discretization strategies has on the performance of SIPPRA-based DBNs, there are several areas to further our understanding of CES health management.

7.3.1 Expanding SFR CES Case Study

- The model used in this research can be expanded by adding additional nodes and arcs to the structure to provide a more detailed representation of SFR system operations following a transient overpower. One such area to investigate would be a richer depiction of the operator interventions currently present in the network model as "Cold Pool Temperature"-"Human Intervention"–"DRACS" causal triangle. This would entail further discretization of the cold pool temperature and the incorporation of thermal pumps into the structure.

- The DET used to represent this accident event provided more information about the states of other components, including the status of primary and secondary pumps. Incorporating information about these components could provide either more understanding about the current scenarios explored in the case study, or provide more information about the impact that time discretiza-

tion strategies have on DBNs constructed for dynamic PRA.

- The hybrid-interval approach was only applied in the case study as a one-time breach of a specific value of net reactivity. However, in most accident scenarios, there are multiple events or thresholds which need to be considered when calculating the future progression of system health. Introducing more complicated discretization strategies would better reflect operational reality and could provide additional insight into the validity of these models as useful decision support tools.

7.3.2 Performing Additional CES Case Studies

- A number of conclusions were drawn on the contrast of performance of models utilizing different time discretization strategies from the model comparisons made in the SFR case study. Carrying out another case study on a different system would help to validate the applicability of these findings across CESes. This secondary case study could be on another accident scenario for a different nuclear reactor, other systems within the nuclear power plant, or even in a completely separate system domain.

- Only a few metrics were selected to analyze the differences of each DBN model built under different time discretization values changed to consolidate research scope. A more detailed effort is needed to provide more differences between the models. This would entail the use of other prognostic and diagnostic model metrics, including outcome accuracy and prognostic horizon. Another metric

to consider further is model uncertainty. As the DBNs are constructed using different subsets of available operational data, there is inherent uncertainty about the goodness of form that should be studied in deeper detail [2].

7.3.3 Moving Beyond DBNs for CES Health Management

- This research relies on DBNs and the construction of their CPTs as the primary source of comparison for model performance. However, DBNs are inherently limited in their ability to represent scenarios in which no data exist. Currently, expert judgement has identified likely CPT values for instances in which no data exists; however, this raises the possibility of inaccurate model outputs. Recently, the use and implementation of probabilistic programming languages [97] enables causally-driven models to be developed outside of these limitations. A better understanding and utilization of these types of models may provide more accurate measurements of the likelihood of an accident scenario or system failure than is currently available.

- The development time for the CPTs for this model could be improved by implementing machine learning techniques that identify and align scenario characteristics associated with related branches in the underlying DET. A further analysis of implementing such clustering techniques may be useful in identifying what other available methods can be leveraged to reduce computational requirements for developing the model.

- The framework for discretizing continuous-time operational data streams into

individual segments can be applied as a pre-processing approach for different system-level PHM techniques, including machine learning and constructing neural networks. Implementing these different time discretization strategies in the context of these different methods may provide further insight into more effective ways of capturing operational and accident data.

7.4 Potential Impact

"Fundamentally, how do we address the challenges with high impact low probability events within conventional prioritization and ROI methodologies...particularly if the cost is high...it's precisely because of the tremendous success and vigor of most of our complex systems that we tend to take their reliability for granted. But we obviously cannot." - Richard Laudenat, Immediate Past President, American Society of Mechanical Engineers (ASME) [3]

In his opening remarks at the ASME Safety Engineering and Risk Analysis Division (SERAD) and the University of Maryland's Center for Risk and Reliability (UMD-CRR) Joint Interactive Seminar and Pre-Workshop on Intersection of PRA and PHM in October 2020, Laudenat stressed the importance of leveraging computing technologies and process simulation tools on complex industrial applications [3]. Although they represented current perspectives on PRA and PHM practices, respectively, Smith [98] and Droguett [99] echoed the same message at the seminar: there is a clear gap in detailed system-level health management practices for CESes.

This research bridges that gap by studying how the application of different

time-discretization methods on continuous operational timelines impacts health insights formed from system-level DBN models. By prioritizing data from operationally relevant accident scenarios and associated conditions as input into DBN CPTs, model designers can provide targeted tools for improved decision support. The use of on-line data to update the health information within system logic structures addresses the challenges of monitoring CES health at the system level, allowing for a more updated assessment of the system's health and provides valuable information for responsive system maintenance and risk management practices.

A primary result of this research is a better understanding of how the construction of SIPPRA-based DBNs impacts the system-level health insights formed from them. The conclusions and technical contributions from this research have significant implications for how CESes are modeled to enable insight into their health. When modeling a CES, there are limitations on data storage and computational availability that restrict what is theoretically possible in a computational setting. Knowing the trade-offs in data requirements and health forecasting capabilities between different time-discretization methods will lead to SIPPRA-based DBNs that are structured for system-level health management and are reflective of the monitoring needs and operational restrictions of a critical system.

In addition to the specific insight about SIPPRA-based DBN time-discretization methods for the health management for CESes, this research provides a structured framework for further efforts to establish rigorous SIPPRA approaches for CESes. By identifying model-design practices of interest, a case study environment, and metrics for PHM comparison, multiple system health models can be constructed for

a single scenario and evaluated over different aspects of model performance. Since the performance metrics are applicable to any health management model, changing either the model-design choice or the system scenario provides new insight into SIPPRA approaches for CESes. If this process is performed repeatedly over a range of model parameters and scenarios, there will be well-documented procedures for developing SIPPRA-based DBN models for CES health management for a given system and across operational scenarios. This information will enable model designers to develop more effective models for CES health management.

Appendix A: Case Study Data

The data used in this case study comes from simulations run by Jankovsky et al. [93] using SAS4A/SASSYS-1 and PRIMAR4 models as part of a project to develop methodologies for merging Dynamic Event Trees (DETs) with operator actions. The DET was designed to "investigate the effects of various mitigating actions and uncertain plant parameters in an SFR following an inadvertent insertion of reactivity." It consisted of seven branching conditions resulting in 2 to 10 child branches each, and two ending conditions: failure by fuel relocation or failure by loss of cladding thickness. This resulted in a collection of 2,052 accident sequences that had the outcomes of model success, clad relocation, or temperature failure. A SAS4A/SASSYS-1 model was then used to calculate the operational data along the tree for each branch.

For this work, a modified version of that tree was used to study the effects of data stream discretizations on DBNs for SIPPRA-based health management. Although the DET used in this research had the same outcomes (failures by fuel relocation or loss of cladding and a successful run of the model), the scope of the scenarios were less detailed, focusing primarily on the human intervention of the direct reactor auxiliary cooling system (DRACS). As this tree only considered the

Table A.1: Branching conditions used in the modified DET for this research

Branching Condition	States	Marginal Probabilities
1: Reactivity Coefficients	10 Different Reactivity Coefficient States	0.1
2: TOP Magnitude	0.06	0.901
	0.3	0.090
	0.5	0.009
3: SCRAM Functionality	SCRAM and Trip Success	0.99999971
	SCRAM Success, Trip Failure	$1.4 * 10^{-}9$
	SCRAM Failure, Trip Success	$1.4 * 10^{-}9$
	SCRAM and Trip Failure	$2.9 * 10^{-}7$
4: RPS Functionality	Operational	0.5
	Not Operational	0.5
5: Human Intervention with DRACS	Intervene	0.5
	Don't Intervene	0.5
6: DRACS Functionality	Enhanced	0.9
	Degraded	0.1

effects of a TOP, this tree consisted of the following branching conditions presented in Table A.1:

The simplification of the previous tree resulted in the use of only 1,920 of the 2,052 potential accident sequences. Furthermore, the marginal probabilities were adjusted from the previous research to account for assumption that a TOP had already occurred.

Due to the limitations of feasibly presenting the data in an effective manner, the data is not provided in this work. However, requests for more detail can be made by contacting Austin Lewis at adlewis@umd.edu.

Appendix B: Case Study DBN Formation

This appendix describes the formation of the DBN for the case study scenario of a sodium fast reactor experiencing a SCRAM mechanism failure during a transient overpower.

Overview

Figure B.1 is the illustration of the DBN model nodes and directed relationships within their respective information regions. This network structure was determined by a general understanding of the nature of the SCRAM mechanism and the two described failure modes in the study by Jankovsky, et al [93]. Table B.1 lists the nodes constructed in the model, the information region they are located in, the number and the value of node states.

In addition to a network structure of nodes and directed arcs, a DBN model requires associated conditional probability tables, as well as an additional initial distribution table for the dynamically changing nodes. Therefore, the following is the list of CPT tables needed for the model designed for the SFR scenario:

- **Static Conditional Probability Tables**

1. $P(SCRAM\ State)$
2. $P(RPS\ Pump)$

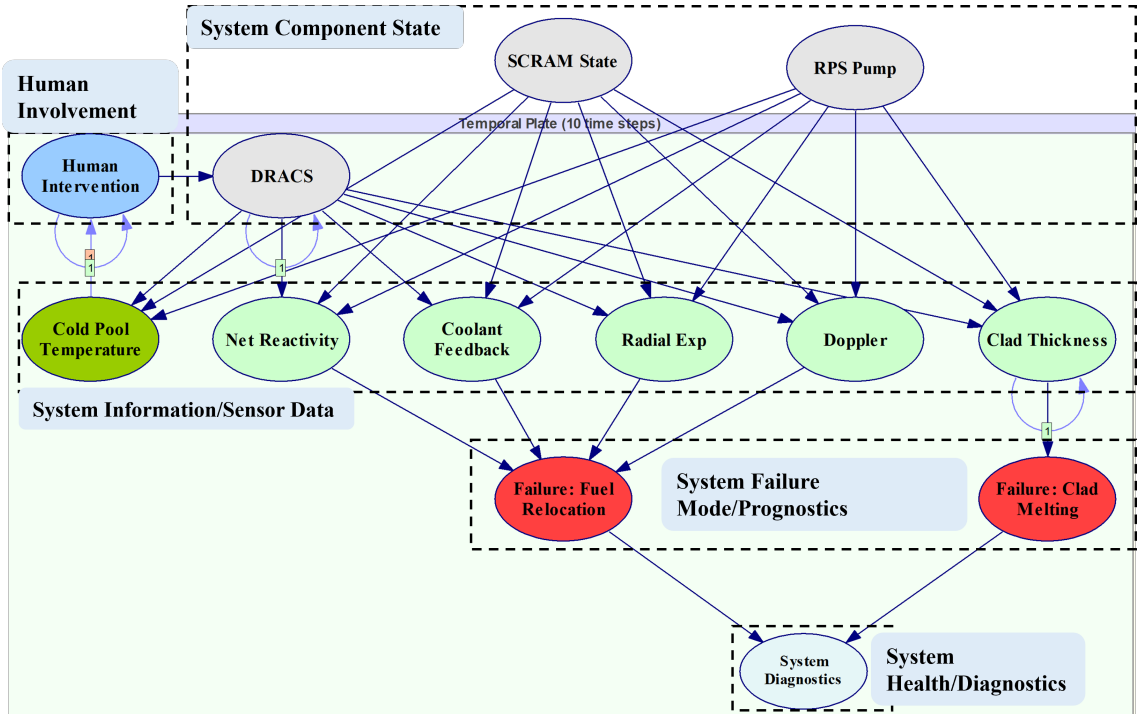


Figure B.1: DBN node structure and relationship graph for SFR TOP case study

Table B.1: Model nodes and node states

Node Name	Type of Node	Number of States	General State Descriptions
SCRAM State	System Component	4	SCRAM and Trip Success, SCRAM Success and Trip Failure, SCRAM Failure and Trip Success, SCRAM and Trip Failure
RPS Pump	System Component	2	Operational, Not Operational
DRACS	System Component	3	Degraded, Nominal, Enhanced
Human Intervention	Human Involvement	3	Yes, No, Undecided
Cold Pool Temperature	System Information/ Sensor Data	3	Below 753K, Above 753K
Net Reactivity	System Information/ Sensor Data	3	Low, Medium, High
Coolant Feedback	System Information/ Sensor Data	3	Low, Medium, High
Radial Expansion	System Information/ Sensor Data	3	Low, Medium, High
Doppler	System Information/ Sensor Data	3	Low, Medium, High
Clad Thickness	System Information/ Sensor Data	11	90-100% (by percent), Below 90%
Failure: Fuel Relocation	System Prognostics	2	Yes, No
Failure: Clad Fraction	System Prognostics	2	Yes, No
System Diagnostics	System Diagnostics	2	Yes, No

3. $P(Human\ Intervention)$
4. $P(DRACS)$
5. $P(Cold\ Pool\ Temperature|SCRAM\ State, RPS\ Pump,\ DRACS)$
6. $P(Net\ Reactivity|SCRAM\ State, RPS\ Pump,\ DRACS)$
7. $P(Coolant\ Feedback|SCRAM\ State, RPS\ Pump,\ DRACS)$
8. $P(Radial\ Expansion|SCRAM\ State, RPS\ Pump,\ DRACS)$
9. $P(Doppler|SCRAM\ State, RPS\ Pump,\ DRACS)$
10. $P(Clad\ Thickness|SCRAM\ State, RPS\ Pump,\ DRACS)$
11. $P(Failure\ :\ Fuel\ Relocation|Net\ React., Coolant\ Feedback, Radial\ Exp., Doppler)$
12. $P(Failure : Clad\ Melting|SCRAM\ State, Power - to - Flow)$
13. $P(System\ Diagnostic|Failure : Clad\ Melting, Failure : Fuel\ Relocation)$

- **Dynamic Conditional Probability Tables**

1. $P(Human\ Intervention|Human\ Intervention(t-1), Cold\ Pool\ Temperature(t-1)$
2. $P(DRACS|Human\ Intervention, DRACS(t-1)$
3. $P(Clad\ Thickness|SCRAM\ State, RPS\ Pump, DRACS, Clad\ Thickness(t-1)$
4. $P(Outlet\ Temperature|SCRAM\ State, Outlet\ Temperature(t-1))$

To quantify these tables, either prior expert knowledge or operational data is required. For this case study, a hybrid approach was used to complete the CPTs. Expert-based opinions were determined from either source documents (i.e.,

the PSID) [100] or mentioned in the study by Jankovsky, et al. [93]. The operational data used was generated based on the description provided in Appendix A.

Coding Scenario Information

The structure of the model relies on data from both system sensors and other monitoring equipment, in addition to situational information regarding different accident scenarios that the reactor may be exposed to. The scenario information describes the different conditions following the accident event as well as the simulated outcome of that particular event sequence (successful system survival or system failure). To allow the DBN model's CPTs to be constructed from the operational data attached to the different scenario sequences, data measurements from the different system parameters (Cold Pool Temperature, Net Reactivity, Coolant Feedback, Radial Expansion and Doppler Feedback) were assigned a number based on the amount of bins available for discretization. For this study, the sensor data were treated with either "High", "Medium", or "Low" relative to baseline operating information.

Creating Operational Timelines for Different Timelines

In order to create the conditional probability tables for each of the nodes, the operational data is formatted into a single timeline. Relevant information was identified from both models; in the case study, the system sensors were the primary indicators for the automatic SCRAM and trip mechanism. Those two data sets were then merged together and sorted based on the timing that the information was received. In some instances, data from one model was received, and not from the other. In those instances, the newer information replaces the earlier measurements received from the same system sensor, whereas all other system information remains

the same.

Separating Scenario Outcomes by Accident Node State (SCRAM State)

Based on the model structure shown in Figure B.1, the accident node "SCRAM state" is connected to each of the system information/sensor data as well as to the system prognostic nodes; as a result, it is important to be able to classify the accident scenario sequences by their accident states. This requires the operational data to be categorized according to what accident the reactor experienced. This situational information is critical for constructing the CPTs for the CES prognostics nodes.

Creating the CPTs

The conditional probability tables for each node were created by measuring the frequency of different data combinations with respect to the different node states and supplementing the available data with expert opinions when there was not information available. For example, for the dynamic table for the inlet temperature, the value $P(Net\ Reactivity|SCRAM\ State,\ RPS\ Pump,\ DRACS)$ was calculated for the different values of net reactivity that were associated with each of the different SCRAM/trip and DRACS states. Each state was then normalized over the same parent node conditions.

The manner of providing expert opinion for the CPTs depended on the nature of the CPTs as well as the location of the node:

- **Accident State (SCRAM State):** The expert opinion used to construct the CPT for the "SCRAM State" node was based on probabilities taken from the PRISM Preliminary Safety Information Document (PSID) and listed in

Table B.2: Prior distribution for SCRAM states.

SCRAM State	Prior Distribution
SCRAM and Trip Success	$1 - \Sigma P(failure) \approx 1$
SCRAM Success and Trip Failure	$1.4 * 10^{-}9$
SCRAM Failure and Trip Success	$1.4 * 10^{-}9$
SCRAM and Trip Failure	$2.9 * 10^{-}7$

Table B.2 [100].

- **System Information/Sensor Data (Cold Pool Temperature, Net Reactivity, Coolant Feedback, Radial Expansion and Doppler):** The initial distribution of each measurement was assumed to be within the normal operating baseline of the variable; therefore, the missing measurement for each system sensor would be placed in the middle bin marked "Medium."

- **System Prognostics (Failure: Fuel Relocation, Failure: Clad Melting):** Due to the limited number of scenarios that result in an overall system failure from the two outcomes specified in the case study, the prognostics CPTs are the most incomplete in the model. To fill the CPTs with values that would not skew the outcome, it was assumed that the empty parent condition cases would result in no likelihood of system failure or further system degradation.

- **System Diagnostics (Diagnostics):** For this model, it is assumed that if a failure occurs, then the system is not healthy. As such, the CPT for the diagnostics node "Diagnostics" is constructed using the conditional probability table shown in Table B.3.

Table B.3: CPT for "System Diagnostics" node.

Diagnostics	Distribution			
Failure: Fuel Relocation	True		False	
Failure: Clad Melting	True	False	True	False
Healthy	0	0	0	1
Not Healthy	1	1	1	0

Appendix C: Common Case Study CPTs

This appendix presents the CPTs that are common across the different DBN models analyzed for this study. These CPTs capture the causal relationships for the boxed nodes in Figure C.1.

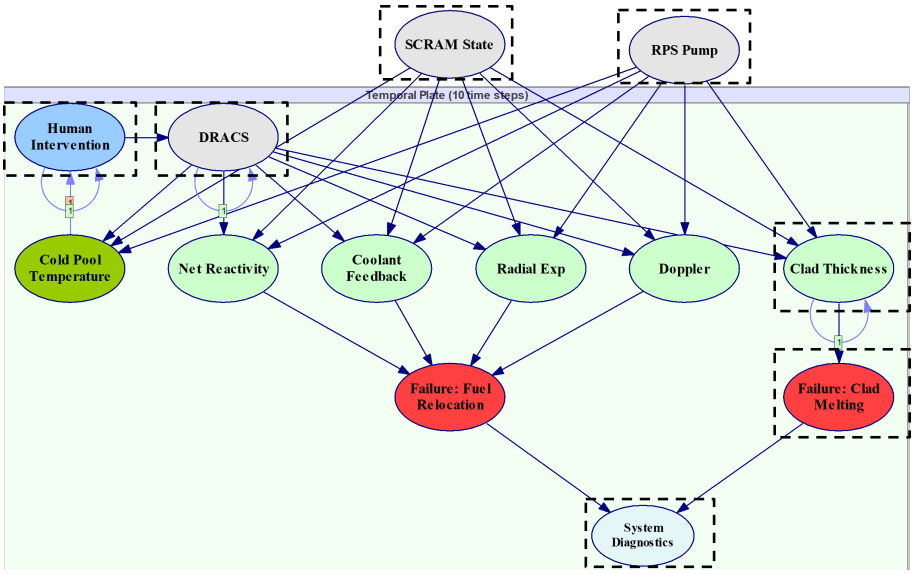


Figure C.1: This appendix provides the CPTs for the boxed DBN nodes.

SCRAM State

SCRAM State	
SCRAM and trip success	~1
SCRAM success, trip failure	1E-8
SCRAM failure, trip success	1E-8
SCRAM and Trip failure	1.3E-9

RPS Pump

RPS Pump State	
Operational	0.5
Not Operational	0.5

Human Intervention

Human Intervention	t=0	
Cold Pool Temperature	Below 785	Above 785
Intervene	0	0
Not Intervene	0	0
Undecided	1	1

Human Intervention	t>=1					
Cold Pool Temperature	Below 785			Above 785		
Human Intervention (t-1)	Intervene	Not Intervene	Undecided	Intervene	Not Intervene	Undecided
Intervene	1	0	0	1	0	0
Not Intervene	1	0	0	1	0	0
Undecided	0	0	1	0	0	1

DRACS

DRACS	**t = 0**		
Human Intervention	Yes	No	Undecided
Enhanced	0	0	0
Nominal	1	1	1
Degraded	0	0	0

DRACS	t>=1								
Human Intervention	**Yes**			No			Undecided		
DRACS [t-1]	Yes	No	Undecided	Yes	No	Undecided	Yes	No	Undecided
Enhanced	1	0.9	0	0	0	0	0	0	0
Nominal	0	0	0	1	1	1	1	1	1
Degraded	0	0.1	1	0	0	0	0	0	0

Clad Thickness

Clad Thickness	**t=0**											
SCRAM State	Success and trip success						SCRAM success, trip failure					
RPS Pump	**Operational**			**Not Operational**			**Operational**			**Not Operational**		
DRACS	Enh.	Nom.	Deg.	Enh.	Nom.	Deg.	Enh.	Nom.	Deg.	Enh.	Nom.	Deg.
99-100%	1	1	1	1	1	1	1	1	1	1	1	1
98-99%	0	0	0	0	0	0	0	0	0	0	0	0
97-98%	0	0	0	0	0	0	0	0	0	0	0	0
96-97%	0	0	0	0	0	0	0	0	0	0	0	0
95-96%	0	0	0	0	0	0	0	0	0	0	0	0
94-95%	0	0	0	0	0	0	0	0	0	0	0	0
93-94%	0	0	0	0	0	0	0	0	0	0	0	0
92-93%	0	0	0	0	0	0	0	0	0	0	0	0
91-92%	0	0	0	0	0	0	0	0	0	0	0	0
90-91%	0	0	0	0	0	0	0	0	0	0	0	0
Below 90%	0	0	0	0	0	0	0	0	0	0	0	0

Clad Thickness	**t=0 (con.)**											
SCRAM State	SCRAM failure, trip success						SCRAM and trip failure					
RPS Pump	**Operational**			**Not Operational**			**Operational**			**Not Operational**		
DRACS	Enh.	Nom.	Deg.	Enh.	Nom.	Deg.	Enh.	Nom.	Deg.	Enh.	Nom.	Deg.
99-100%	1	1	1	1	1	1	1	1	1	1	1	1
98-99%	0	0	0	0	0	0	0	0	0	0	0	0
97-98%	0	0	0	0	0	0	0	0	0	0	0	0
96-97%	0	0	0	0	0	0	0	0	0	0	0	0
95-96%	0	0	0	0	0	0	0	0	0	0	0	0
94-95%	0	0	0	0	0	0	0	0	0	0	0	0
93-94%	0	0	0	0	0	0	0	0	0	0	0	0
92-93%	0	0	0	0	0	0	0	0	0	0	0	0
91-92%	0	0	0	0	0	0	0	0	0	0	0	0
90-91%	0	0	0	0	0	0	0	0	0	0	0	0
Below 90%	0	0	0	0	0	0	0	0	0	0	0	0

Failure: Clad Melting

Clad Thickness	99-100%	98-99%	97-98%	96-97%	95-96%	94-95%	93-94%	92-93%	91-92%	90-91%	Below 90%
True	0	0	0	0	0	0	0	0	0	0	1
False	1	1	1	1	1	1	1	1	1	1	0

System Diagnostics

Failure: Clad Melting	**True**		**False**	
Failure: Fuel Relocation	True	False	True	False
Healthy	0	0	0	1
Not Healthy	1	1	1	0

Appendix D: Case Study Results

This appendix presents the full results from the case study comparison presented in Chapter 6.

D.1 Results from the Accuracy Alignment Comparison

Measured health assessments from the time-based discretization models

	Time-Step Length (s)			
	1200	120	60	9
Model Risk Assessment	2.68E-07	2.59E-07	2.57E-07	2.73E-07

Measured health assessments from the state-based discretization models

	Net Reactivity Threshold (%)			
	-0.1	0	0.02	0.2
Model Risk Assessment	1.50E-07	8.00E-08	5.16E-08	1.26E-08

Measured health assessments from the hybrid time-based discretization models

			Primary Time-Step Length			
		Threshold Value	**1200**	**120**	**60**	**9**
Secondary Time-Step Length (s)	**1200**	**0.2**		2.47E-07	2.45E-07	2.60E-07
		0.02		2.47E-07	2.45E-07	2.54E-07
		0		2.47E-07	2.45E-07	2.55E-07
		-0.1		2.67E-07	2.65E-07	2.86E-07
	120	**0.2**	9.46E-08		2.46E-07	2.63E-07
		0.02	8.65E-08		2.46E-07	2.62E-07
		0	8.65E-08		2.46E-07	2.62E-07
		-0.1	9.46E-08		2.55E-07	2.84-07
	60	**0.2**	5.67E-08	1.85E-07		2.63E-07
		0.02	4.96E-08	1.83E-07		2.60E-07
		0	4.96E-08	1.85E-07		2.62E-07
		-0.1	5.67E-08	1.85E-07		2.80E-07
	9	**0.2**	2.06E-07	2.14E-07	2.45E-07	
		0.02	8.61E-10	1.36E-07	1.03E-07	
		0	8.77E-10	1.29E-07	2.06E-07	
		-0.1	9.5E-10	1.46E-07	2.06E-07	

D.2 Results from the CPT Construction Time

Measured CPT construction time from the time-based discretization models

	Time-Step Length (s)			
	1200	**120**	**60**	**9**
Non-Observable Parameters	383.3	3,420.4	6,974.9	73,604.8
Observable Parameters	2,035.6	19,590.1	39,060.4	225,895.5
Fail: Fuel Relocation	58.7	58.7	58.7	58.7
Dynamic Clad Thickness	1.0	10.2	18.2	149.8
Total Computational Time	**2,478.6**	**23,032.9**	**46,053.5**	**299,708.8**

Measured CPT construction time from the state-based discretization models

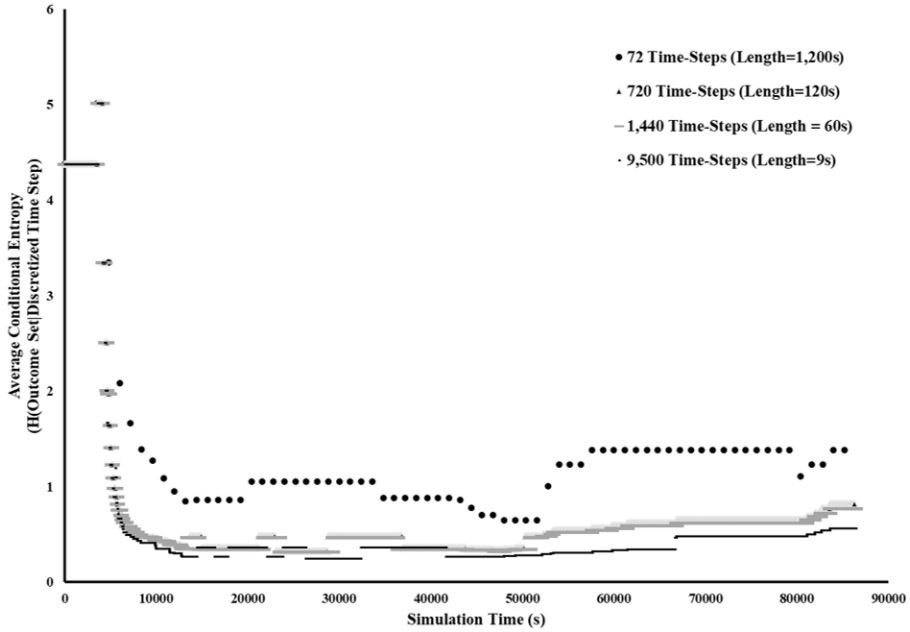
	Net Reactivity Threshold (%)			
	-0.1	**0**	**0.02**	**0.2**
Non-Observable Parameters	1,934.4	838.2	326.1	16.1
Observable Parameters	32,423.9	9,958.6	450.2	2.2
Fail: Fuel Relocation	65.5	15.1	15.1	37.8
Dynamic Clad Thickness	891.0	305.3	21.8	2.2
Total Computational Time	**36,121.1**	**12,033.3**	**898.7**	**22.3**

Measured CPT construction time from the hybrid time-based discretization models

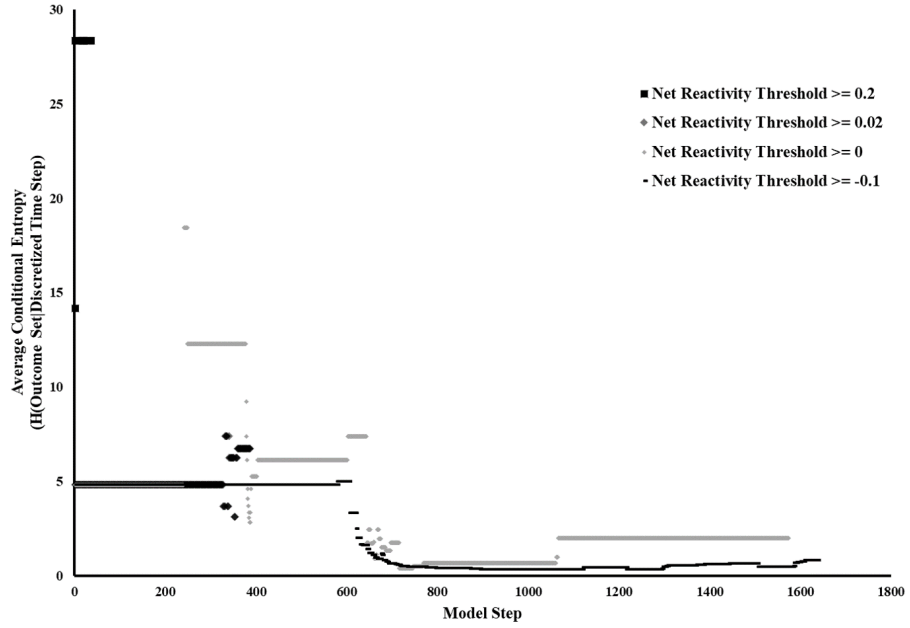
				Primary Time-Step Length (s)			
		Threshold Value (%)	Computational Time for...	**1200**	**120**	**60**	**9**
Secondary Time-Step Length (s)	**1200**	**0.2**	Non-Observable Parameters		1,050.9	1,782.7	9,279.6
			Observable Parameters		2,167.9	2,355.1	4,197.6
			Fail: Fuel Relocation		15.6	16.4	15.8
			Dynamic Clad Thickness		310.7	637.7	4,152.1
			Total Computational Time		**3,545.1**	**74,791.9**	**17,645.1**
		0.02	Non-Observable Parameters		1,223.0	1,827.5	8,874.2
			Observable Parameters		2,229.9	2,353.0	4,227.9
			Fail: Fuel Relocation		17.1	15.4	14.7
			Dynamic Clad Thickness		338.4	635.2	4,030.2
			Total Computational Time		**3,808.4**	**4,831.1**	**17,147.0**
		0	Non-Observable Parameters		1,235.0	1,806.5	8,932.2
			Observable Parameters		2,173.4	2,350.0	4,277.1
			Fail: Fuel Relocation		14.6	14.9	15.0
			Dynamic Clad Thickness		312.4	624.7	3,955.2
			Total Computational Time		**3,735.4**	**4,796.1**	**17,179.5**
		-0.1	Non-Observable Parameters		4,379.6	8,909.9	56,200.3
			Observable Parameters		14,114.1	27,354.2	164,494.20
			Fail: Fuel Relocation		15.2	15.9	15.6
			Dynamic Clad Thickness		311.7	593.1	3,986.2
			Total Computational Time		**18,820.6**	**36,783.1**	**229,696.3**
	120	**0.2**	Non-Observable Parameters	617.5		5,779.1	12,525.2
			Observable Parameters	2,116.1		20,095.1	21,786.5
			Fail: Fuel Relocation	16.0		15.7	15.0
			Dynamic Clad C0C0C0	30.4		624.0	3,969.0
			Total Computational Time	**2,780.0**		**26,513.9**	**38,295.7**
		0.02	Non-Observable Parameters	4,068.6		5,785.4	12,752.2
			Observable Parameters	2,610.3		19,788.3	21,546.4
			Fail: Fuel Relocation	16.4		14.7	15.5
			Dynamic Clad Thickness	31.9		614.3	3,920.8
			Total Computational Time	**6,727.2**		**26,202.7**	**38,234.9**
		0	Non-Observable Parameters	4,012.0		5,856.1	12,946.7
			Observable Parameters	8,734.6		20,312.3	22,185.5
			Fail: Fuel Relocation	15.5		15.8	16.6
			Dynamic Clad Thickness	29.6		613.1	4,367.0
			Total Computational Time	**12,791.7**		**26,797.3**	**39,515.8**
		-0.1	Non-Observable Parameters	4,597.5		10,303.3	57,294.1
			Observable Parameters	19,489.7		33,480.1	175,244.4
			Fail: Fuel Relocation	16.0		14.8	63.3
			Dynamic Clad Thickness	31.2		622.2	3,962.2
			Total Computational Time	**24,134.4**		**44,420.4**	**236,564.0**
	60	**0.2**	Non-Observable Parameters	578.3	5,629.7		17,001.1
			Observable Parameters	2,007.4	20,141.6		40,945.9
			Fail: Fuel Relocation	14.6	65		16.1
			Dynamic Clad Thickness	31.8	323.1		4,030.3
			Total Computational Time	**2,632.1**	**26,159.5**		**61,993.4**
		0.02	Non-Observable Parameters	8,327.7	9,520.7		17,184.4
			Observable Parameters	2,511.6	36,389.1		41,560.9
			Fail: Fuel Relocation	15.5	64.2		14.9
			Dynamic Clad Thickness	30.1	304.3		3,929.4
			Total Computational Time	**10,884.9**	**46,278.3**		**62,689.6**
		0	Non-Observable Parameters	8,878.1	9,414.4		17,356.3
			Observable Parameters	16,023.2	38,898.7		41,345.3
			Fail: Fuel Relocation	15.5	16.3		15.0
			Dynamic Clad Thickness	30.6	297.6		3,972.8
			Total Computational Time	**24,947.4**	**48,627.0**		**62,689.4**
		-0.1	Non-Observable Parameters	9,024.3	9,566.4		59,513.1
			Observable Parameters	38,768.9	39,566.5		186,103.6
			Fail: Fuel Relocation	66.0	15.8		15.5
			Dynamic Clad Thickness	30.1	309.7		3,946.2
			Total Computational Time	**47,889.3**	**49,458.4**		**249,578.4**
	9	**0.2**	Non-Observable Parameters	671.7	6,367.6	12,470.6	
			Observable Parameters	2,008.6	20,242.3	40,033.3	
			Fail: Fuel Relocation	14.0	16.1	15.5	
			Dynamic Clad Thickness	30.7	304.4	605.6	
			Total Computational Time	**2,725.0**	**26,930.4**	**53,125.0**	
		0.02	Non-Observable Parameters	5,720.8	59,366.0	58,600.3	
			Observable Parameters	5,244.3	235,737.2	222,087.5	
			Fail: Fuel Relocation	15.0	13.5	16.0	
			Dynamic Clad Thickness	32.1	306.6	600.4	
			Total Computational Time	**62,493.2**	**295,423.3**	**281,304.2**	
		0	Non-Observable Parameters	57,254.8	52,928.1	59,636.0	
			Observable Parameters	98,377.8	251,224.8	251,689.8	
			Fail: Fuel Relocation	66.4	12.3	63.6	
			Dynamic Clad Thickness	30.9	310.3	628.3	
			Total Computational Time	**155,729.9**	**304,475.5**	**312,017.7**	
		-0.1	Non-Observable Parameters	58,796.7	58,197.5	58,851.6	
			Observable Parameters	253,911.8	254,894.3	258,539.9	
			Fail: Fuel Relocation	64.0	15.2	67.3	
			Dynamic Clad Thickness	30.2	305.6	610.5	
			Total Computational Time	**312,802.7**	**313,412.6**	**318,069.3**	

D.3 Results from the Information Content

Conditional entropy chart for the time-based discretization models

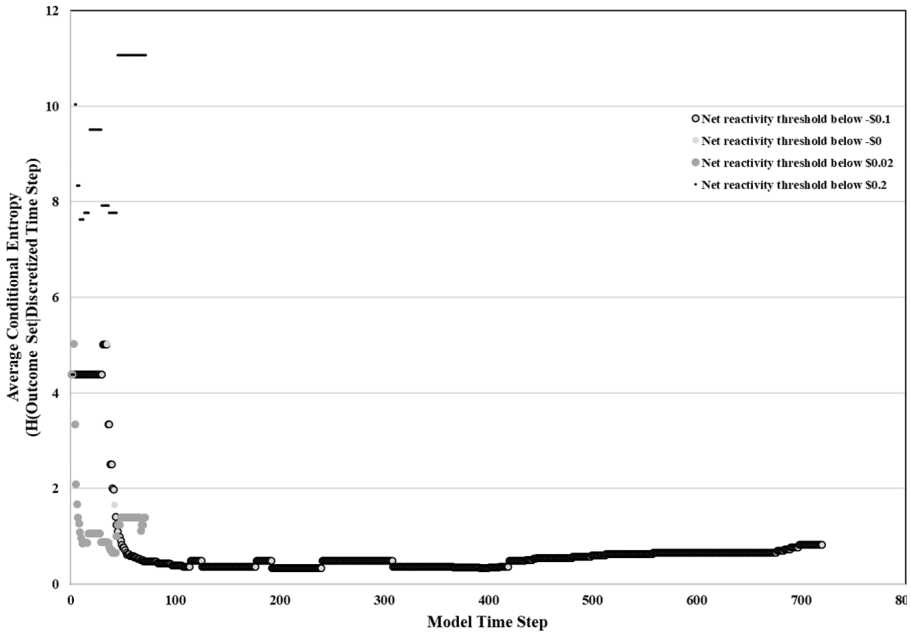


Conditional entropy chart for the state-based discretization models

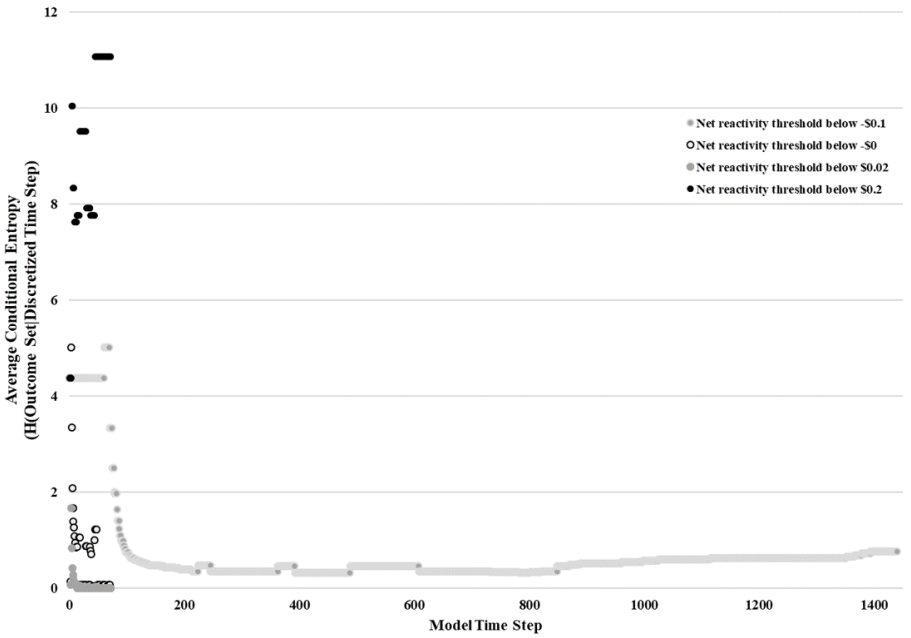


Conditional entropy charts for the hybrid time-based discretization models with

primary measurement rate of 120s and a secondary measurement rate of 1200s

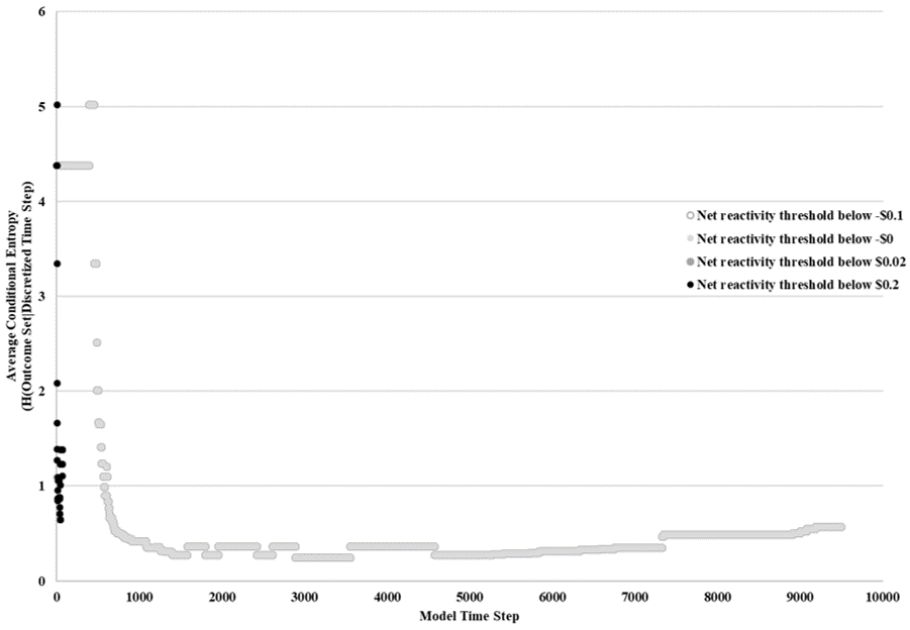


Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 60s and a secondary measurement rate of 1200s

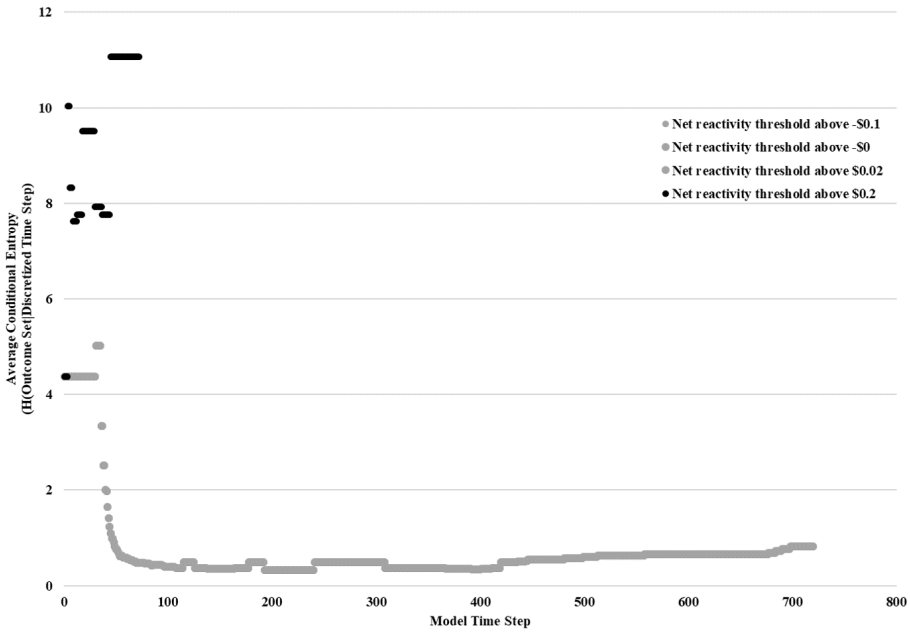


Conditional entropy charts for the hybrid time-based discretization models with

primary measurement rate of 9s and a secondary measurement rate of 1200s

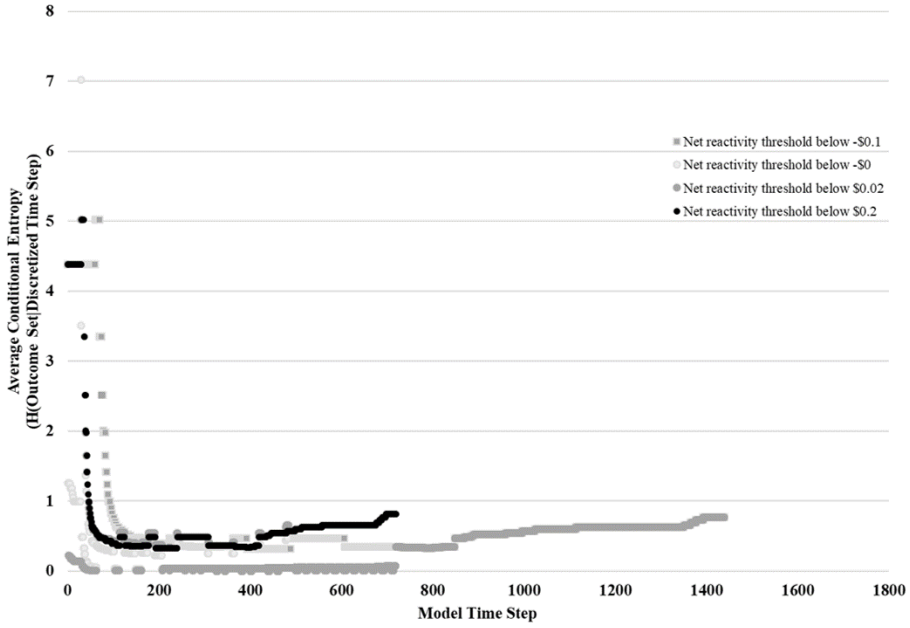


Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 1200s and a secondary measurement rate of 120s

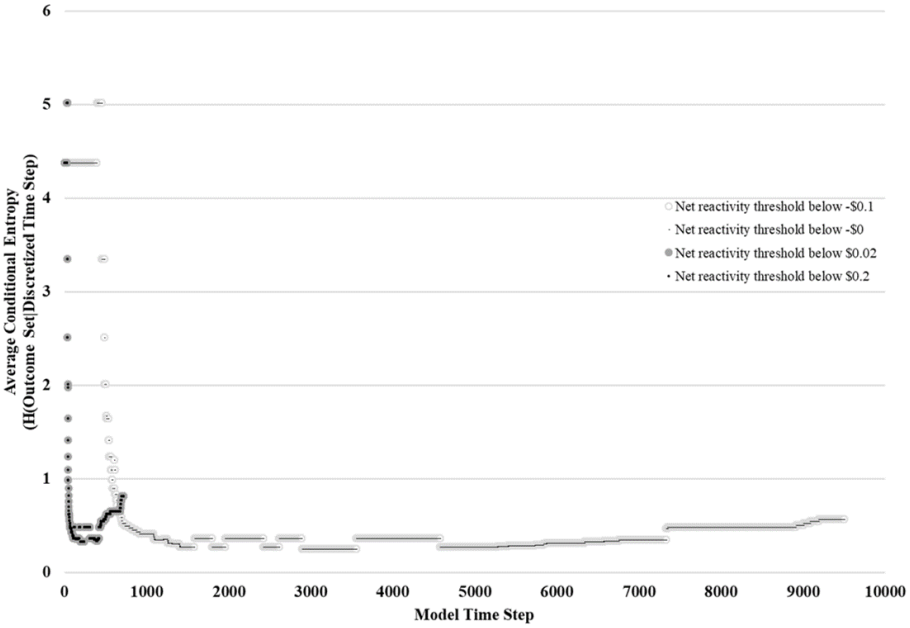


Conditional entropy charts for the hybrid time-based discretization models with

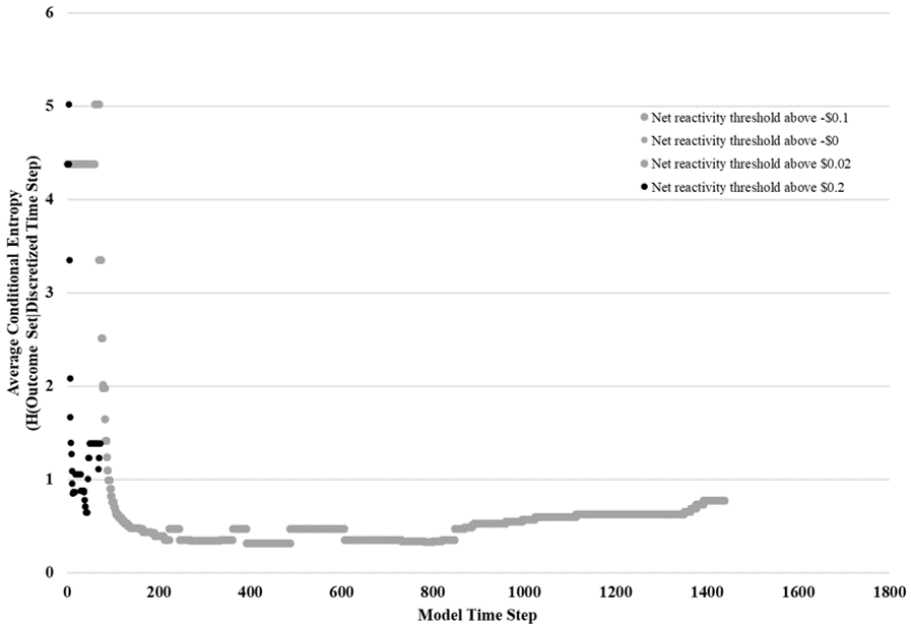
primary measurement rate of 60s and a secondary measurement rate of 120s



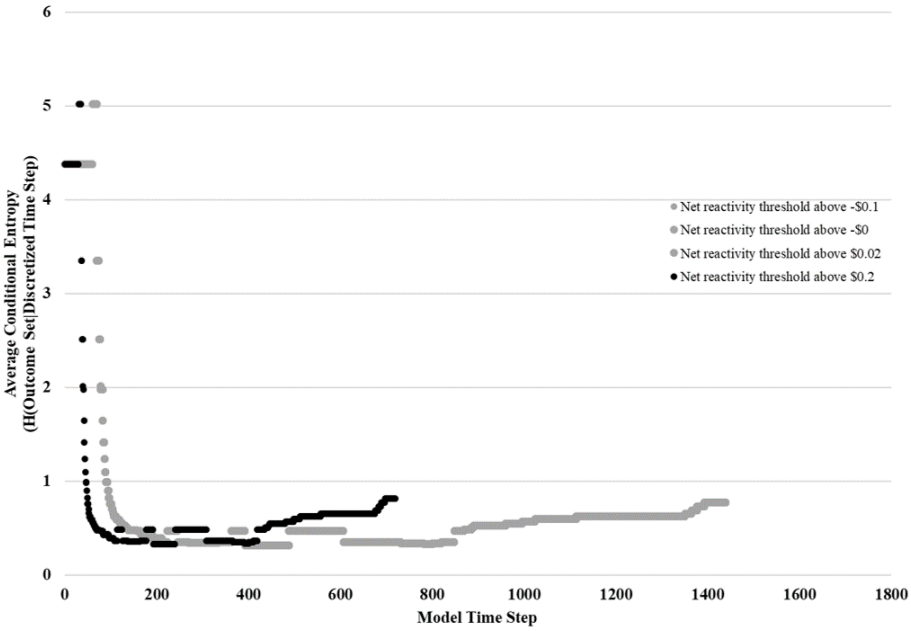
Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 9s and a secondary measurement rate of 120s



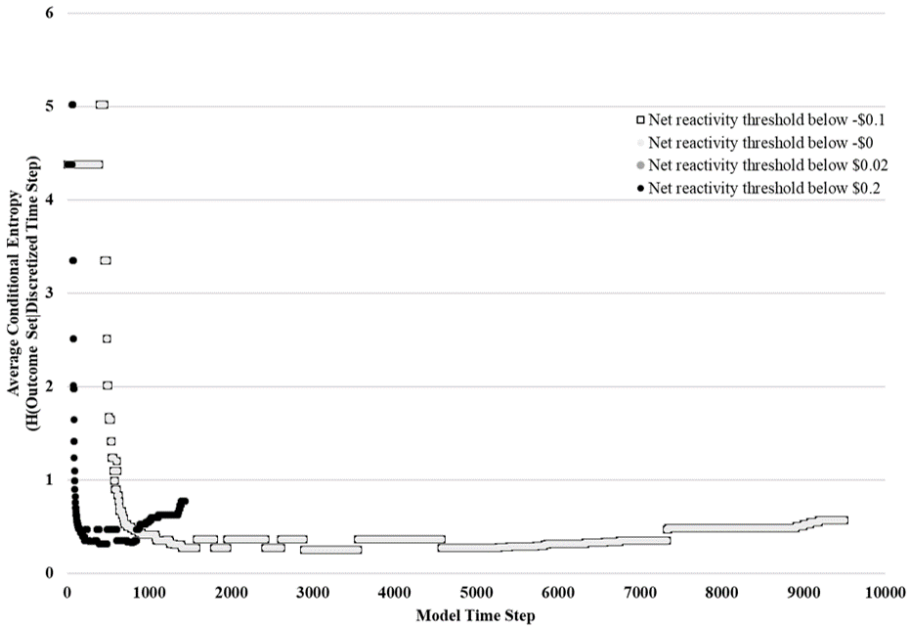
Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 1200s and a secondary measurement rate of 60s



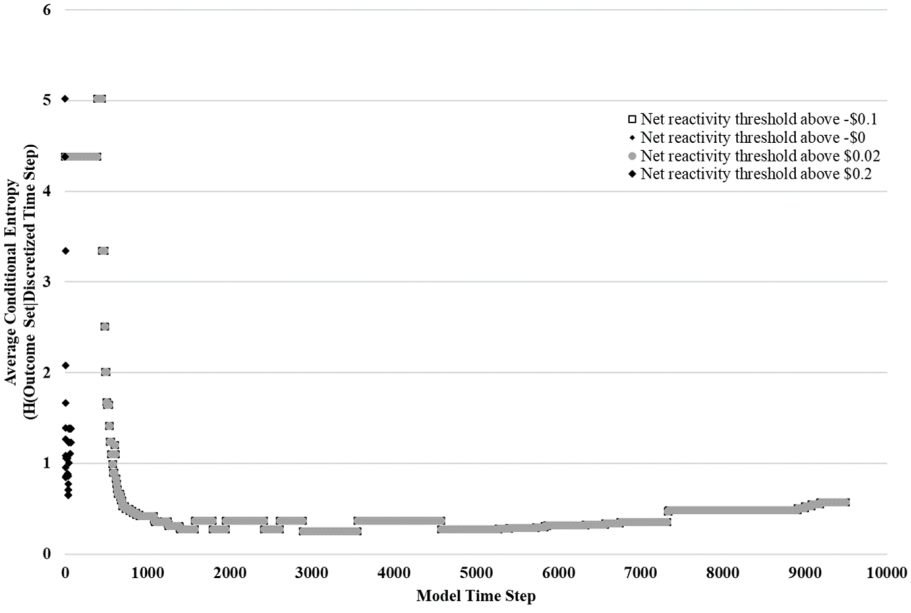
Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 120s and a secondary measurement rate of 60s



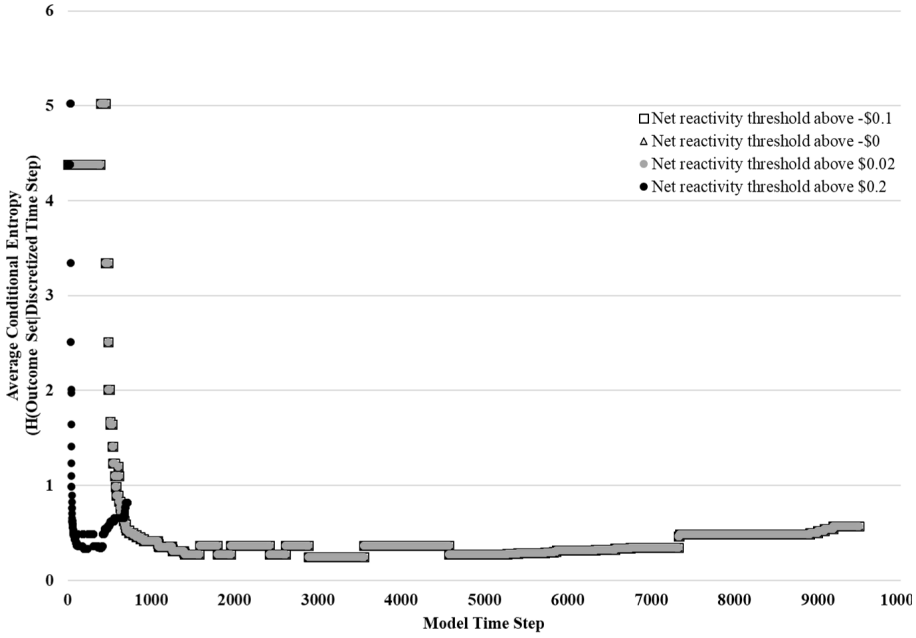
Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 9s and a secondary measurement rate of 60s



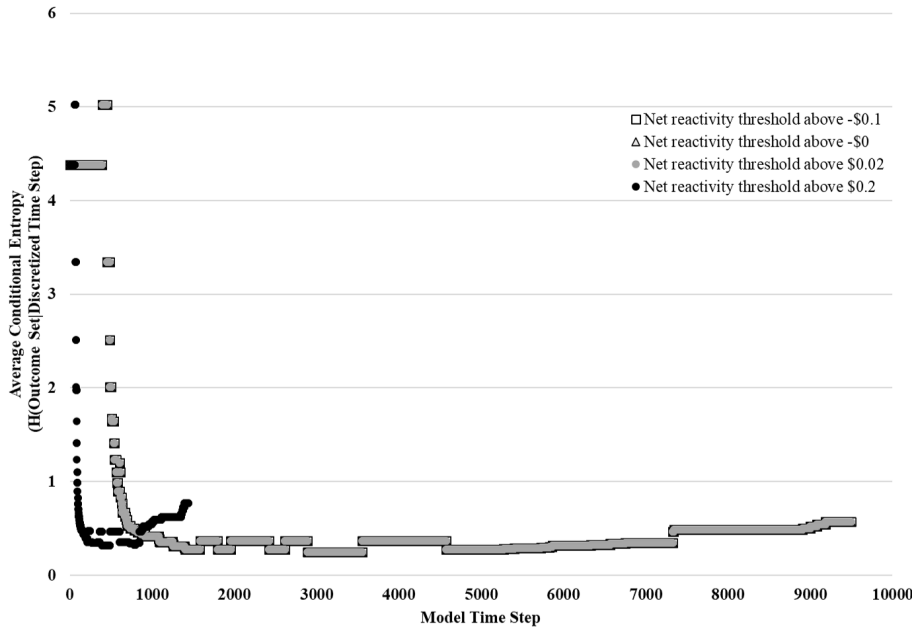
Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 1200s and a secondary measurement rate of 9s



Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 120s and a secondary measurement rate of 9s



Conditional entropy charts for the hybrid time-based discretization models with primary measurement rate of 60s and a secondary measurement rate of 9s



Bibliography

[1] Martin Mayfield, Giuliano Punzo, Richard Beasley, Ginny Clarke, Nic Holt, and Stuart Jobbins. "Challenges of complexity and resilience in complex engineering systems". *ENCORE Network+ White Paper* (2018).

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

[3] Richard Laudenat. "Opening Remarks". *Proceedings from the ASME-SERAD and UMD-CRR Joint Interactive Seminar and Pre-Workshop on Intersection of PRA and PHM*. Oct. 2, 2020. URL: `http://hdl.handle.net/1903/26664`.

[4] Ahmed K Noor. "The world is more than complicated". *Mechanical Engineering* 133.11 (2011), pp. 30–35.

[5] Vicki M Bier. "An overview of probabilistic risk analysis for complex engineered systems". *Fundamentals of risk analysis and risk management*. Ed. by Vlasta Molak. Lewis Publishers, 1997.

[6] Charles R Farrar and Keith Worden. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[7] Judea Pearl. "The seven tools of causal inference, with reflections on machine learning". *Communications of the ACM* 62.3 (2019), pp. 54–60.

[8] Michael Pecht. "Prognostics and health management of electronics". *Encyclopedia of structural health monitoring* (2009).

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

[10] Katrina Groth, Matthew Denman, Michael Darling, Thomas Jones, and George Luger. "Building and using dynamic risk-informed diagnosis procedures for complex system accidents". *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 3.1 (2020), pp. 193–207.

[11] Takehisa Kohda and Weimin Cui. "Risk-based reconfiguration of safety monitoring system using dynamic Bayesian network". *Reliability Engineering & System Safety* 92.12 (2007), pp. 1716–1723.

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

[13] Sandia National Laboratories. *Transient Overpower Data for Sodium Fast Reactor*. 2019.

[14] Austin Lewis and Katrina Groth. "Comparison of performance of DBN models for SIPPRA-based health monitoring based on different data stream discretization methods" (). In progress.

[15] Austin Lewis and Katrina Groth. "A review of methods for discretizing continuous-time accident sequences". *Proceedings of the 29th European Safety and Reliability Conference* (2019), pp. 754–761.

[16] Austin Lewis and Katrina Groth. "A multi-interval method for discretizing continuous-time event sequences". *2021 Annual Reliability and Maintainability Symposium* (2021), pp. 1–7.

[17] Austin D. Lewis and Katrina M. Groth. "A dynamic Bayesian network structure for joint diagnostics and prognostics of complex engineering systems". *Algorithms* 13 (Mar. 2020). Special Issue Bayesian Networks: Inference Algorithms, Applications, and Software Tools, pp. 64+. DOI: 10.3390/a13030064. Invited paper & cover article.

[18] Austin Lewis and Katrina Groth. "Impact of complex engineering system data stream discretization techniques on the performance of Dynamic Bayesian Network-Based Health Assessments" (). In progress.

[19] Diego Mandelli, Andrea Alfonsi, Congjian Wang, Zhegang Ma, Carlo Parisi, Tunc Aldemir, Curtis Smith, and Robert Youngblood. "Mutual Integration of Classical and Dynamic PRA". *Nuclear Technology* (2020), pp. 1–13.

[20] Gregory W Vogl, Brian A Weiss, and Moneer Helu. "A review of diagnostic and prognostic capabilities and best practices for manufacturing". *Journal of Intelligent Manufacturing* 30.1 (2019), pp. 79–95.

[21] Andrew Rae, Rob Alexander, and John McDermid. "Fixing the cracks in the crystal ball: A maturity model for quantitative risk assessment". *Reliability Engineering & System Safety* 125 (2014), pp. 67–81.

[22] Ali Mosleh. "PRA: a perspective on strengths, current limitations, and possible improvements". *Nuclear Engineering and Technology* 46.1 (2014), pp. 1–10.

[23] Ranganath Kothamasu, Samuel H Huang, and William H VerDuin. "System health monitoring and prognostics—a review of current paradigms and practices". *The International Journal of Advanced Manufacturing Technology* 28.9-10 (2006), pp. 1012–1024.

[24] Abhinav Saxena, Jose Celaya, Bhaskar Saha, Sankalita Saha, and Kai Goebel. "Metrics for offline evaluation of prognostic performance". *International Journal of Prognostics and health management* 1.1 (2010), pp. 4–23.

[25] Sreerupa Das, Richard Hall, Stefan Herzog, Gregory Harrison, Michael Bodkin, and Lockheed Martin. "Essential steps in prognostic health management". *2011 IEEE Conference on Prognostics and Health Management*. IEEE. 2011, pp. 1–9.

[26] Jian Guo, Zhaojun Li, and Meiyan Li. "A Review on Prognostics Methods for Engineering Systems". *IEEE Transactions on Reliability* (2019).

[27] Andrew KS Jardine, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance". *Mechanical Systems and Signal Processing* 20.7 (2006), pp. 1483–1510.

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

[29] Michele Compare, Piero Baraldi, and Enrico Zio. "Challenges to IoT-enabled predictive maintenance for industry 4.0". *IEEE Internet of Things Journal* 7.5 (2019), pp. 4585–4597.

[30] Philippe Weber and Lionel Jouffe. "Complex system reliability modelling with dynamic object oriented Bayesian networks (DOOBN)". *Reliability Engineering & System Safety* 91.2 (2006), pp. 149–162.

[31] Antonio J Guillén, Juan F Gómez, Adolfo Crespo, Alejandro Guerrerro, Antonio Sola, and Luis Barbera. "Advances in PHM application frameworks: Processing methods, prognosis models, decision making". *Chemical Engineering* 33 (2013), pp. 391–396.

[32] Rui Li, Wim JC Verhagen, and Richard Curran. "A systematic methodology for Prognostic and Health Management system architecture definition". *Reliability Engineering & System Safety* 193 (2020), p. 106598.

[33] Enrico Zio and Francesco Di Maio. "A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system". *Reliability Engineering & System Safety* 95.1 (2010), pp. 49–57.

[34] Alexandre Muller, Marie-Christine Suhner, and Benoît Iung. "Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system". *Reliability Engineering & System Safety* 93.2 (2008), pp. 234–253.

[35] Enrico Zio. "Some challenges and opportunities in reliability engineering". *IEEE Transactions on Reliability* 65.4 (2016), pp. 1769–1782.

[36] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. "Industry 4.0". *Business & information systems engineering* 6.4 (2014), pp. 239–242.

[37] Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. "Digital twin in industry: State-of-the-art". *IEEE Transactions on Industrial Informatics* 15.4 (2018), pp. 2405–2415.

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

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

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

[41] 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". *IEEE Systems Journal* 9.4 (2014), pp. 1197–1207.

[42] Norman Fenton and Martin Neil. *Risk assessment and decision analysis with Bayesian networks*. Crc Press, 2019.

[43] Zhengdao Zhang and Feilong Dong. "Fault detection and diagnosis for missing data systems with a three time-slice dynamic Bayesian network approach". *Chemometrics and Intelligent Laboratory Systems* 138 (2014), pp. 30–40.

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

[45] Nima Khakzad, Genserik Reniers, Rouzbeh Abbassi, and Faisal Khan. "Vulnerability analysis of process plants subject to domino effects". *Reliability Engineering & System Safety* 154 (2016), pp. 127–136.

[46] MA Djeziri, S Benmoussa, and M EH Benbouzid. "Data-driven approach augmented in simulation for robust fault prognosis". *Engineering Applications of Artificial Intelligence* 86 (2019), pp. 154–164.

[47] Ahmed Mosallam, Kamal Medjaher, and Noureddine Zerhouni. "Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction". *Journal of Intelligent Manufacturing* 27.5 (2016), pp. 1037–1048.

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

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

[50] Philippe Weber and Lionel Jouffe. "Reliability modelling with dynamic bayesian networks". *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*. 2003.

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

[52] Xianguo Wu, Huitao Liu, Limao Zhang, Miroslaw J Skibniewski, Qianli Deng, and Jiaying Teng. "A dynamic Bayesian network based approach to safety decision support in tunnel construction". *Reliability Engineering & System Safety* 134 (2015), pp. 157–168.

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

[54] Kamal Medjaher, Jean-Yves Moya, and Noureddine Zerhouni. "Failure prognostic by using dynamic Bayesian Networks." *2nd IFAC Workshop on Dependable Control of Discrete Systems*. 2009.

[55] Yunfei Zhao, Jiejuan Tong, Liguo Zhang, and Qin Zhang. "Pilot study of dynamic Bayesian networks approach for fault diagnostics and accident progression prediction in HTR-PM". *Nuclear Engineering and Design* 291 (2015), pp. 154–162.

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

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

[58] Michael C Darling, George F Luger, Thomas B Jones, Matthew R Denman, and Katrina M Groth. "Intelligent modeling for nuclear power plant accident management". *International Journal on Artificial Intelligence Tools* 27.02 (2018), p. 1850003.

[59] Thomas Dean and Keiji Kanazawa. "A model for reasoning about persistence and causation". *Computational intelligence* 5.2 (1989), pp. 142–150.

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

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

[62] Ruihua Jiao, Kaixiang Peng, Jie Dong, and Chuanfang Zhang. "Fault monitoring and remaining useful life prediction framework for multiple fault modes in prognostics". *Reliability Engineering & System Safety* 203 (2020), p. 107028.

[63] Enrico Zio. "Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice". *Reliability Engineering & System Safety* 218 (2022), p. 108119.

[64] Austin Lewis and Katrina Groth. "Metrics for evaluating the performance of complex engineering system health monitoring models" (). Accepted in Reliability Engineering & System Safety.

[65] Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. "Fusing physics-based and deep learning models for prognostics". *Reliability Engineering & System Safety* 217 (2022), p. 107961.

[66] Manuel A Vega, Zhen Hu, Travis B Fillmore, Matthew D Smith, and Michael D Todd. "A Novel Framework for Integration of Abstracted Inspection Data and Structural Health Monitoring for Damage Prognosis of Miter Gates". *Reliability Engineering & System Safety* 211 (2021), p. 107561.

[67] Taotao Zhou, Mohammad Modarres, and Enrique López Droguett. "Multi-unit risk aggregation with consideration of uncertainty and bias in risk metrics". *Reliability Engineering & System Safety* 188 (2019), pp. 473–482.

[68] Inger Lise Johansen and Marvin Rausand. "Foundations and choice of risk metrics". *Safety science* 62 (2014), pp. 386–399.

[69] Jeffrey W Herrmann. *Engineering decision making and risk management*. John Wiley & Sons, 2015.

[70] Liang Tang, Marcos E Orchard, Kai Goebel, and George Vachtsevanos. "Novel metrics and methodologies for the verification and validation of prognostic algorithms". *2011 Aerospace Conference*. IEEE. 2011, pp. 1–8.

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

[72] Zhiguo Zeng, Francesco Di Maio, Enrico Zio, and Rui Kang. "A hierarchical decision-making framework for the assessment of the prediction capability of prognostic methods". *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 231.1 (2017), pp. 36–52.

[73] Victor R Basili1 Gianluigi Caldiera and H Dieter Rombach. "The goal question metric approach". *Encyclopedia of software engineering* (1994), pp. 528–532.

[74] Mark Davidson. "The 28 metrics every plant needs to monitor: SMART: specific, measurable, actionable, realistic, and time-based". *Plant Engineering* 67.9 (2013), pp. 10–13.

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

[76] M. Drouin, A. Gilberston, G. Parry, J. Lehner, G. Martinez-Guiridi, J. LaChance, and T. Wheeler. *Guidance on the treatment of uncertainties associated with PRAs in risk-informed decisionmaking*. Technical Report NUREG-1855. U.S. Nuclear Regulatory Commission, Mar. 2017.

[77] Pentti Tarvainen. "Adaptability evaluation of software architectures; a case study". *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*. Vol. 2. IEEE. 2007, pp. 579–586.

[78] Zebin Yang, Aijun Zhang, and Agus Sudjianto. "Enhancing explainability of neural networks through architecture constraints". *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[79] Michael Pecht and Myeongsu Kang. *Prognostics and health management of electronics - fundamentals, machine learning, and the internet of things*. John Wiley & Sons, 2018.

[80] Carl Malings and Matteo Pozzi. "Value-of-information in spatio-temporal systems: Sensor placement and scheduling". *Reliability Engineering & System Safety* 172 (2018), pp. 45–57.

[81] Sunderrajan Krishnan, Alexandre Boucher, and Andre G Journel. "Evaluating information redundancy through the tau model". *Geostatistics*. Springer, 2005, pp. 1037–1046.

[82] Neta Aizenbud-Reshef, Brian T Nolan, Julia Rubin, and Yael Shaham-Gafni. "Model traceability". *IBM Systems Journal* 45.3 (2006), pp. 515–526.

[83] MARCO ENRICO Ricotti and Enrico Zio. "Neural network approach to sensitivity and uncertainty analysis". *Reliability Engineering & System Safety* 64.1 (1999), pp. 59–71.

[84] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. "Manipulating and measuring model interpretability". *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–52.

[85] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks". *Digital Signal Processing* 73 (2018), pp. 1–15.

[86] Chunsheng Yang, Yanni Zou, Jie Liu, and Kyle R Mulligan. "Predictive model evaluation for PHM". *International Journal of Prognostics and Health Management* 5.2 (2014).

[87] Shunfeng Cheng, Michael Azarian, and Michael Pecht. "Sensor system selection for prognostics and health monitoring". *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 43277. Citeseer. 2008, pp. 1383–1389.

[88] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. "Segmenting time series: A survey and novel approach". *Data mining in time series databases*. World Scientific, 2004, pp. 1–21.

[89] Gregory F Cooper, Eric J Horvitz, and David E Heckerman. *A method for temporal probabilistic reasoning*. Working Paper 88-30. Stanford, California: Knowledge Systems Laboratory, 1988.

[90] Ying Yang and Geoffrey I Webb. "A comparative study of discretization methods for naive-Bayes classifiers". *Proceedings of PKAW*. Vol. 2002. 2002.

[91] Jianxiong Zhou, Shanbi Wei, and Yi Chai. "Using improved dynamic Bayesian networks in reliability evaluation for flexible test system of aerospace pyromechanical device products". *Reliability Engineering & System Safety* 210 (2021), p. 107508.

[92] Elizabeth Bismut and Daniel Straub. "Optimal adaptive inspection and maintenance planning for deteriorating structural systems". *Reliability Engineering & System Safety* 215 (2021), pp. 1–18.

[93] Zachary K Jankovsky, Matthew R Denman, and Tunc Aldemir. "Dynamic event tree analysis with the SAS4A/SASSYS-1 safety analysis code". *Annals of Nuclear Energy* 115 (2018), pp. 55–72.

[94] *GeNIe*. Version 3.0.R2. 2019. URL: `https://bayesfusion.com/genie/`.

[95] *Python*. Version 3.7. 2019. URL: `https://python.org/`.

[96] Austin Lewis and Katrina Groth. "Impact of different time discretization methods on dynamic Bayesian network-based dynamic probabilistic safety assessments". *Proceedings of the 2021 International Topical Meeting on Probabilistic Safety Assessment and Analysis* (2021), pp. 410–419.

[97] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. "Pyro: Deep universal probabilistic programming". *The Journal of Machine Learning Research* 20.1 (2019), pp. 973–978.

[98] Curtis Smith. "A PRA Perspective on Prognostics and Health Management". *Proceedings from the ASME-SERAD and UMD-CRR Joint Interactive Seminar and Pre-Workshop on Intersection of PRA and PHM*. Oct. 2, 2020. URL: `http://hdl.handle.net/1903/26664`.

[99] Enrique Lopez Droguett. "The PHM Perspective: A Review on Prognostics and Health Management". *Proceedings from the ASME-SERAD and UMD-CRR Joint Interactive Seminar and Pre-Workshop on Intersection of PRA and PHM*. Oct. 2, 2020. URL: `http://hdl.handle.net/1903/26664`.

[100] General Electric. *PRISM Preliminary safety information document*. Tech. rep. GEFR-00793. NRC Accession Number: ML082880422. San Jose, CA, Dec. 1987.