INL/RPT-23-74530

Light Water Reactor Sustainability Program

Development of Analysis Methods that Integrate Numeric and Textual Equipment Reliability Data

September 2023

U.S. Department of Energy

Office of Nuclear Energy

INL/RPT-23-74530

Development of Analysis Methods that Integrate Numeric and Textual Equipment Reliability Data

D. Mandelli. C. Wang, J. Cogliati, V. Agarwal

September 2023

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Light Water Reactor Sustainability Program

SUMMARY

Within the Light Water Reactor Sustainability program, the Risk-Informed Systems Analysis (RISA) Pathway is performing collaborative research on the development and deployment of technologies designed to assist operating nuclear power plants (NPPs) to reduce operating costs and improve plant reliability and availability. One of the Risk-Informed Systems Analysis research areas is focusing on the development of methods and tools designed to optimize plant operations (e.g., maintenance and replacement schedules, optimal maintenance postures for plant structures, systems, and components) in a manner that is more cost effective than current approaches and makes better use of available structures, systems, and components health data. The Risk-Informed Asset Management project targets this research area by creating a direct bridge between component equipment reliability (ER) data and system engineer decision making regarding maintenance activity scheduling and component aging management.

In this respect, one challenge that NPP system engineers are facing is that the amount of ER data being continuously generated is not only extremely large but in different forms: textual (e.g., condition or maintenance reports) and numeric (e.g., generated by monitoring systems). All these data elements provide them with valuable insights and information regarding the discovery of anomalous behaviors or degradation trends, the identification of the possible causes behind such behaviors and trends, and the prediction of their direct consequences. However, several challenges have proved to be roadblocks to this process. While some of these challenges are technical in nature (i.e., data are often distributed over several physical servers or databases), others are conceptual in nature: data elements come in different formats (e.g., numeric or textual) and measured values have different scales (e.g., vibration spectra and oil temperature).

The activities performed by the Risk-Informed Asset Management project during Fiscal Year 2023 directly tackles the need to simultaneously integrate the analysis of ER data in all its forms, numeric and textual. Note that such a task has never been performed before due to the complexity of the systems under consideration but, most importantly, because of the technical challenges behind the harmonization of ER data formats and the lack of adequate computational methods to analyze them. Our approach borrows ideas and concepts from the medical field where the integration of several data sources is vital to assist medical practitioners to perform correct diagnosis and indicate optimal treatments. In our view an NPP asset or system is equivalent to a patient in a medical context. The main difference is that the complexity of a human body is a magnitude more complex when compared to typical assets or systems commonly present in NPPs

(e.g., centrifugal pumps, or motor-operated valves). This simplifies our first requirement when analyzing heterogenous ER data formats: to put data into "context". Context is here intended as the additional piece of information needed by ER data analysis tools to understand what these data elements are referring to, that is which kind of knowledge they are generating. In our context, this knowledge can be translated into models that capture the form and functional architecture of assets and systems, their dependencies, and how they interact. These models actually emulate the knowledge that NPP system engineers possess about assets and systems; this is their key of success when analyzing ER data, their challenge is the ability to handle a large amount of data.

Here, we employ model-based system engineering (MBSE) models of systems and assets to represent and capture their architecture and functional (i.e., cause-effect) relations. Then, ER data elements are processed by identifying first which elements of the developed MBSE elements they are referring to. For numeric ER data this task is fairly easy since it is possible to precisely pinpoint what MBSE elements the corresponding sensor are observing (e.g., bearing temperature of a centrifugal pump). Task is much harder for textual data since the information contained in issue or maintenance reports needs to "be understood" by a computational tool. Here we called this process "knowledge extraction". Once again, we borrow experience in the medical field where methods to extract knowledge from textual data have been developed in the past decade. The missing element for us is the availability of a complete dictionary of NPP related concepts (in addition to the MBSE models presented earlier) that can put "text into context". In Fiscal Year 2023, such a dictionary has been developed by INL under RISA along with all the computational elements required for knowledge extraction.

Lastly, once numeric and textual ER data elements have been processed and "understood", the last step is the discovery of possible cause-effect relations among them. This is performed by observing if a logical connection through the MBSE models exists, and if there is a temporal relation among them. The logic and temporal are the two main ingredients to perform "machine reasoning" from ER data.

In this report we show in detail how the integration and reasoning from numeric and textual ER data elements is performed. First, we present the developed library of MBSE models that focus on common NPP systems and assets. Then we show the development of computational methods designed to process and analyze numeric and textual ER data elements simultaneously. Here, we apply the developed computational methods on the circulating water system of an existing NPP. Our final considerations provide additional insights on further potential applications of these methods to support additional NPP decisions.

CONTENTS

FIGURES

TABLES

ACRONYMS

CWS	circulating water system
ER	equipment reliability
FY	fiscal year
LWR	light-water reactor
MBSE	model-based system engineering
ML	machine learning
NPP	nuclear power plant
NLP	natural language processing
OPM	object process methodology
PHM	prognostic and health management

DEVELOPMENT OF ANALYSIS METHODS THAT INTEGRATE NUMERIC AND TEXTUAL EQUIPMENT RELIABILITY DATA

1. INTRODUCTION

In past decades, existing nuclear power plants (NPPs) have been transitioning from corrective and periodic maintenance to predictive maintenance strategies to reduce operation and maintenance costs. While corrective maintenance is performed only when the asset fails (with high costs due to asset replacement and unexpected system and plant unavailability, e.g., loss of generation), periodic maintenance is performed at specific time intervals based on reliability factors and past operational experience (with high costs due to continuous maintenance operations that may not be warranted). On the other hand, predictive (i.e., performance based) maintenance operations are designed to be performed only when the asset under consideration requires it. The approach requires advanced prognostic (Okoh, 2001) or health management (PHM) techniques (Pecht and Kang, 2019) and this can be achieved by constantly monitoring asset status and performances and processing such data (through anomaly detection, diagnostic, and prognostic computational algorithms) to identify asset degradation trends and faulty states (Zio, 2013).

The transition from periodic or corrective maintenance to predictive maintenance (see Figure 1) is designed so that maintenance occurs only when the asset requires it (i.e., before its imminent failure). This guarantees that asset availability is maximized and operation and maintenance costs are minimized (Xingang, 2021). These benefits can be achieved by employing monitoring sensors, automated data acquisition systems, data analysis methods, and improved decision processes. When combined together, they can provide precise information about the health of an asset, track its degradation trends, and provide information of its expected failure time. With such information, maintenance operations can be scheduled and performed for each asset only when needed.

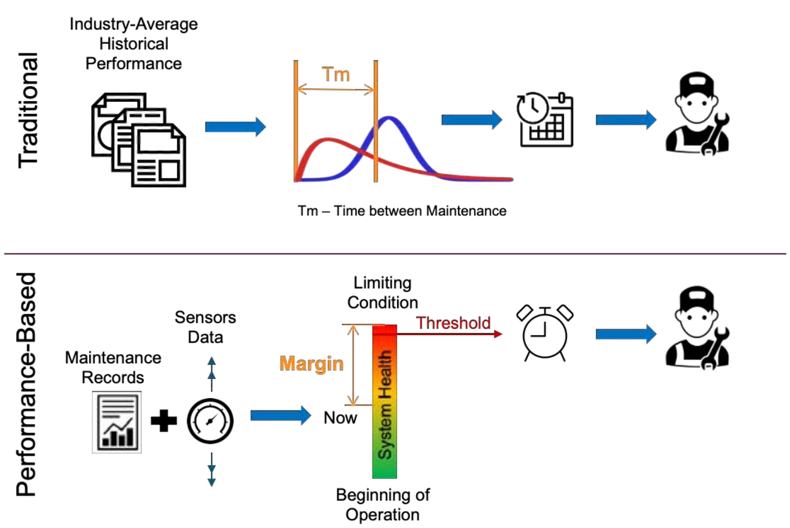

Figure 1. Graphical representation of traditional (i.e., periodic) vs. performance based (i.e., only when needed) maintenance approaches.

NPP PHM activities produce a large amount of equipment reliability (ER) data that contain information about the status of component, assets, and systems. Such data can come in several forms, such as online monitoring data (e.g., pump vibration data, pump mass flowrate), surveillance and testing data (performed by plant operators at regular intervals), condition reports (which typically contain anomalous conditions), or maintenance reports (which indicate operations performed to restore component or asset health). All these data elements precisely record assets and systems performance and health throughout their lifecycle. In addition, such data have the potential to provide insights to system engineers about the presence of anomalous behaviors or degradation trends, the possible causes of such behaviors and trends, and identify in advance their direct consequences.

However, several challenges have proven to be roadblocks to reach such potentials. While some are technical in nature (i.e., data is often distributed over several physical servers and databases), some others are conceptual: data elements have different formats (e.g., numeric, textual) and measured values have different scales (e.g., vibration spectra, oil temperature). This report directly tackles the latter roadblock and focuses on the integration of numeric and textual data elements to assist plant system engineers analyze ER data.

This task starts by extracting knowledge from textual data using natural language processing (NLP) methods and quantifying system, asset, and component health from numeric data. Then we employ model-based system engineering (MBSE) models of systems and assets to identify their architecture and functional (i.e., cause and effect) relations. ER data elements are then associated with a single MBSE entity based on their nature. This bonding of MBSE models and ER data elements constitutes the first-of-its-kind knowledge graph of a system of an NPP. At this point, data elements are organized in a structured way such that system engineers can identify cause-effect patterns between data elements and act accordingly.

This report is structured as follows:

- Sections 2–6 provide an overview of the activities performed during Fiscal Year (FY) 2023 in terms of the analysis of numeric and textual ER data and their integration in order to assist systems engineers on the identification of degraded performance and correlation between events.
- Appendices A and B provide more technical details of this activity in the form of journal papers, which will be submitted shortly after the release of this report.

2. DIGITAL REPRESENTATION OF SYSTEM KNOWLEDGE

The ability of NPP system engineers to analyze ER data relies on their knowledge about system architecture and the physical and logical interdependencies between the assets that are part of such a system. Current ER data analysis tools rely only on available data, and they are blind on the actual operating *context* that have generated such data. The term context here refers to the actual physical element being monitored and observed, the function(s) supported by such a physical element, and the other elements directly linked to it.

In order to address this limitation, we have developed a set of methods that are based not solely on data but also models. The objective of these models is to emulate system engineer knowledge and capture system architecture and the physical and logical interdependencies between the assets that are part of such a system. Here, we are employing state-of-the-art MBSE methods, which provide several solutions to represent systems, assets, and components from both *form* (i.e., which elements are part of the structures, systems, and components) and *functional* (i.e., how systems and assets interact with each other and which functions they support) points of view. These solutions are based on MBSE languages that represent system and asset form and functional elements through a set of diagrams. The most commonly used languages are object process methodology (OPM) (Dori, 2002), unified modeling language (William, 2004), and systems modeling language (Friedenthal, 2008). For the scope of this project, we have chosen the OPM language because it provides basic modeling elements we are looking for, and more importantly, it is possible to automatically generate digital data structures (i.e., graphs) from OPM diagrams. Each element of an OPM diagram indicates either a *function* or *form* element. Links between OPM elements have precise meaning (Dori, 2002).

As an example, Figure 2 shows a simplified OPM model of a centrifugal pump; this diagram indicates the existence of form and functional elements (shown using rectangular and ellipsis shapes, respectively). The links among them have a precise grammar, and a portion of them are shown in Figure 2:

- Characterization link identifies an attribute (e.g., pressure) of a form entity (i.e., the fluid operand)
- Transformation link identifies how a function affects a form entity
- Transformation link identifies the function that is supported by a form entity
- Composition link identifies the constituent elements of a form entity.

During FY23, we have developed a library of MBSE OPM diagrams that cover several light-water reactor (LWR) systems, assets, and components, as listed in Table 1. Such a list covers the many relevant assets of existing LWRs that are normally under PHM observation. Through these models we can put ER data (both textual and numeric) into context and perform machine reasoning (i.e., cause-effect analyses).

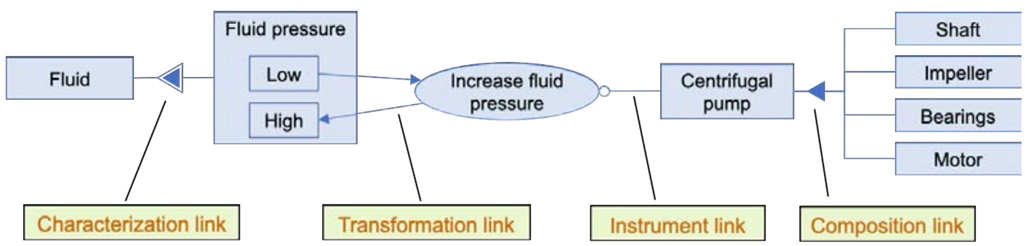

Figure 2. Simplified OPM model of a centrifugal pump.

Table 1. List of developed MBSE models.

LWR Systems	LWR Assets and Components
Reactor building	Pressurizer
Primary system	Steam generator
Secondary system	Centrifugal pump
Reactor coolant pump (RCP) system	Check valve
Reactor pressure Vessel (RPV) system	Closed feedwater heater
Pressurizer system	Motor-operated valve
CWS system	Turbine-driven centrifugal pump
	Bearings

3. ANALYSIS OF NUMERIC EQUIPMENT RELIABILITY DATA

Many NPP assets are continuously monitored (e.g., vibration data, oil temperature, and outlet water pressure) via advanced PHM systems in order to identify data trends that may inform system engineers of degraded performance or failure of the considered asset. One challenge with complex systems (not only nuclear) is the numeric quantification of the health of each asset such that is independent from the type of the available data (either condition, diagnostic, or prognostic data).

Mandelli (2023) describes a valid approach to overcome these challenges using a margin-based approach (see Figure 3). Such an approach converts the data generated by condition-assessment, diagnostic, and prognostic systems into margin values that serve as a quantitative measure of asset health. The margin value of an asset is not static but changes with time, depending on asset conditions. As an example, if degradation due to usage is observed from the monitoring data, the corresponding asset margin value decreases. Conversely, if a maintenance operation is performed on that same asset (e.g., restoration of centrifugal pump bearings), the asset margin value increases.

A relevant benefit of a margin approach is that it is possible to effectively propagate margin values from the asset level to the system level in order to assess system health using well known reliability models (such as reliability block diagrams). From there, it is then possible to identify which elements are more critical to guarantee system operation (see Figure 4). With that information, system engineers have an analytical approach to prioritize maintenance operations. Such an approach is unique in this respect since it harmonizes different numeric ER data sources through the margin definition illustrated in Figure 3.

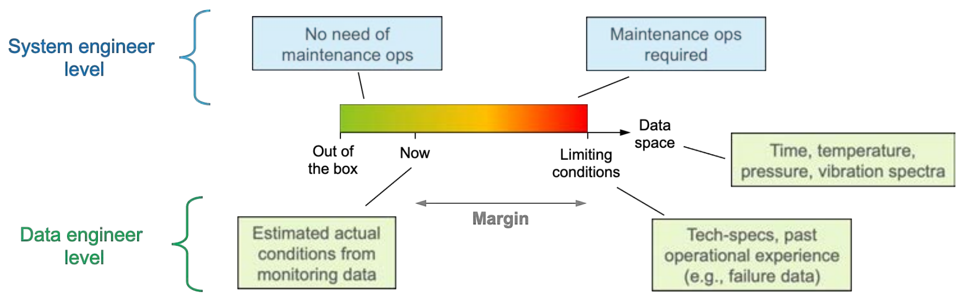

Figure 3. Graphical representation of margin-based actual asset monitoring data.

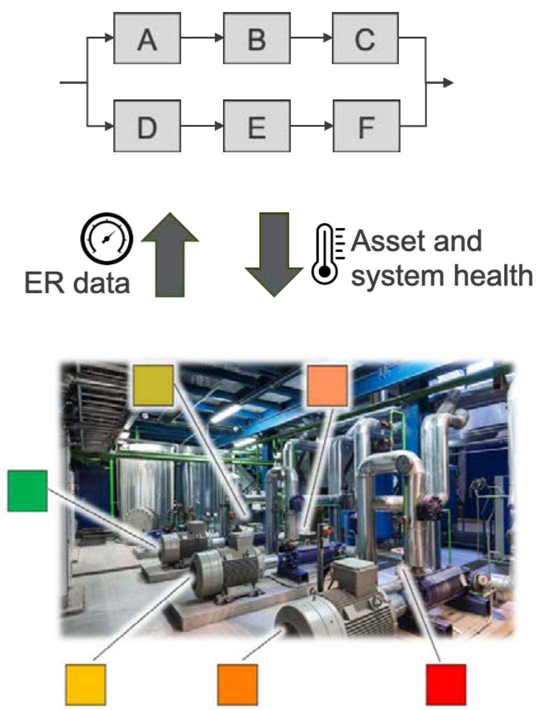

Figure 4. Graphical presentation of asset margin assessment and propagation from the asset to the system level to prioritize maintenance operations.

4. ANALYSIS OF TEXTUAL EQUIPMENT RELIABILITY DATA

In our context, the objective of analyzing textual ER data is to automate the extraction of quantitative knowledge from textual data and assist system engineers in assessing system health. The concept of "knowledge extraction" is very broad, and its definition might vary depending on the application context. In this paper, we have focused our attention on three types of common analysis from ER textual data: health evaluation of a reported event, causal relation between events, and temporal relationship between reported events.

The goal of our NLP methods is to extract both qualitative and quantitative knowledge. Hence, machine learning (ML) methods based on a supervised or unsupervised algorithm do not really suit our scope since they only provide qualitative information (e.g., which user-specified class a sentence belongs to). Our approach relies on both ML and rule-based methods. More specifically, for each of the three-analyses listed above, our NLP methods are looking within each sentence and paragraph at specific keywords, sentence architecture relations, and structures.

From a general view, our set of methods consists of the following functionalities that cover the actions needed to perform ER data analysis:

- *Data preprocessing*
    - Abbreviation handling: replace identified abbreviations with the full term
    - Acronym identification: identify the meaning of identified acronyms
    - Spellcheck: automatically correct the misspelled word
- *Entity recognition*
    - Temporal attributes identification: identification of dates and time of events
    - Measured quantities identification: identification of numbers and corresponding unit of measure obtained from measurement assessments
    - Nuclear keywords identification: identification of nuclear related entities such as mechanical, hydraulic components and assets, chemical reactions, and degradation mechanisms
- *Knowledge extraction*
    - Health status identification: evaluate the nature of the event (e.g., issue report, maintenance report)
    - Temporal sequencing of events identification: assess if the report provides a temporal relationship between events
    - Cause-effect relations identification: assess if the report provides a causal relationship between events
    - Conjecture identification: determine whether the report contains information about future prediction (e.g., an event that can occur in the future) or hypothesis about past events (e.g., a failure that might have occurred).

Figure 5 illustrates an example of knowledge extraction from an issue report (IR) where several entities have been recognized; from a semantic point of view, a conjecture causal relation between two events has been detected:

$$(Cracks, pump, shaft) \xrightarrow{conj} (pump, failure)$$

As a last comment, note that our models do not require any form of training provided past data; rather, they can be employed right away assuming the context is similar.

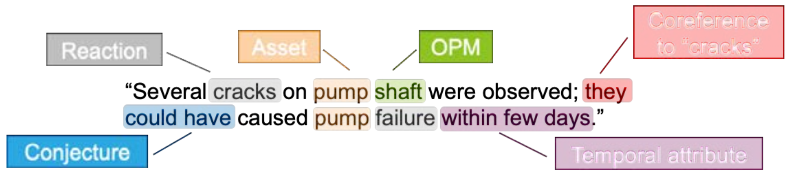

Figure 5. Example of knowledge extraction from an ER textual data element.

5. CAPTURING CAUSAL RELATIONSHIP BETWEEN EVENTS

As initially mentioned in Section 1, a causal relationship between events is defined here as the combination of their mutual temporal and logical relation. More specifically, by logical relation we imply that the occurrence of an event has triggered a series of phenomena which can be either physics based[1] or digital[2]. The temporal relation is an additional requirement we impose to avoid that two events that are logically related are too far apart from a temporal point of view.

When dealing with two events detected only by textual ER data elements (or only by numeric ER data elements), causal analysis can be performed simply by directly employing MBSE models to check the logical and temporal relations. On the other hand, when dealing with the mixed case, numeric and textual data elements, slightly different thinking applies from a temporal standpoint. More precisely, we need to understand if the occurrence of an event (e.g., reported as an IR) has triggered a change in the numeric counterpart or vice versa. Once the NLP knowledge extraction is performed, it is possible to temporally characterize it (i.e., define event time of occurrence and its duration, if available). The following step is to assess the behavior of the time series prior, after, and during the occurrence of such an event; in our work such assessment is performed through a classical two-sample testing algorithm.

An example is shown in Figure 6 where two events (indicated as E_1 and E_2) are analyzed along with a time series TS (shown in blue). Events E_1 and E_2 (provided in textual form) are analyzed to capture the nature of the event along with their temporal attributes (i.e., time of occurrence and duration, if available). Then, the provided numeric time series is analyzed in order to identify if there is a temporal correlation with each event. For E_1, the developed two-sample testing flags the existence of such correlation; for example, *after* the occurrence of E_1 the time series behavior changes (which is indicated as $E_1 \rightarrow TS$). Similarly, we obtain a temporal correlation between E_2 and the time series, which is confirmed by the behavior of the time series *while* event E_2 is occurring (which is indicated as $E_2 \leftrightarrow TS$).

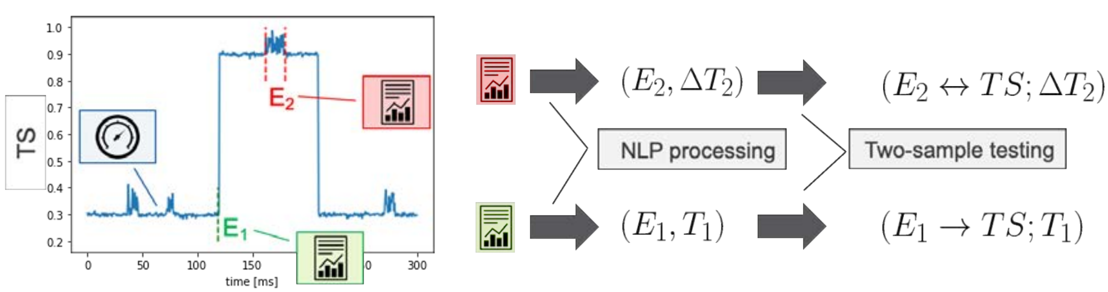

Figure 6. Identification of the temporal relations between numeric and textual events (adapted from [Luo, 2014]).

[1] Through an exchange of mass, momentum, or energy.

[2] Through an exchange of digital data via a communication system.

6. EQUIPMENT RELIABILITY DATA ANALYSIS APPLIED TO CIRCULATING WATER SYSTEM

The developed methods have been tested and applied to the analysis of the circulating water system (CWS) of an existing pressurized-water reactor plant (see Figure 7). The CWS is an important non-safety-related system. As the heat sink for the main steam turbine and associated auxiliaries, the CWS is designed to maximize steam power cycle efficiency (Agarwal et al., 2021a; 2021b). A CWS consists of the following major equipment: vertical motor-driven circulating pumps (each with an associated fixed trash rack and traveling screen at the pump intake to filter out debris and marine life), main condenser, condenser waterbox air removal system, and circulating water sampling system.

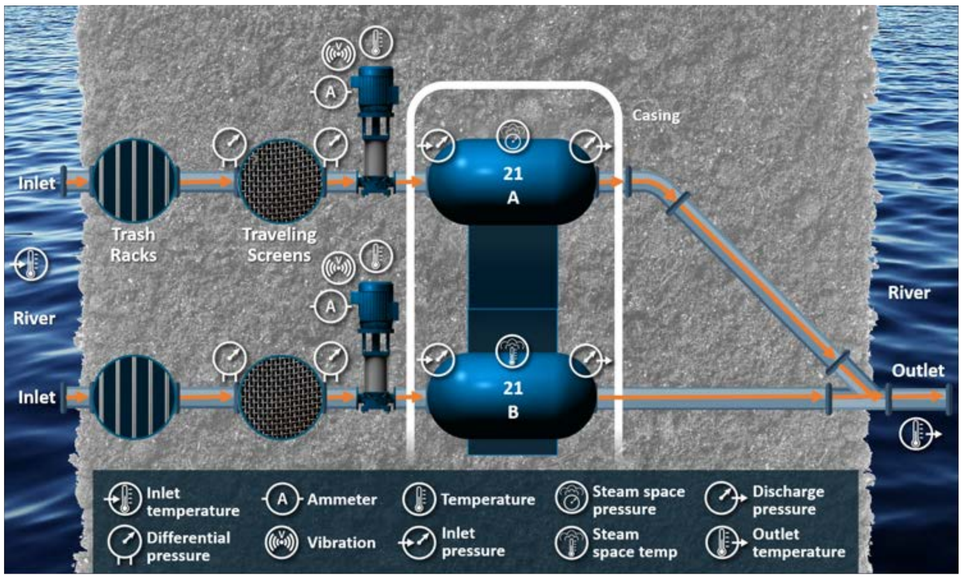

Figure 7. Plant CWS system with sensors and instrumentation.

A large amount of historical data has been collected and employed to track the health profile of the system. This dataset consists of a large number of variables of a different nature (e.g., temperature, motor current, vibration data). An important observation here is that in some temporal windows, data entries for these monitored variables might be missing due to a failure of the monitoring system. By performing a separate analysis of the available maintenance logs, it has been possible to identify several faulty states throughout the history of the CWS system.

We applied the methods described in Section 3, 4, and 5, and more specifically, by following these steps:

1. Develop an MBSE model of the CWS system and its major assets. In this case, the centrifugal pumps since most of the monitored data were located around these assets. Once linked, we obtained a network of MBSE models that capture the form and functional architecture of the overall system. Then, we deployed our analysis tools to transform MBSE models into digital structures (i.e., graph-based), which will form the skeleton of the knowledge graph (see Section 2).
2. Develop margin models for the CWS system given available numeric ER data (see Section 3). In our case, we relied on the models that were based on the available condition-based data and ML models (Hastie, 2001) developed in collaborations with the plant modernization pathway (again under the Light Water Reactor Sustainability Program). Such margin models allow us

to detect transitions from healthy conditions to faulty states. These transitions are the elements that are relevant to be captured from an ER knowledge point of view.

3. Apply our NLP knowledge extraction methods to the available set of work orders (WOs) and IRs generated from the CWS system. The textual description of the events contained in these reports are usually very short single sentences that often contains acronyms and abbreviations. Without data cleaning and curation, knowledge extraction would not be possible. Our developed methods (see Section 4) were able to clean enough elements required to extract the most relevant information. Further development and testing will continue soon.

4. The digital structures (i.e., graphs) generated from the MBSE models are then populated by:

    a. Assigning textual elements that mention specific MBSE entities to those entities

    b. Assigning abnormal events from the margin analysis to the corresponding MBSE entities

    c. Identifying possible causal links between the data elements assigned to the MBSE entities; note while this task might appear computationally expensive, due to the large amount events, the initial evaluation of the logical relationship among them filters out unnecessary temporal evaluations.

7. CONCLUSIONS

This report has presented a summary of the activities performed during FY23 regarding the analysis of ER data to support system engineers' decision making. The starting point is the challenges behind the analysis of ER due to their large size (i.e., large number of monitored sensors and data recorded over decade of operation) and their heterogenous formats (e.g., numeric and textual). Such ER data sources are very difficult to analyze manually; however, system engineers have the valuable knowledge to interpret ER data elements and identify possible temporal or casual-effect links between them given their knowledge of the systems and assets under considerations. Our work focused on development of both models—designed to emulate such system engineer's knowledge in a digital form—and methods to put ER data into context.

The developed methods have been tested on the CWS system of an existing NPP; the provided data consisted of both monitored data and textual data. The analyses of these two classes started separately (i.e., knowledge extraction for textual ER data and margin analysis for numeric ER data) while data integration is finally completed using the CWS MBSE model as a common data skeleton. From there, our ER analysis methods can detect temporal and logical relationships between ER data elements to perform a first-of-its-kind application "machine reasoning".

The details of our work are presented in the two appendices, while the first focus mainly on the process of knowledge extraction from textual data, the second one focuses on the actual integration of numeric and textual ER data. These two appendices also present few considerations with the goal of introducing additional applications of the developed methods. As an example, the sentence similarity measures are proving very effective to identify if a just observed event (reported in textual form) has occurred in the past.

REFERENCES

Agarwal, V., K. A. Manjunatha, J. A. Smith, A. V. Gribok, V. Yadav, H. Palas, M. Yarlett et al. (2021a). "Machine Learning and Economic Models to Enable Risk-Informed Condition Based Maintenance of a Nuclear Plant Asset." INL/EXT-21-61984, Idaho National Laboratory. https://www.osti.gov/servlets/purl/1770866.

Agarwal, V., K. A. Manjunatha, A. V. Gribok, T. J. Mortenson, H. Bao, R. D. Reese, T. A. Ulrich et al. (2021b). "Scalable Technologies Achieving Risk-Informed Condition-Based Predictive Maintenance Enhancing the Economic Performance of Operating Nuclear Power Plants." INL/EXT-21-64168, Idaho National Laboratory. https://doi.org/10.2172/1894498.

Hastie, T., R. Tibshirani, and J. Friedman. (2001). *The Elements of Statistical Learning*. New York: Springer. https://doi.org/10.1007/978-0-387-84858-7.

Luo, C., J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang. (2014). "Correlating Events with Time Series for Incident Diagnosis." KDD'14: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1583–1592. https://doi.org/10.1145/2623330.2623374

Mandelli, D., C. Wang, and S. Hess (2023). "On the Language of Reliability: A System Engineer Perspective." Nuclear Technology, Selected papers from the PSA 2021 special issue. https://doi.org/10.1080/00295450.2022.2143210.

Okoh, C., R. Roy, J. Mehnen, and L. Redding. (2014). "Overview of Remaining Useful Life Prediction Techniques in Through-Life Engineering Services." Procedia CIRP 16, pp. 158–163. https://doi.org/10.1016/j.procir.2014.02.006.

Pecht, M., and M. Kang. (2019). "Introduction to PHM. Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things." IEEE, pp. 1–37. https://doi.org/10.1002/9781119515326.ch1.

Xingang, Z., J. Kim, K. Warns, X. Wang, P. Ramuhalli, S. Cetiner, H. G. Kang, and M. Golay. (2021). "Prognostics and Health Management in Nuclear Power Plants: An Updated Method-Centric Review with Special Focus on Data-Driven Methods." *Frontiers in Energy Research*, 9, 696785. https://doi.org/10.3389/fenrg.2021.696785.

Zio, E. (2013). "Prognostics and health management of industrial equipment." In *Diagnostics and Prognostics of Engineering systems: Methods and Techniques*, edited by S. Kadry, pp. 333–356. Hershey, PA, USA: IGI Global. https://doi.org/10.4018/978-1-4666-2095-7.ch017.

APPENDIX A

Natural Language Processing Toolset for Equipment Reliability Data Analysis

Natural Language Processing Toolset for Equipment Reliability Data Analysis

Congjian Wang[a], Diego Mandelli[a], J. Cogliati[a]

[a] *Idaho National Laboratory*

ABSTRACT

Complex engineering systems such as nuclear power plants (NPPs) are generating and collecting a large amount of equipment reliability (ER) element data that contain information about the status of components, assets, and systems. Some of this information is in textual from where, typically, events such as issue reports and maintenance activities are described. The analyses of textual data in current NPPs using natural language processing methods have grown in the last decade and only recently the potentials of this kind of analyses have emerged. So far, applications of natural language processing methods have been limited to mostly classification and prediction to identify the nature of the textual element (e.g., safety or non-safety relevant). Here we are targeting a more complex problem: the automatic generation of knowledge out of a textual element to assist system engineers in assessing system health. “Knowledge extraction” is very broad concept, and its definition might vary depending on the application context. In our context, knowledge extraction means that, out of an ER textual element, we want to identify the systems or assets mentioned in it and the type of event described in it (e.g., a component failure or a maintenance activity). In addition, we want to identify details such as measured quantities and temporal or cause-effect relations between events. This paper describes how ER textual data elements are first preprocessed to handle typos, acronyms, and abbreviations and then machine learning and rule-based algorithms are employed to identify physical entities (e.g., systems, assets, components) and specific phenomena (e.g., failure, degradation). Several applications relevant from a NPP ER point of view are presented as well.

Keywords: natural language processing, knowledge extraction, machine learning

List of acronyms

ER	equipment reliability
IR	issue report
LWR	light-water reactor
ML	machine learning
NLP	natural language processing
NPP	nuclear power plant
OPM	object-process methodology
POS	part of speech
PWR	pressurized-water reactor
SSCs	structures, systems, and components
WO	work order

1 Introduction

To reduce operation and maintenance costs, existing nuclear power plants (NPPs) are moving from corrective and periodic maintenance to predictive maintenance strategies. This transition is designed so that maintenance occurs only when component requires it (e.g., before its imminent failure). This guarantees that component availability is maximized and that maintenance costs are minimized. However, these benefits require changes in the data that need to be retrieved and the type of decision processes to be employed. Advanced monitoring and data analysis technologies are essential to support predictive strategies. They can in fact provide precise information about the health of a component, track its degradation trends, and provide an estimate of its expected failure time. With such information, maintenance operations for a component can be performed right before its expected failure time.

This dynamic context of operations and maintenance operations (i.e., predictive) requires new methods to process and analyze equipment reliability (ER) data. A relevant issue is that ER data can have heterogenous data formats: textual, numeric, image, etc. The analysis of numeric ER data has been addressed in many works (Xingang et al., 2021) and applied to many operational directions, including anomaly detection, diagnosis, and prognosis. The analysis of textual data has been investigated only recently using machine learning (ML) methods (Young et al., 2018) designed to assess their nature (e.g., safety or non-safety related), and there is no unified toolset that system engineers could adopt to analyze textual ER data. The information contained in ER data in textual form refers to events such as issue reports (IRs) or maintenance activities (or work orders [WOs]) that have been performed.

This paper primarily focuses on applying NLP methods for ER data analysis to support robust decisions in a plant operation context. NPPs are constantly monitoring the status and performance of many systems and components. Hence, a large amount of textual data is continuously being generated. The objective of analyzing textual ER data is to automate the extraction of quantitative knowledge from textual data and assist system engineers in assessing system health. The concept of "knowledge extraction" is very broad, and its definition might vary depending on the application context. In this paper, we have focused our attention on three types of analysis that are common for ER textual data:

- Health evaluation of a reported event
- Causal relation between events
- Temporal relationship between events

The goal of our NLP methods is to extract both qualitative and quantitative knowledge. Hence, ML methods based on a supervised or unsupervised algorithm do not really suit our scope since they only provide qualitative information (e.g., which user-specified class a sentence belongs to). Our approach relies on both ML and rule-based methods. More specifically, for each of the three analyses listed above, our NLP methods are looking within each sentence and paragraph at specific keywords, sentence architecture relations, and structures. To improve the clarity of this paper, the following elements are here defined:

- *Text*: the actual raw content of an issue report (IR) that can be composed of multiple sentences
- *Sentence*: the base element of a text that expresses a complete concept, which can be simple (i.e., single clause) or complex (i.e., multiple clauses)
- *Clause*: a grammatical constituent that consists of a subject and a predicate.

2 Library Overview

2.1 Software Architecture

Our software analysis tool provides algorithms and functions for ER data analysis built on the `SpaCy` library. It follows the ASME NQA-1 quality assurance standard and a procedural programing style, and provided pipelines (e.g., functions or modules) are organized as shown in Figure 1. The code is developed on top of well-established `Python` libraries, such as `SpaCy`, `numpy`, and `Pandas`. With the basic NLP capabilities provided by `SpaCy` (dependency parsing, part of speech (POS) tagging, tokenizing), our methods focus on extracting health status, cause-effect information, and temporal relationship from ER data that can support a robust decision in the plant operation context.

2.2 Software Functionalities or Capabilities

From a general view, our set of methods consists of the following functionalities and pipelines that cover the actions needed to perform ER data analysis as illustrated in Figure 1:

- Utilities for text processing, such as removing, replacing, and normalizing of the text
- Abbreviation handling replaces the abbreviations with their full name
- Spellcheck automatically correct the misspelled text
- Acronym identification identifies the meaning for the acronyms
- Temporal attributes identification
- Temporal sequencing of events identification
- Measured quantities identification
- Nuclear keywords identification
- Conjecture identification determines whether the clause is conjecture or not
- Cause-effect relations identification
- Health status identification
- Text similarity identification
- Knowledge graph construction

3 Capabilities

3.1 Leverage Capabilities from External Libraries

In this work, we have integrated SpaCy (https://github.com/explosion/spaCy), PySBD (https://github.com/nipunsadvilkar/pySBD), Coreferee (https://github.com/msg-systems/coreferee), and few other libraries for text data analyses. SpaCy is an open-source Python library including tagging, parsing, NER, text classification, and more. It features state-of-the-art speeds and provides a variety of linguistic annotations to give insights into a text's grammatical structure. PySBD is a rule-based sentence boundary disambiguation Python package, released under the MIT license, to determine sentence boundaries. Coreferee is also an open-source Python library, released under the MIT license, to resolve coreferences. Table 1 presents a list of analysis steps we employed to process digital text data.

3.2 Tokenization

The first step in processing the text is to tokenize it using a `SpaCy` tokenizer (i.e., segment it into a list of words, punctuation, and so on) by applying rules specific to raw text, as illustrated by Figure 2. First, the raw text is split on whitespace characters. Then, the tokenizer processes the text from left to right. On each substring, it performs two checks:

1. Does the substring match a tokenizer exception rule? For example, “don’t” does not contain whitespace but should be split into two tokens, “do” and “n’t.”
2. Can a prefix, suffix, or infix be split off, such as punctuation like commas, periods, hyphens, or quotes?

If there’s a match, the rule is applied, and the tokenizer continues its loop, starting with the newly split substrings. This way, the tokenizer can split complex, nested tokens like combinations of abbreviations and multiple punctuation marks.

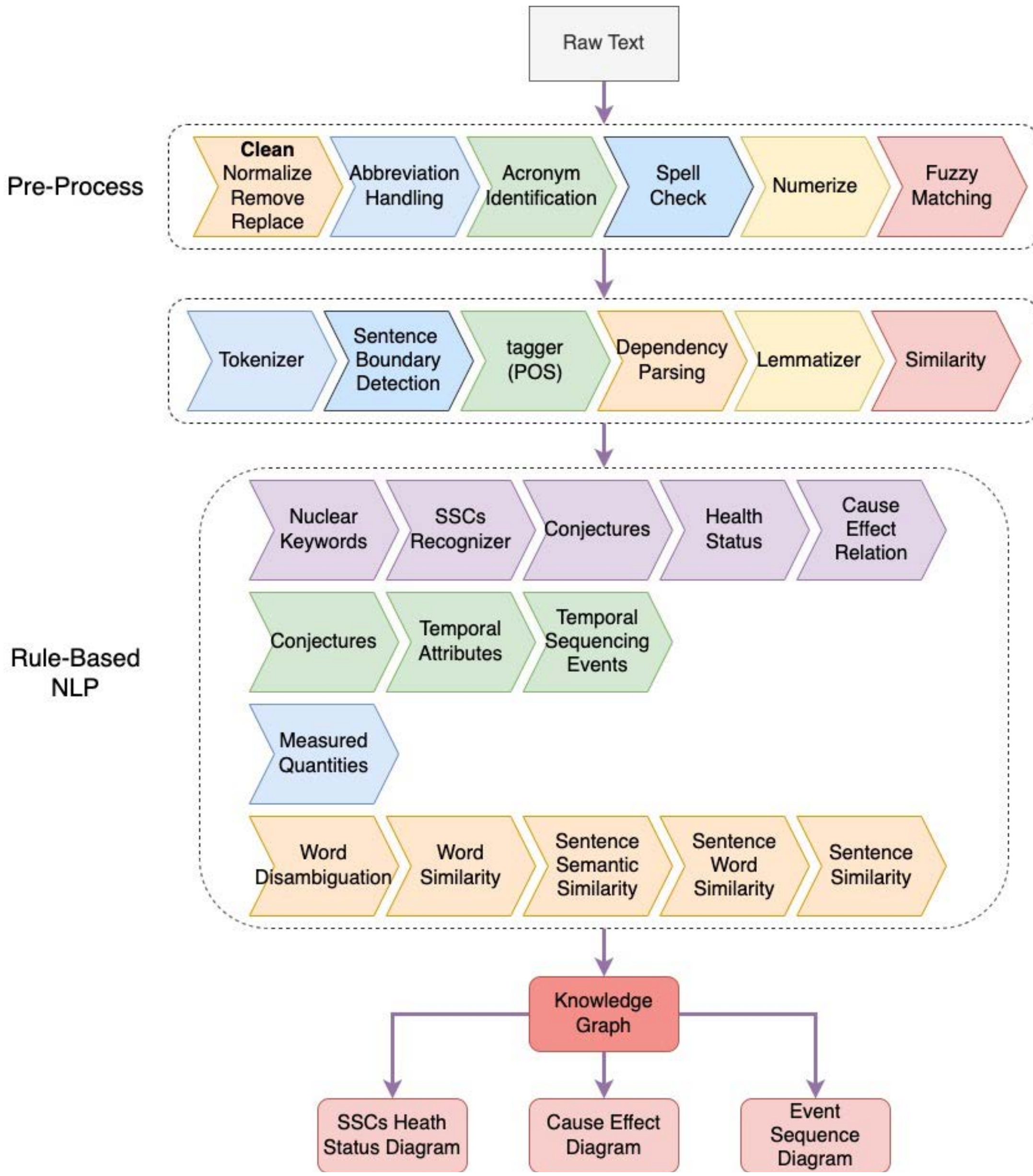

Figure 1. Graphical illustration of the developed NLP analysis tools.

Table 1. Leveraged capabilities from external libraries.

ID	NLP Steps	NLP Pipeline	Note
1	Tokenization	tokenizer (SpaCy)	Segmenting text into words, punctuations marks, etc.
2	Sentence segmentation	pysbdSentenceBoundaries (PySBD)	Finding and segmenting individual sentences
3	POS	tagger (SpaCy)	Assigning word types to tokens, like verb or noun
4	Dependency parsing	parser (SpaCy)	Assigning syntactic dependency labels and describing the relations between individual tokens, like subject or object
5	Lemmatization	lemmatizer (SpaCy)	Assigning the base forms of words, such as the lemma of "was" is "be" and the lemma of "pumps" is "pump"
6	Similarity	tok2vec (SpaCy)	Comparing words, text spans, and documents and how similar they are to each other
7	Rule-based entity recognition	entity_ruler (SpaCy)	Finding sequences of tokens based on their texts and linguistic annotations and labeling named SSCs
8	Coreference	Coreferee (Coreferee)	Resolving coreference situations where two or more words within a text refer to the same entity

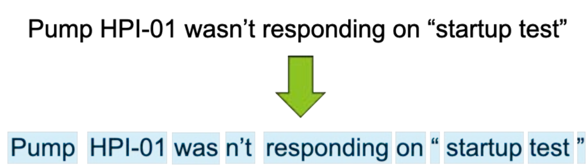

Figure 2. Tokenization process: given the provided text, the obtained tokens are highlighted in blue.

3.3 Sentence Segmentation

The next important step is to determine the sentence boundaries, that is, segment the text into a list of sentences. It is a key underlying task for NLP process. In this work, we employ PySBD (https://github.com/nipunsadvilkar/pySBD), a rule-based sentence boundary disambiguation Python package, to detect the sentence boundaries. We have developed a custom pipeline using PySBD that is used with SpaCy to split text into a list of sentences. In general, there are three different approaches to segment sentences: 1) rule based, requiring a list of hand-crafted rules, 2) supervised ML, requiring training datasets with labels and annotations, and 3) unsupervised ML, requiring distributional statistics derived from raw text. We choose the rule-based approach because the errors are interpretable and the rules can be adjusted incrementally. Moreover, the performance can be better than the ML models. For example, `PySBD` passes 97.93% of the Golden Rule Set exemplars (a language-specific set of sentence boundary exemplars) for English, with an improvement of 25% over the next best open-source `Python` tool (Sadvilkar and Neumann, 2020).

3.4 Part of Speech

After the correct segmentation of sentences, `SpaCy` tagger is used to parse each sentence and tag each token in the sentence. Both "TAG" and "POS" attributes are generated for each token after the `SpaCy` tagger process. "POS" is the simple universal POS tag, does not include information for any morphological features, and only covers the word type (https://universaldependencies.org/u/pos/). The morphology is the process by which a root form of a word is modified by adding prefixes or suffixes that specify its grammatical function but do not change its POS. These morphological features are added to each token after the POS process and can be accessed through token's "morph" attribute. In addition, the "TAG" attribute expresses the POS and some amount of morphological information. For example, the POS "VERB" tag is expanded into six "TAG" tags, "VB" (verb, base form), "VBD" (verb, past tense), "VBG" (verb, gerund, or present participle), "VBN" (verb, past participle), "VBP" (verb, non-third person singular present), and "VBP" (verb, third person singular present). In this work, we employ these POS and TAG tags to determine the description of the SSC health status (conjecture or qualitative observations).

3.5 Dependency Parsing

After the correct segmentation of sentences, that `SpaCy` tagger is used to parse each sentence and tag each token in the sentence. Both "TAG" and "POS" attributes are generated for each token after the `SpaCy` tagger process. "POS" is the simple universal POS tag, does not include information for any morphological features, and only covers the word type (https://universaldependencies.org/u/pos/). The morphology is the process by which a root form of a word is modified by adding prefixes or suffixes that specify its grammatical function but do not change its POS. These morphological features are added to each token after the POS process and can be accessed through the token's "morph" attribute. In addition, the "TAG" attribute expresses the POS and some amount of morphological information. For example, the POS "VERB" tag is expanded into six "TAG" tags, "VB" (verb, base form), "VBD" (verb, past tense), "VBG" (verb, gerund, or present participle), "VBN" (verb, past participle), "VBP" (verb, non-third person singular present), and "VBP" (verb, third person singular present). In this work, we employ these POS and TAG tags to determine the description of the SSC health status (conjecture or qualitative observations).

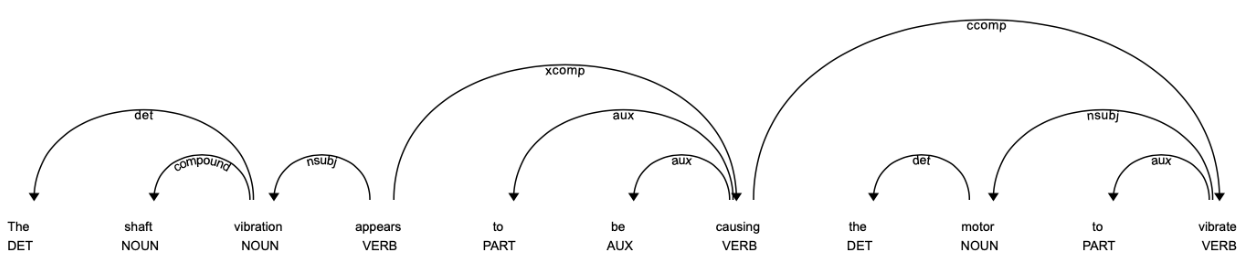

Figure 3. POS tagging and dependency parsing.

3.6 Lemmatization

A lemma is the base form of a token. As an example, the word “fail” is the lemma of “failing,” “fails,” and “failed.” Lemmatization is the process of reducing words to their base forms or lemmas. In this study, we employ the `SpaCy` lemmatizer to reduce inflectional forms or derivationally related forms of a word to a common base form. In this case, we only need to provide the base forms of keywords that leads to a significant reduction in the number of keywords.

3.7 Rule-Based Entity Recognition

In this work, we can create a set of SSCs either from the U.S. Nuclear Regulatory Commission (NRC) reports or NPP engineering models. We utilize the set of SSCs to construct patterns that can be directly passed into the `SpaCy` “entity_ruler” to identify and label the SSCs as recognized entities. The “entity_ruler” is a `SpaCy` pipeline that lets us add named entities, which makes it easy to combine rule-based and statistical NER for even more powerful pipelines. The “entity_ruler” finds matches in the text and labels them using the specified pattern label. If any matches were to overlap, the pattern matching most tokens takes priority. If they also happen to be equally long, the match occurring first in the text is chosen.

Entity patterns are dictionaries with two keys: “label,” specifying the label to assign to the entity if the pattern is matched, and “pattern,” the match pattern. The entity ruler accepts two types of patterns:

- Phrase patterns for exact matches (string)

  {“label”: “SSC”, “pattern”: “pump”}

- Token patterns with one dictionary describing one token (list)

  {“label”: “SSC”, “pattern”: [{“LOWER”: “pump”}, {“LOWER”: “shaft”}]}.

The “entity_ruler” can also accept an “id” attribute for each pattern. Using the “id” attribute allows multiple patterns to be associated with the same entity.

3.8 Coreference Resolution

Coreference are situations that often occur in texts where pronouns (e.g., it, they) are used to reference elements in the text. Coreference resolution targets the identification of the actual textual element linked to a pronoun. An example is shown in Figure 4, where the pronoun “they” refers to the previously defined textual element “cracks.” In our analysis tools we employ `Coreferee` (https://github.com/msg-systems/coreferee), an open-source `Python` library, to resolve coreferences within English texts. It uses a mixture of neural network and programmed rules to identify potential coreference mentions. In this work, we have developed several custom `SpaCy` pipelines to make `Coreferee` work seamlessly with `SpaCy`, which helps us identify causal relations among multiple sentences.

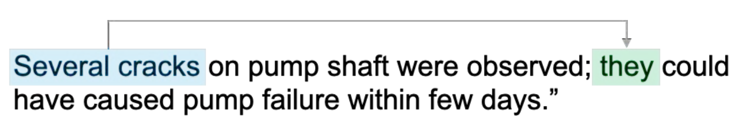

Figure 4. Example of coreference resolution.

3.9 Spellcheck, Acronym, and Abbreviation Handling

NPP IRs and WOs are usually short sentences that often contain abbreviations. The presence of abbreviations negatively impacts the ability to extract knowledge from such texts. Hence, we have developed an NLP pipeline designed to identify abbreviations and replace them with their corresponding complete word. The starting point is a library of words abbreviations that have been collected from documents available in-line. This library is basically a dictionary that contains the corresponding set of words for each identified abbreviation. A challenge here is that a single abbreviation might have multiple words associated with it. Similarly, a word might have multiple ways to be reduced.

Handling abbreviations in each sentence is performed first by identifying misspelled words. Then each misspelled word is searched for in the developed library. If an abbreviation in the library matches the misspelled word, then it is replaced by the corresponding complete word. If no abbreviation in the library is found, then we proceed by searching for the closest one. If multiple words match the obtained abbreviation, then the word that best fits the sentence context is chosen.

Another class of textual elements that are often present in ER textual data are acronyms (e.g., HPI is an acronym for the high-pressure injection system), and they typically refer to specific assets or systems of an NPP. The handling of such situations has been performed in a similar way as indicated above for abbreviations where a library of acronyms using openly available NRC, Electric Power Research Institute, and Nuclear Energy Institute documents has been developed.

After the abbreviation and acronym handler methods are completed, then the remaining misspelled words are parsed through our spellchecking methods for a last correction.

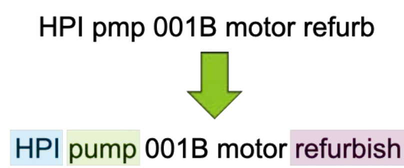

Figure 5. Example of spellcheck (i.e., word “pmp”), acronym (i.e., HPI), and abbreviation (i.e., “refurb”) handling.

3.10 Identification of Temporal Quantities

Temporal attributes indicate time instances when specific events have occurred. Time of occurrence is an important factor from a causal point of view since the emergence of an effect is always preceded by its cause. Hence, temporal information can be valuable to identify the possible links between recorded events.

Temporal quantities might come in different forms; for the scope of this article, we portioned these forms into four classes (see Table 2) that specify the occurrence of an event in absolute terms—date or time—or in relative terms (i.e., duration or frequency).

A relevant observation is related that the provided temporal information might contain some uncertainty. The handling of this situation was performed by defining a specific list of keywords that indicate approximation and their corresponding set of relations based on observed datasets (see Table 3). The set of temporal relations that have been developed are shown in Table 4.

The developed pipelines have been designed to capture temporal quantities for the four forms listed in Table 2; in this respect, an example of this identification is shown in Figure 6.

Table 2. Examples of date, time, duration, and frequency temporal expression.

Date	Time	Duration	Frequency
11/3/2005	Friday morning	10 hours	every Friday
November 3rd 2005	12:30 a.m.	last 5 months	every 4 hours
Yesterday	3 pm	2 days	every month
Tomorrow	12:30	2 days	twice a year
Thursday	12:00 am	couple of days	thrice a day
Last Week	20 minutes ago	1988–1992	

Table 3. Portion of the list of approximations that might be associated with a temporal attribute.

Approximation
About
Almost
Nearly
Roughly
Approximately
Nearly
Around
Closely
Circa
Close
Like
More or less
Roughly

Table 4. List of relations that indicate a temporal attribute.

Relations
[verb] + [at, on] + "time instance"
[verb] + [at, on] + [approximation] + "time instance"
[verb] + for + "time duration"
[verb] + for + [approximation] + "time duration"
[noun] + [verb] + "time duration"
[noun] + [verb] + [approximation] "time duration"

The valve is about twenty-nine years old.
Test was performed on 9th October
The event occurred 20 minutes ago prior the test

Figure 6. Example of identification of temporal attributes.

3.11 Identification of Temporal Sequencing of Events

Another class of textual data elements that can often be retrieved from NPPs includes IRs that report multiple events linked by temporal relations. Temporal relations can be both quantitative (e.g., an event has occurred two hours after another event) and qualitative (e.g., an event has occurred before another event). Note that a temporal relation does not necessarily imply a causal relation. In this respect, system engineering models can be employed to reconstruct the causal relationship between events if additional ER data is available. In this research, we are following the work of Moerchen (2012), which lists the major temporal relations between events (see Figure 7):

- *Order*: sequential occurrence of events
- *Concurrency*: (almost) simultaneous occurrence of events from beginning to end
- *Coincidence*: temporal intersection of events

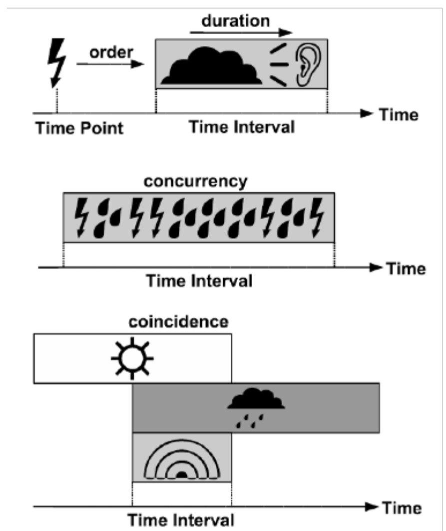

Figure 7. Graphical concepts of time-based relations: order, duration, concurrence, and coincidence of events (Moerchen, 2012).

Note that event duration (which is also indicated in Figure 7) does not provide information about temporal relations between events; instead, event duration is here considered a temporal attribute. The analysis of sentences containing temporal relations involves identifying specific keywords, relations, and grammatical structures in each sentence. In this respect, Table 5 and Table 6 provide the identified keywords (i.e., verbs, adjectives, and adverbs) and Table 7 and Table 8 provide grammatical structures that indicate the order and coincidence of events.

The example provided in Figure 8, shows how two temporal attributes have been identified which indicate a temporal sequence and concurrency of events.

Table 5. List of sample keywords and structures that indicate order of events.

Keywords			Structures
Verbs	**Adjectives**	**Adverbs**	
Antedate	After	Afterward	Soon after
Follow	Before	Consecutively	After that
Postdate	Consecutive	Consequently	After a while

Precede Predate Succeed	Earlier Following Former Later Next Past Precedent Previous Prior Subsequent Succeeding Successive	Directly Hereafter Later Next Previously Subsequently Successively Then Thenceforth Thereafter	

Table 6. List of sample keywords that indicate the concurrence and coincidence of events.

Keywords			**Structures**
Verbs	**Adjectives**	**Adverbs**	
Accompany Conform Correspond Harmonize Parallel	Accompanying Attending Coexistent Concomitant Concurrent Imminent Simultaneous Synchronic	When Thereupon While During	At that point At that moment At that time At that instant In the end On that occasion

Table 7. List of relations that indicate the order of events.

Relations
Event_1 + [order verb] + Event_2
Event_1 + [verb] + [adverb] + Event_2
Event_1 + [verb] + [adjective] + Event_2

Table 8. List of relations that indicate the concurrence and coincidence of events.

Relations
Event_1 + [verb] + [adverb] + Event_2
Event_1 + [verb] + [adjective] + Event_2

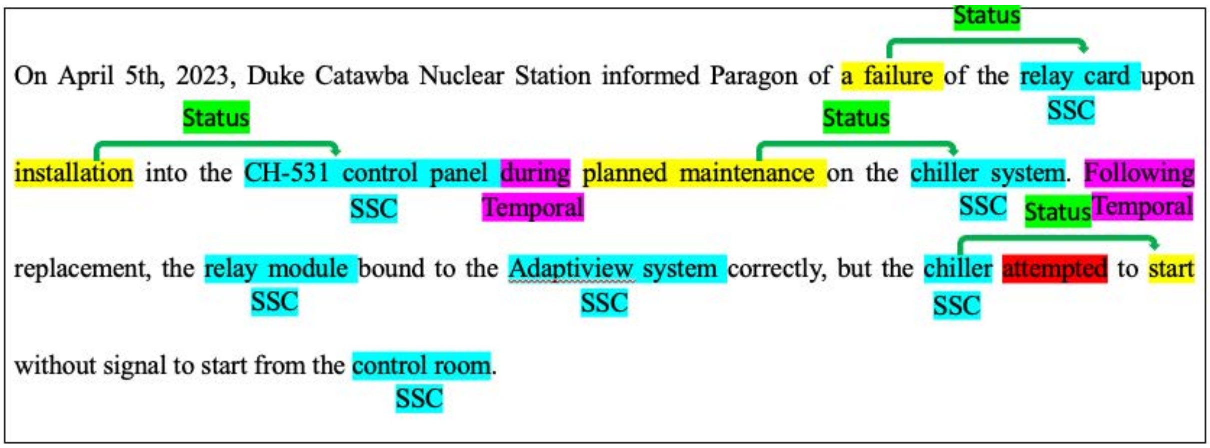

Figure 8. Example of analysis of sentences containing health status (highlighted in green) and the corresponding SSCs (highlighted in blue), temporal entities (highlighted in purple) identified from https://www.nrc.gov/docs/ML2320/ML23207A076.pdf.

3.12 Identification of Measured Quantities

Here we aim to identify a precise observation (i.e., a measured point value or delta estimate) of a measured variable. This observation requires a numeric value followed by its unit; however, it is not unusual that the unit might be missing. A relevant observation to be made at this point is that, based on the observed NPP ER textual, data measured quantities can be specified in a large variety of ways, and not only in the classic form "number + unit of measure." Examples of these possible ways are shown in Table 9.

Table 9. Example of quantitative observations.

one half
three halves
0.1
10%
3 cm
multiplied by 2
75–80%
4:1 ratio
5th percentile
within 5th and 95th percentile
the 3rd quartile
scored 6 on a 7 point scale
between three and four

For the scope of our ER applications, we focused on the development of structural relations that are listed in Table 10. Note that when referring to delta estimates, verbs and nouns convey that qualitative information (positive, negative, or neutral) can be present. In our tool, we are leveraging the existing tool quantulum3 (https://github.com/nielstron/quantulum3) and text syntactic structure for measured quantities extractions. Quantulum3 could be used to identify all possible numerical values either with or without units, while syntactic information will help to disambiguate the units from the actual natural language.

An example of identification of measured quantities are shown in Figure 9; the textual elements are taken from few U.S. nuclear regulatory commission (NRC) licensee event reports (LERs). The quantities that have been correctly identified are highlighted in blue while the one that have not been identified are highlighted in red.

Table 10. List of sentence relation for quantitative observation.

Relation
[neutral verb] + "quantity value"
[neutral verb] + "quantity delta value"
"quantity value" + [neutral noun]
"quantity delta value" + [neutral noun]
[negative verb] + "quantity value"
[negative verb] + "quantity delta value"
"quantity value" + [negative noun]
"quantity delta value" + [negative noun]
[positive verb] + "quantity value"
[positive verb] + "quantity delta value"
"quantity value" + [positive noun]
"quantity delta value" + [positive noun]

The gauge is a Berthold Model LB7440D s/n FT314 and contains a 30 mCi Cesium-137 source. The gauge contained an 8 milliCurie cesium-137 and a 40 milliCurie americium-241/beryllium source. The plan of treatment was for [the treating physician] to deliver 120 Gy to the patient's left hepatic lobe with 1.62 GBq (43.78 milliCuries) of Y-90. The initial wipes on the surface of the generator cart were 17,697 dpm and 112,368 dpm (2 different areas of the top of the cart). The enhancement is an increase in the size of the hardware to 1/4 inch bolts that connects the side panels to the bottom panel through 5/16 inch through holes with a nut and washers. The source retriever's pocket dosimeter had a reading of 155 millirem at the conclusion of the retrieval. Upon survey at receipt, the container exhibited dose rates of 3.4 rem/hr on contact, 240 mrem/hr at 12 inches, and 18 mrem/hr at 3.3 feet.

Figure 9. Example of identification of measured quantities from text taken from https://www.nrc.gov/reading-rm/doc-collections/event-status/event/2020/index.html.

3.13 Identification of Location Attributes

Similar to temporal attributes, location attributes provide qualitative information where specific events have occurred. While location information does not provide additional health information to a system engineer, it might contain clues about the health of a specific component when a reported event has occurred near it. As an example, the textual report:

"An oil puddle was found nearby pump MFW-1A" identifies an element (i.e., oil) that might have a relation to a nearby pump (i.e., MFW-1A pump). Thus, this textual report indicates how the oil element is no longer part of the OPM diagram. The identification of location attributes is being performed by looking at specific prepositions and relations listed in Table 11 and Table 12, respectively.

An example of identification of location attributes are shown in Figure 10; the textual elements are taken from few U.S. NRC LERs. Here, the identification of these attributes is very robust.

Table 11. List of sample keywords that indicate a location attribute.

Proximity	**Located Above**	**Located Below**
Across from	Above	Below
Adjacent	Anterior	Beneath
Alongside	Atop	Bottom
Approaching	Beyond	Deep
Beside	High	Down
Close	On top of	Down from
Close by	Over	Downward
Contiguous	Overhead	Low
Distant from	Upward	Posterior
In proximity		Under
Near		Underneath
Nearby		
Neighboring		
Next to		
Receding from		
Remote		
Retreating from		

Table 12. List of relations that indicate a location attribute.

Relations
[verb] + “location keyword” + noun
Subj + “location keyword” + obj

On 02/22/99, additional actions were taken to investigate the alarms which included isolating

sections of piping near the smokeheads. A fire of approximately 30' x 15' was discovered in the Camp
Location_proximity
Canoi recreation area at a location adjacent to the site of a fire on 01/12/99. There are two welds for
Location_proximity
the 1-inch pad on top of the tank that are still holding, and the licensee stated that the steam leak
Location_up
appears to be coming from an inside weld through a tell tail on the 1-inch pad. Personnel observing
Location_proximity
the HPCI surveillance locally saw water discharging from underneath the insulation on the check
Location_down
valve.

Figure 10. Example of identification of location attributes from text taken from https://www.nrc.gov/reading-rm/doc-collections/index.html#event.

3.14 Identification of Nuclear Entities

Similar to the medical field, NLP knowledge extraction methods require the capability to identify specific entities. In the nuclear field such entities include system assets and components that can be found in any NPP. Such a library for the nuclear field has been developed in past years using NRC and Electric Power Research Institute available textual data. The entities contained in such a library (about 5,000 and growing) have been grouped in eight main classes and subsequently divided into groups (mainly for data management purposes). In this respect, Table 13 lists the set of classes and groups created so far along with a few examples.

By investigating NPP IR data, we have observed that chemical elements are often reported. Hence, the identification of these elements can play a major role into the overall knowledge extraction process. In this respect, we are employing the `chemnlp` library (https://github.com/usnistgov/chemnlp) to identify chemical elements and compounds from textual elements.

An example of application of the developed method is shown in Figure 11 which is reporting an event which indicates few entities, systems, and chemical entities in either full or abbreviated form that are correctly identified.

Table 13. Class and groups of nuclear-related keywords.

Class	Group	Examples
Mechanical components	Fasteners Rotary elements Structural Purpose specific	Anchor bolt, cap screw, latch, pin Cam, shaft, gear, pulley Beam, column, sleeve, socket Filter, manifold, blade
Non-mechanical components	Electrical/electronic Hydraulic/Pneumatic	Amplifier, relay, buzzer, capacitor Coupler, filter, pipe
Assets	Mechanical	Engine, vessel

	Electrical Hydraulic/Pneumatic Electronic I&C Nuclear fuel	AC bus, alternator, generator, transformer Pump, valve, condenser, fan Computer, tablet, controller Digital meter, FPGA, transmitter, sensor Fuel rod, control blade
NPP elements	Systems Architectural	Feedwater, switchyard, feedwater Containment, control room, pump house
Tools and treatments	Tools Treatments	Jigsaw, solder gun, tape, crane Bolting, riveting, grinding, infrared testing
Operands	Electrical Hydraulic/Pneumatic	AC current, electromagnetic Compressed air, steam, gasoline, water
Compounds	Materials	Plastic, plywood, concrete, polyethylene
Reactions	Chemical reaction Degradation mechanism Failure type	Combustion, oxidation, evaporation, Corrosion, dissolution, fatigue Leak, rupture, brittle fracture

On 12/17/98, at 0611 hours, Unit 1 was placed in Hot Standby in order to perform repairs on a lube oil cooler for Reactor Coolant Pump (RCP) 1-3. On 12/18/98, during routine inspections of the work area, a buildup of boric acid was discovered. At 1800 hours on 12/18/98, the source of the boric acid was determined to be RCS leakage from the lower radial bearing RTD thermowell on RCP 1-3. In addition, the leakage was determined to be RCS pressure boundary leakage. In accordance with plant Technical Specifications, preparations are being made to place the Unit in Cold Shutdown. The magnitude of the leakage is significantly less than 1 gpm.

Figure 11. Example of identification of nuclear entities.

3.15 Identification of Conjectures

Here we are considering textual elements that contain information about future prediction (e.g., an event that can occur in the future) or hypothesis about past events (e.g., a failure that might have occurred). Even though the reported event has not occurred (or could happen), this evaluation might be relevant for future ER diagnosis (identify possible causes from observed events) or prognostic (identify consequences from observed phenomena) purposes.

In this context, the verb tense plays a role in the identification of this kind of report. Future predictions are characterized by present and future tense verbs; hypotheses about past events are typically characterized by past tense verbs. Also for these kind of reports, we have identified specific keywords (see Table 14) and relations (see Table 15) that can inform our methods that we are dealing with a conjecture observation.

An example of identification of conjectures about past and future events is shown in Figure 12: the full NRC LER 2021-001-00 “Atmospheric Steam Dump Valves Inoperable Due to Relay Failure” has been analyzed and Figure 12 reports three identified conjectures.

Table 14. Examples of keywords that indicate a conjecture observation.

Keyword
Expected
Possible
Probable
Feasible
Plausible
Presumed
Hypothetical(ly)
Likely
Unlikely
Potential
Uncertain
Anticipated
Foreseen
Impending
Upcoming
Brewing
Looming
Forthcoming

Table 15. List of relations that indicate a conjecture observation.

Relation	Example
Subj + "future verb"	The pump will fail
Subj + "conjecture keyword" + "verb"	The pump is likely to fail
Conditional + subj + "verb" + "conjecture keyword" + "verb"	If the pump overheats, it is expected to fail
Subj + "past verb" + hypothesis	The pump failed because it overheated

The capacity of the ASDVs is adequate to prevent lifting of the main steam safety valves following a turbine and reactor trip.

Although the steam dump system is arranged for automatic operation, the ASDVs may be manually controlled from either control room or engineered safeguards control panels.

This action will appropriately prioritize maintenance for the relay and prevent recurrence of this failure.

Figure 12. Example of identification of conjectures identified from NRC LER 2021-001-00 "Atmospheric Steam Dump Valves Inoperable Due to Relay Failure".

3.16 Identification of Health Status

The analysis of a health status report can have different forms. By looking at an initial textual data set, we were able to identify three types of health status reports:

- *Qualitative observation*: the report provides a qualitative observation (e.g., good, degraded, increase, decrease, stable, stop) about an event
- *Quantitative observation*: the report provides a precise observation (i.e., report on point value or delta estimate of a measured variable) of an event (see Section 3.8)
- *Conjecture observation*: the report provides information about a future prediction or hypothesis about the past (see Section 3.10)

In addition, we have identified two attributes that might be contained in health status reports we aim to extract from the raw text: temporal (see Section 3.5) and location attributes.

IRs that report a qualitative observation provide a fairly simple (i.e., qualitative) observation about an event. Table 16 provides a list of relations that have been identified along with ER related examples. As indicated in Table 16, the set of relations are based on specific sets of nouns, adjectives, verbs, and adverbs. Given the nature of these observations, each of these grammatical entities (i.e., nouns, adjectives, verbs, and adverbs) can convey qualitative information: positive, negative, or neutral. In this respect, Table 17, Table 18, and Table 19 provide a subset of grammatical entities for each of the three classes (positive, negative, or neutral).

We applied the developed health identification method to the openly available NRC LER 2021-001-00 "Atmospheric Steam Dump Valves Inoperable Due to Relay Failure"; in this respect, Table 20 lists a subset of health status that have been identified.

Table 16. List of sentence relations for qualitative observation.

Relation	Example
Subj + "status verb"	Pump was not functioning
Subj + "status verb" + "status adjective"	Pump performances were acceptable
Subj + "status verb" + "status adverb" + obj	Pump was partially working
"status adjective" + subj + "status verb"	Unresponsive pump was observed
"status noun" + "prep" + "status verb"	Deterioration of pump impeller was observed

Table 17. Partial list of keywords that indicate negative information.

Nouns	Verbs	Adjectives	Adverbs
Breakdown	Disabled	Unacceptable	Inaccurately
Collapse	Reject	Improper	Erroneously
Decline	Stop	Inadmissible	Wrongly
Deficiency	Block	Undesirable	Inadequately
Deterioration	Halt	Unsatisfactory	Incompletely
Failing	Oppose	Unacceptable	Partially
Decay	Inhibit	Unsuitable	Imperfectly
Downfall	Hinder	Unwanted	

Table 18. Partial list of keywords that indicate positive information.

Nouns	Verbs	Adjectives	Adverbs
Accomplishment	Enable	Ready	Accurately
Achievement	Empower	Fit	Nicely
Enhancement	Facilitate	Capable	Perfectly
Progression	Permit	Apt	Precisely
Solution	Set up	Available	Properly
	Endow	Adequate	Rightly
	Let	Competent	Accurately
	Make	Proficient	Appropriately

Table 19. Partial list of keywords that indicate neutral information.

Nouns	Verbs	Adjectives
Analysis	Inspect	Acceptable
Assessment	Monitor	Usable
Diagnosis	Measure	Attainable
Evaluation	Witness	Consistent
Exploration	Examine	Constant
Investigation	Note	Stable
Probe	Recognize	Unaffected
	View	Uninterrupted
	Watch	Untouched
		Intact

Table 20. Examples of identified health status identified from NRC LER 2021-001-00 "Atmospheric Steam Dump Valves Inoperable Due to Relay Failure".

SSC Entities	Status/Health Status
control room	an acrid odor
steam dump control relay	failed
atmospheric steam dump valves	inoperable
relay	replaced
asdvs	service
control room	an acrid odor
steam dump control relay	failed
atmospheric steam dump valves	inoperable
steam dump and bypass system	four automatically actuated asdvs

3.17 Identification of Cause-Effect Relations

A common pattern in textual ER data is a report of multiple events along with a causal relationship among them. In its simplest form, this paragraph contains an event (i.e., the cause) that triggered a second event (i.e., the effect). However, the structure of this type of paragraph can have different variants (see Figure 13):

- An event that has been identified as not being the cause of another event
- Multiple causes that trigger a single effect
- A single cause that triggers multiple effects

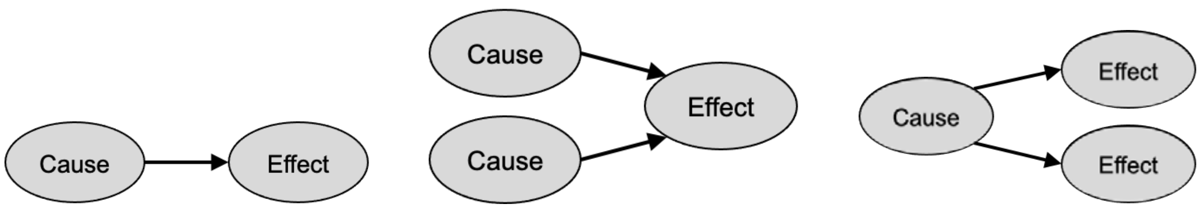

Figure 13. Graphical representation of elemental cause-effect structures: direct cause-effect association (top left), invalid association (top right), multiple causes and single effect association (center left), multiple effects and single cause association (center right), and causal homeostasis.

Here, our methods are not employing ML algorithms (e.g., through classification methods [Mohri and Rostamizadeh, 2012]) but they are rule-based (Doan et al., 2019) since our goal is to extract actual quantitative information from textual data rather than "classifying" the nature of the raw text. These rules are based on the identification of:

- Keywords, such as nouns, verbs, and adverbs, that identify the possibility that the sentence might contain a causal relation between the subject(s) and the object(s) contained in that sentence (see Table 21)
- NLP structures (or constructs) composed of multiple words that indicate a casual transition between clauses contained in a sentence or between sentences (see Table 22).
- Relations between sentence subjects and verbs that are designed to reconstruct the node (see Table 23).

We applied the developed cause-effect identification methods to the openly available NRC LER 2021-001-00 "Atmospheric Steam Dump Valves Inoperable Due to Relay Failure"; in this respect, Figure 14 presents a subset of three cause-effect relations that have been identified. In particular, for each of the three identified relations, Figure 14 shows the original text and details about the relation using the following format:

(cause, status), cause-effect keyword, (effect, status)

Table 21. Partial list of keywords that indicate a cause-effect paragraphs.

Nouns	**Verbs**	**Adverbs**
Augment	Augment	Afterwards
Backfire	Backfire	Consequently

Begin Bring about Build-up Cause Change Combat Compensate Counter Create Deactivate Decelerate Decrease	Begin Bring about Build-up Cause Change Combat Compensate Counter Create Deactivate Decelerate Decrease	Eventually Finally Hence So Subsequently Then Therefore Thus Ultimately

Table 22. List of relations that indicate a cause-effect paragraphs.

Relations	**DAG**
Event_A + "causal verb" (active) + Event_B	A → B
Event_A + "causal verb" (passive) + Event_B	B → A
Event_A + [to be] a "causal noun" + Event_B	A → B
Event_A + [to be] a "effect noun" + Event_B	B → A
The "causal noun" of + Event_A + [to be] + Event_B	B → A
The "effect noun" of + Event_A + [to be] + Event_B	A → B
Clause_A ; + "cause/effect structure" + Clause_B	A → B or B → A
"Cause/effect structure" + Clause_A ; + Clause_B	A → B or B → A
Clause_A . "Cause/effect structure" + Clause_B	A → B or B → A
Event_A + (verb, "causal adverb") + Event_B	A → B

Table 23. List of structures that indicate a cause-effect paragraphs.

Structures
In response to Attributed to As a result of For this reason In consequence In this way In such a way

Investigation revealed that the steam dump control relay had failed, rendering all four atmospheric steam dump valves inoperable.

(investigation,) revealed (steam dump control relay, failed)
(investigation,) rendering (atmospheric steam dump valves, inoperable)
(steam dump control relay, failed) rendering (atmospheric steam dump valves, inoperable)

The opening of the fuse resulted in loss of power to the im13 scheme, which disabled the automatic fast-open function, as well as the manual operation, of the asdvs.

(fuse, the opening) resulted in (im13 scheme, loss of power)

The cause of the sdcr coil failure is overheating due to the age of the relay coil being beyond the vendor recommended life for a normally energized relay.

(relay coil, the age) the cause (sdcr coil, the failure)
(relay, a normally energized) the cause (sdcr coil, the failure)

Figure 14. Example of identification of cause-effect relations (source: NRC LER 2021-001-00 “Atmospheric Steam Dump Valves Inoperable Due to Relay Failure”).

3.17.1 Cause-Effect Paragraphs—Compound Sentence

Here we discuss a case where a single sentence contains multiple clauses (typically two or three) linked together by a causal relationship. An example of a compound sentence with a causal relationship between two clauses is provided in Figure 15. In that example, note that the causal structure “for that reason” is creating a causal relationship between two events contained in two separate clauses. Each clause is then processed using the NLP methods presented in Section 3.15. The NLP analysis workflow designed to extract the causal relationship between clauses contained in a single sentence is shown in Table 24. In addition, Table 24 provides the outcome of each step for the example indicated in Figure 15.

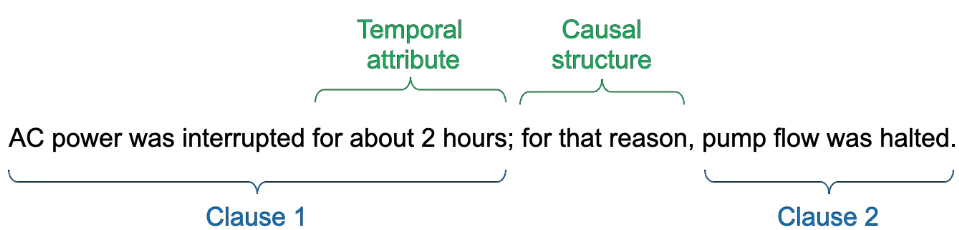

Figure 15. Example of compound sentence containing a causal relationship between two clauses.

Table 24. NLP steps to extract a causal relationship between clauses in a single sentence.

ID	Step	Example (see Figure 15)
1	Identify clauses from sentence	Clause 1 = AC power interruption Clause 2 = pump flow halted
2	Identify transition keywords (see Table 25)	“for that reason”
3	Process each clause (see Section 3.3)	Element_1 = (AC power, degraded) Element_2 = (pump flow, degraded)
4	Create corresponding directed acyclic graph	Element_1 → Element_2

Table 25. List of transition keywords that indicate a causal relationship between clauses in a single sentence.

Structures
Accordingly
Consequently
Hence
On account of
So
As a result
Due to
If … then
Results in
Therefore
Since
Thus
Because of
For that reason
Leads to
As such
It follows that
Thereupon
Ergo
Being that
So that

3.17.2 Cause-Effect Paragraphs—Multiple Sentences

Here we discuss the case where multiple sentences (which might contain multiple clauses) are linked together by a causal relationship. An example of text containing a causal relationship between two sentences is provided in Figure 16. From that example, note that the causal structure "consequently" is creating a causal relationship between two events in two separate sentences. Each sentence is then processed using the NLP methods presented in Section 3.15. The NLP analysis workflow designed to extract a causal relationship between multiple sentences is similar to the one shown in Table 24.

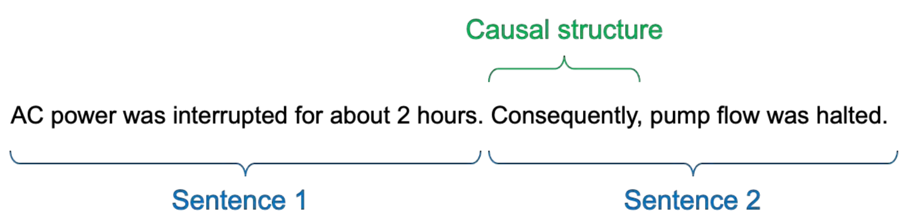

Figure 16. Example of text containing a causal relationship between two sentences.

4 Identification of Text Similarity

The words, sentences and documents similarity analyses are part of NLP, and they play a crucial role in text analytics, such as text summarization and representation, text categorization, and knowledge discovery. There are a wide variety of methodologies that have been proposed during last two decades. Mainly, these techniques can be classified into five groups: 1) lexical knowledge base approach, 2) statistical corpus approach (word co-occurrence), 3) machine learning and deep learning approach, 4) sentence structure-based approach, and 5) hybrid approach. However, there are several common major drawbacks for these approaches: 1) computationally inefficient, 2) lack of automation, and 3) lack of adaptability and flexibility.

In this paper, we are trying to address these drawbacks via developing a tool that can be used generally in applications requiring similarity analysis. As shown in Figure 17, we are trying to leverage POS, disambiguation, lexical database, domain corpus, word embedding and vector similarity, sentence word order, and sentence semantic analysis to calculate sentence similarity. POS is used to parse a sentence and tag each word and token with a POS tag and syntactic dependency (DEP) tag. Such information will provide syntactic structure information (i.e., negation, conjecture, and syntactic dependency) about the sentence that can be used to guide the similarity measuring process.

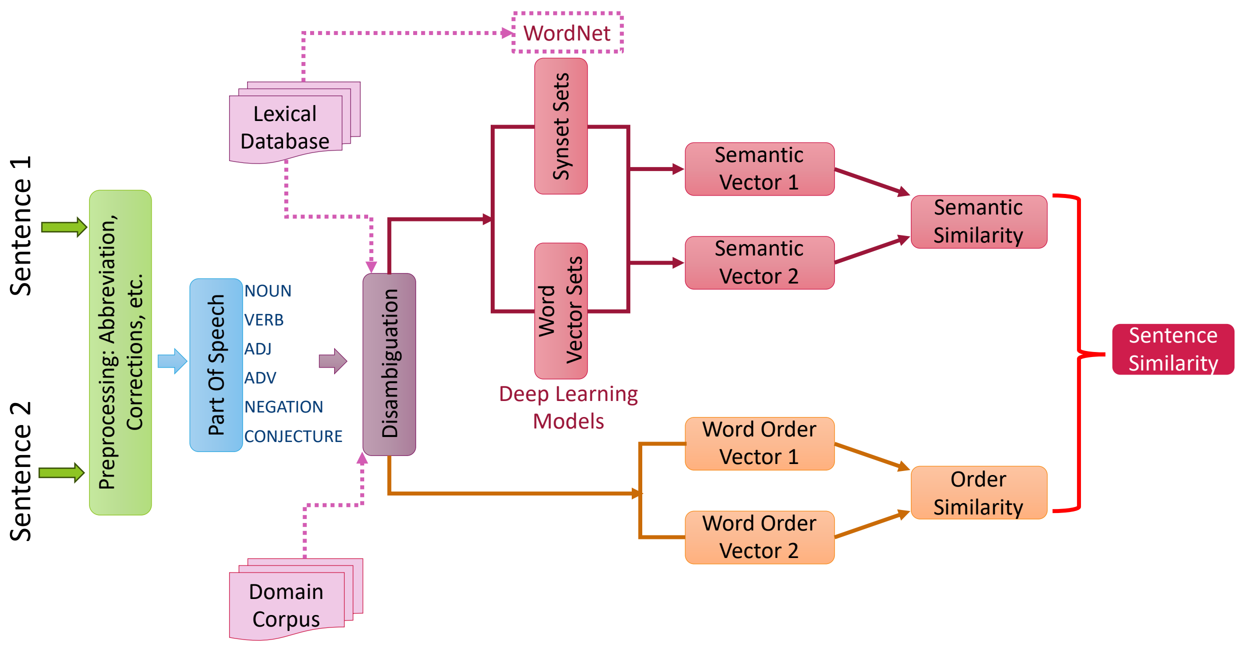

Figure 17. Illustration of sentence similarity calculation.

The disambiguation approach is employed to determine the best sense of the word, especially when coupled with specific domain corpus. It will ensure the right meaning of the words (e.g., the right synsets of the words in a lexical database) among the sentence for comparison. Then, a predefined word hierarchy from a lexical database (i.e., WordNet) is used to compute the word similarity. However, some words are not contained in the lexical database since it only connects four types of POS—nouns, verbs, adjectives, and adverbs. Moreover, these words are grouped separately and do not have interconnections. For instance, nouns and verbs are not interlinked (i.e., the similarity score between “calibration” and “calibrate” is 0.091 when using WordNet). In this case, machine learning based word embedding is introduced to enhance the similarity calculation. For previous example, the similarity score will become 0.715 instead. The next step is to compute sentence similarity by leveraging both sentence semantic information and syntactic structure. The semantic vectors are constructed using the previously introduced word similarity approach, while the syntactic similarity is measured by word order similarity. The following sections further describe each of the steps in more details.

4.1 Part of Speech for Similarity Analysis

As mentioned in Section 3.4 and 3.5, POS provides information about word types and morphological features, and dependency parsing provides dependency syntactic information between words. Utilizing POS and dependency parsing can help to identify the important information, such as NOUN, VERB, ADJ, ADV, negation, conjecture, subject, and object, which will be used to compute the sentence syntactic similarity.

4.2 Lexical Database

Lexical databases, such as WordNet, have semantic connections between words, which can be utilized to determine their semantic similarity. WordNet is a database of lexical information of words originally created by Princeton University. It contains words, their meanings (e.g., synsets) and their semantic relationships, which are stored in a hierarchy tree-like structure via linked synsets. Each synset denotes the precise meaning of a particular word, and its relative location to other synsets can be used to calculate the similarity between them.

As summarized by Navigli (2019), there are many different methods to compute word similarity using WordNet, and sometimes these methods are combined to enhance the similarity calculation. In this work, we employ the method proposed by Li (2006) to compute the similarity score between two words and synsets, as presented in Eq. (1). This method combines the shortest path distance between synsets and the depth of their subsumer (e.g., the relative root node of the compared synsets) in the hierarchy. In other words, the similarity score is higher when the synsets are close to each other in the hierarchy, or their subsumer locates at the lower layer of the hierarchy. This is because the lower layer has more specific features and semantic information, as compared to the upper layer.

$$S_w(w_1, w_2) = f_{length}(l) \cdot g_{depth}(d) = e^{-\alpha l} \cdot \frac{e^{\beta d} - e^{-\beta d}}{e^{\beta d} + e^{-\beta d}} \tag{1}$$

where $\alpha \in [0,1], \beta \in (0, 1]$ are parameters scaling the contribution of the shortest path length and depth, respectively. The optimal values of α and β are dependent on the knowledge base used and can be determined using a set of word pairs with human similarity ratings. For WordNet, the optimal parameters for the proposed measure are $\alpha = 0.2$ and $\beta = 0.45$, as reported by Li (2003).

4.3 Associating Word With the Best Sense (Disambiguation and Domain Specific Corpus)

A sense represents the precise meaning of given word under specific context. Disambiguation is the process to identify the best sense for the word in the context of a statement. Without proper disambiguation, errors could be introduced at the early stage of the similarity calculation when using lexical databases. For example, in WordNet, synsets are used to denote the senses of the word, and they are linked to each other by their explicit semantic relationships. When different synsets are used in calculating word pair similarity, their semantic relationship can be drastically different, which can significantly affect the similarity score. In this work, we try to disambiguate the word sense by considering the context of the word. One way to do it is to take into account the surrounding words since they can provide contextual information. However, this may not work for simple or short sentences. In this case, the domain specific corpus can be leveraged to disambiguate the word. Once the best senses are identified for the words, the word similarity measure from Section 4.2 can be employed.

4.4 Word Embedding or Vector

A word embedding or word vector is typically a numerical vectorization of words or documents. It maps words with semantic similarities to have close embedding vectors. Thus, word embedding can be utilized to measure semantic similarities utilizing the cosine similarity metric between the embedded vectors, especially when WordNet fails in situations such as similarities between words that have different POS tags. In this work, word embedding is leveraged to assist the word similarity calculation. Once the similarity score from the WordNet similarity calculation is below 0.2 (e.g., the two words are not similar), the word embedding similarity calculation will be employed.

4.5 Sentence Similarity

As proposed by Li (2006), sentence similarity contains semantic and syntactic similarity. Semantic similarity is captured via word semantic similarity, as discussed in previous sections, while syntactic similarity is measured by word order similarity. Word order similarity is a way to assess sentences similarity considering word order. As well described by Li (2006), the semantic vectors and word order vectors are constructed, which can be used to compute the sentence similarity. Here we will briefly introduce the methods to construct these vectors and recommend the reader refer to Li (2006) for more details.

Given two sentences, T_1 and T_2, a joint word set is formed (e.g., $T = T_1 \cup T_2$) with all the distinct words from T_1 and T_2. The vectors derived from computing word similarities in $(T, T1)$ and $(T, T2)$ are called the semantic vectors, denoted by s_1 and s_2, respectively. Each entry of the semantic vectors corresponds to the maximum similarity score between a word in T and a word in T_1 or T_2, so the dimension equals the number of words in the joint word set. The semantic similarity between two sentences is defined as the cosine coefficient between two vectors:

$$S_s = \frac{s_1 \cdot s_2}{\|s_1\| \|s_2\|} \tag{2}$$

As proposed by Li (2006), the word order similarity of two sentences is defined as:

$$S_r = 1 - \frac{\|r_1 - r_2\|}{\|r_1 + r_2\|} \tag{3}$$

where the word order vectors r_1 and r_2 are formed from $(T, T1)$ and $(T, T2)$, respectively. For example, for each word w_i in T, the r_1 vector with the same length of T_1 is formed as follows: if the same word is present in T_1, the word index in T_1 is used as the value for r_1. Otherwise, the index of the most similar word in T_1 will be used in r_1. A preset threshold (i.e., 0.4) can also be used to remove spurious word similarities. In this case, the entry of w_i in r_1 is 0.

Both semantic and syntactic information (in terms of word order) play a role in measuring the similarity of sentences. Thus, the overall sentence similarity is defined by Li (2006) as a combination of:

$$S(T_1, T_2) = \delta S_s + (1 - \delta) S_r \tag{4}$$

where $\delta \in (0, 1]$ represents the relative contribution of semantic information to the overall similarity computation.

5 Applications

These examples demonstrate how to preprocess text and identify SSCs and their corresponding health statuses, as well as cause-effect relations between SSCs. In general, text preprocessing is performed manually and is very time-consuming. The developed preprocess methods helps to clean it up. In these examples, we have collected a list of typical examples of IR descriptions (see Table 26) to test the effectiveness of such methods.

In the first example, the extracted SSC entities and their health status are highlighted in blue and yellow in Table 26, respectively. In order to have a better illustration of the extracted data, we have presented the pair of SSC entities and health statuses in Table 27. As we observed, there are two misidentifications highlighted in green. The first one, (*pump, test*), can be easily resolved if we also include the health status keyword "failed" (highlighted in red) in the health status as marked in Table 26. There are two health status options for second one, "found in proximity of rcp" or "oil puddle". In order to determine the right health status for "pump," we employ the word/phrase/sentence similarity (see Section 4) to compute the similarity scores between the SSCs and their potential health statuses. The one with the highest similarity score will be selected as the identified health status. In this case, the similarity score between "puddle" and "pump" is 0.25 while the similarity score between "proximity" and "pump" is 0.027, thus "puddle" with additional information "oil" will be selected as the final health status for "pump."

Table 26. Example of information extraction.

A leak was noticed from the RCP pump 1A. RCP pump 1A pressure gauge was found not operating. RCP pump 1A pressure gauge was found inoperative. RCP pump 1A had signs of past leakage. The Pump is not experiencing enough flow during test. Slight Vibrations is noticed - likely from pump shaft deflection. Pump flow meter was not responding. Rupture of pump bearings caused pump shaft degradation. Rupture of pump bearings caused pump shaft degradation and consequent flow reduction. Power supply has been found burnout. Pump test failed due to power supply failure. Pump inspection revealed excessive impeller degradation. Pump inspection revealed excessive impeller degradation likely due to cavitation. Oil puddle was found in proximity of RCP pump 1A. Anomalous vibrations were observed for RCP pump 1A. Several cracks on pump shaft were observed; they could have caused pump failure within few days. RCP pump 1A was cavitating and vibrating to some degree during test. This is most likely due to low flow conditions rather than mechanical issues. Cavitation was noticed but did not seem severe. The pump shaft vibration appears to be causing the motor to vibrate as well. Pump had noise of cavitation which became faint after OPS bled off the air. Low flow conditions most likely causing cavitation. The pump shaft deflection is causing the safety cage to rattle. The Pump is not experiencing enough flow for the pumps to keep the check valves open during test. Pump shaft made noise. Vibration seems like it is coming from the pump shaft. Visible pump shaft deflection. Pump bearings appear in acceptable condition. Pump made noises - not enough to affect performance. Pump shaft has a slight deflection.

Table 27. Extracted SSC entities and their health status.

SSC Entities	Status/Health Status	SSC Entities	Status/Health Status
Pump	A leak from rcp	Impeller	Excessive degradation
Pump	Not gauge operating	Pump	Found in proximity of rcp (Oil puddle)
Pump	Gauge inoperative	Pump	Anomalous vibrations for 1a
Pump	1a signs of past leakage	Pump shaft	Several cracks
Pump	Not enough flow during test	Pump	Failure
Pump shaft	Deflection	Pump	cavitating
Pump	Not meter responding	Pump shaft	Vibration
Pump bearings	Rupture	Motor	Vibrate
Pump shaft	Degradation	Pump	Noise of cavitation …

Pump bearings	Rupture	Pump shaft	Deflection
Pump shaft	Degradation	Pump	Not enough flow for the pumps
Power supply	Burnout	Pump shaft	Noise
Pump	Test	Pump shaft	Vibration
Pump supply	Failure	Pump shaft	Deflection
Pump	Inspection	Pump bearings	Acceptable condition
Impeller	Excessive degradation	Pump	Noises
Pump	Inspection	Pump shaft	A slight deflection

In the second example, the extracted cause-effect relations between SSCs for the text in Table 26 are presented in Table 28. We employ a set of rule templates based on specific trigger words and relations (see Section 3.17). Once the SSCs entities and their health status has been identified, we can apply these rules to identify the cause-effect relations. There is one cause-effect relation that is not captured because "safety cage" is not listed as the identified SSC entity. This can help us to enhance our knowledge base about SSCs.

Table 28. Causal relations identified.

Text After Rule-Based NER	**Identified Cause-Effect Relations**
Rupture of pump bearings caused pump shaft degradation.	(pump bearings: Rupture) "caused" (pump shaft: degradation)
Rupture of pump bearings caused pump shaft degradation and consequent flow reduction.	(pump bearings: Rupture) "caused" (pump shaft: degradation)
Pump test failed due to power supply failure.	(Pump: test failed) "due to" (power supply: failure)
Pump inspection revealed excessive impeller degradation.	(Pump: inspection) "revealed" (impeller: degradation)
Pump inspection revealed excessive impeller degradation likely due to cavitation.	(Pump: inspection) "revealed" (impeller: degradation)
Several cracks on pump shaft were observed; they could have caused pump failure within few days.	(pump shaft: Several cracks) "caused" (pump: failure)
The pump shaft deflection is causing the safety cage to rattle.	None

The third example focuses on the identification of coreference. This process is tasked with finding the expressions that refer to the same entity in the text. This is particularly relevant where the text includes several sentences and a reference to an entity is not indicated with its proper name but with a pronoun. Through our methods, we can correctly identify the coreferences in the text presented in Table 26 as shown in Table 29.

Table 29. Example of coreference identification.

Coreference Examples	**Identified Coreference**
Several cracks on pump shaft were observed; they could have caused pump failure within few days.	(Several cracks, they)

Vibration seems like it is coming from the pump shaft.	(Vibration, it)

Conjecture means the information provided by the sentence is about future prediction (e.g., an event that can occur in the future) or hypothesis about past events (e.g., a failure that might have occurred). In this context, the verb tense plays a role in the identification of this kind of report. Future predictions are characterized by present and future Table 30 tense verbs; hypotheses about past events are typically characterized by past tense verbs. Based on the text provided in Table 26, the sentences with conjecture information have been correctly identified and listed in Table 30.

Table 30. Identified conjecture sentences.

Pump inspection revealed excessive impeller degradation likely due to cavitation.
Several cracks on pump shaft were observed; they could have caused pump failure within few days.
Vibration seems like it is coming from the pump shaft.

The last example utilizes a manually generated sentence based on an NRC licensee event report to demonstrate the capability of the package on a complex sentence. The text is provide in Table 31, while the results about health status and cause-effect relations are presented in Table 32 and illustrated in Figure 18. In this example, since "investigation" is not among the valid SSC entities and "revealed" is a valid cause-effect keyword, "investigation" is treated as a potential SSC entity and its cause-effect relations are also reported in the outcomes.

Table 31. Sentence based on NRC licensee event report.

Investigation revealed that the steam dump control relay had failed, rendering all four atmospheric steam dump valves inoperable.

Table 32. Identified health status and cause-effect relations for the text listed in Table 31.

(investigation: -) "revealed" (steam pump control relay: failed)
(investigation: -) "rendering" (atmospheric steam dump valves: inoperable)
(steam pump control relay: failed) "rendering" (atmospheric steam dump valves: inoperable)

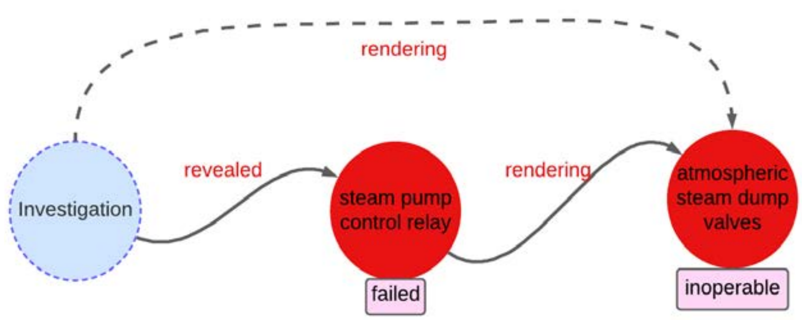

Figure 18. Cause-effect relations for NRC LER example (SSC entity: red cycle, potential entity: blue cycle, health status: pink box, cause-effect relation: black arrow, potential cause-effect relation: dashed black arrow).

The main impact of the created software package on the community is its combination of different methods into one library. Our methods significantly simplify text preprocessing for a variety of NLP tasks in ER data analysis. The package provides a Python implementation of rule-based health status extraction, cause-effect relation identification, and temporal sequencing of events determination. All algorithms in the package are designed to have simple calls with flexible parameters, allowing users with minimal Python experience to process ER data and, by simply combining different pipelines, generate knowledge graphs to assist system engineers with their maintenance activities.

6 Conclusions

This paper has presented an overview of a computational tool designed to extract information from ER textual data. This tool is composed of several methods tasked to parse sentences in search-specific text entities, such as measured quantities, temporal dates, and system, assets, and component IDs. Then semantic analysis tools are designed to capture the semantic meaning of the event(s) described in the provided texts, such as health information, cause-effect relations, or temporal sequence of events. An important element here is the set of preprocessing tools devised to clear textual elements from acronyms, abbreviations, and grammatical errors. Such cleaning methods are essential elements to improve the performance of the knowledge extraction methods.

We have presented few applications that are not limited to the analysis of NPP IRs and WOs. Here, even though the nature of the ER textual elements is short, our tools were able to successfully extract the semantic meaning and identify the great majority of the specified entities. We have also indicated how our sentence similarity measures can be used to parse past outage databases in order to inform the plant outage manager of the historic duration required to complete specific activities. The analysis of NRC reports provided a few good examples on how our methods were able to capture either cause-effect or temporal relations among events.

The capabilities of developed tools are unique in the nuclear arena, and they are based on the parallel development in the medical field. We in fact rely on a few libraries developed for knowledge extraction from medical textual data element (e.g., patients medical reports and doctor diagnosis). The extension of such methods to a different field (i.e., nuclear) required the development of additional methods and libraries to fit new use cases.

References

Doan, S., E. W. Yang, S. S. Tilak, P. W. Li, D. S. Zisook, and M. Torii (2019). "Extracting health-related causality from twitter messages using natural language processing." BMC Medical Informatics and Decision Making, vol. 19, no. 3, pp. 71–8. https://doi.org/10.1186/s12911-019-0785-0.

Honnibal, M., and M. Johnson (2014). "Joint Incremental Disfluency Detection and Dependency Parsing." Transactions of the Association for Computational Linguistics, vol. 2: pp. 131–142. https://doi.org/10.1162/tacl_a_00171.

Li, Yuhua, Z. A. Bandar, and D. McLean (2003). "An approach for measuring semantic similarity between words using multiple information sources." IEEE Transactions on knowledge and data engineering vol. 15, no. 4, pp. 871-882. https://doi.org/10.1109/TKDE.2003.1209005.

Li, Yuhua, et al. (2006). "Sentence similarity based on semantic nets and corpus statistics." IEEE transactions on knowledge and data engineering, vol. 18, no. 8, pp. 1138-1150. https://doi:10.1109/TKDE.2006.130.

Moerchen, F. (2010). “Temporal Pattern Mining in Symbolic Time Point and Time Interval Data”. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD’10. https://doi.org/10.1145/1835804.1866289.

Mohri, M., A. Rostamizadeh, and A. Talwalkar (2012). *Foundations of Machine Learning*. Cambridge, MA: The MIT Press.

Navigli, R., and F. Martelli (2019). “An Overview of Word and Sense Similarity.” Natural Language Engineering, vol. 25, no. 6, pp. 693-714. https://doi:10.1017/S1351324919000305.

Xingang, Z., J. Kim, K. Warns, X. Wang, P. Ramuhalli, S. Cetiner, H. G. Kang, and M. Golay (2021). “Prognostics and Health Management in Nuclear Power Plants: An Updated Method-Centric Review with Special Focus on Data-Driven Methods.” Frontiers in Energy Research 9: 696785. https://doi.org/10.3389/fenrg.2021.696785.

Young, T., H. Devamanyu, S. Poria, and E. Cambria (2018). “Recent Trends in Deep Learning Based Natural Language Processing.” IEEE Computational Intelligence Magazine, vol. 13, no. 3, pp. 55–75. https://doi.org/10.1109/MCI.2018.2840738.

APPENDIX B

Correlating Equipment Reliability Events Through a Knowledge Graph

Correlating Equipment Reliability Events Through a Knowledge Graph

D. Mandelli[a*], C. Wang[a], J. Cogliati[a], and V. Agarwal[a]

[a] Idaho National Laboratory, Idaho Falls, ID

* Corresponding Author: diego.mandelli@inl.gov

ABSTRACT

Nuclear power plants collect large amounts of equipment reliability data elements that contain information on the statuses of component, assets, and systems. Such data may take the form of online monitoring data (e.g., pump vibration data or pump mass flow rate), data from surveillance and testing performed by plant operators at regular intervals, condition reports (which typically contain anomalous conditions), and maintenance reports (which indicate operations performed to restore component or asset health). All these data elements precisely record asset and system performance and health throughout the lifecycle of those assets and systems. In addition, such data carry the potential to provide system engineers with insights into anomalous behaviors or degradation trends as well as the possible causes behind them and to predict their direct consequences. However, several challenges have proved to be roadblocks to this process. While some of these challenges are technical in nature (i.e., data are often distributed over several physical servers or databases), others are conceptual in nature (i.e., data elements come in different formats, numeric or textual), and measured values have different scales (e.g., vibration spectra and oil temperature). The present study directly tackles the last two issues, focusing on the integration of numeric and textual data elements in order to assist plant system engineers in analyzing equipment reliability data. This task begins with preprocessing the data by extracting knowledge from textual data via natural language processing methods and quantifying system, asset, and component health based on numeric data. We then employed model-based system engineering (MBSE) models of systems and assets to identify their architecture and functional (i.e., cause and effect) relations. These models were translated into graph structures in which each element of the graph represents either the "form" of a system, asset, and component or their supporting "function." Data elements were then associated with a single MBSE graph element, based on their nature. This bonding of MBSE models and data elements constitutes a first-of-its-kind knowledge graph of a nuclear power plants system, with data elements being organized in a structured manner that enables system engineers to identify cause-effect trends in data elements and carry out appropriate actions in response.

Keywords: Reliability, Natural Language Processing, Data Fusion, Model-Based System Engineering

List of acronyms

CWS	circulating water system
ER	equipment reliability
IR	incident report
MBSE	model-based system engineering
ML	machine learning
NLP	natural language processing
NPP	nuclear power plants
OPM	object-process methodology
PHM	prognostic and health management
UML	Unified Modeling Language
SysML	Systems Modeling Language
WO	work order

1 Introduction

Complex systems such as nuclear power plants (NPPs) generate large amounts of equipment reliability (ER) data regarding the performance of their components and assets. Such data can be used to optimize system operation by tracking component and asset degradation and scheduling maintenance operations. In an NPP context, ER data typically take different forms depending on the given component or asset and its operational setting. For example, online monitoring data (e.g., pump vibration data or pump mass flow rate) can provide real-time insights into the degraded performance and possible failure conditions of assets that are in continuous operation. Assets that are normally in a standby configuration (e.g., safety-related assets only employed under accident or abnormal conditions) regularly undergo surveillance and testing by plant operators. Once degradation in a component or asset is identified and deemed detrimental to its function(s), it is logged in an incident report (IR) so that maintenance operations can be performed to restore its health. The maintenance operations are then logged in maintenance reports or work orders (WOs). Note that these ER data elements afford different types of insights into the reliability performance of the given asset. Section 2 provides a taxonomy for the aforementioned types of ER data, showing the functional relationships among them.

All ER data elements carry the potential to provide system engineers with insights into anomalous behaviors or degradation phenomena and to inform them of the optimal restoration strategy. However, several challenges exist as roadblocks to this process. While some of these challenges are technical (e.g., data are often distributed over several physical servers or databases), others are conceptual in nature: data elements come in different formats (e.g., numeric and textual), and measured values have different scales (e.g., vibration spectra and oil temperature).

The present study directly tackles the last two issues, focusing on the integration of numeric and textual data elements in order to aid plant system engineers in analyzing ER data. This task begins with preprocessing the ER data. To preprocess textual ER data, we employed knowledge extraction methods based on natural language processing (NLP) algorithms (Lane, 2019). Meanwhile, we employed a margin-based approach to quantify system, asset, and component health based on numeric data. We then employed model-based system engineering (MBSE) (Borky, 2018) models of systems and assets to identify their architectures and functional relations. These models were translated into graph structures, with each element representing either the "form" of a system, asset, and component, or its supporting "function." The ER data elements (both textual and numeric) were then associated with a single MBSE graph entity, based on their nature. We then employed logic rules and temporal analysis algorithms to determine whether ER data elements were linked to each other via a cause-effect relation. The resulting knowledge graph reflects a structured, complete, and organized architecture for analyzing and visualizing complex ER datasets and effectively provides a useful overview of the ER performance of systems and assets.

In this paper, we show how such a knowledge graph can be constructed using the ER data generated from a circulating water system (CWS) of an existing pressurized-water reactor (Agarwal, 2021a). Throughout this paper, *system* indicates a collection of assets designed to carry out a specific function (e.g., to generate alternating current power or provide high-pressure injection during a loss-of-coolant accident). *Asset* indicates a system element designed to support the system function (e.g., a diesel generator, motor-operated valve, or centrifugal pump). *Component* denotes an asset subelement that is subject to degradation and aging and may require maintenance (e.g., a transmission gear in a diesel generator, the drive sleeve of a motor-operated valve, or the impeller of a centrifugal pump) to guarantee proper asset operation.

2 Equipment Reliability Data Taxonomy

As indicated in Section 1, NPP ER data can be heterogenous in nature (e.g., numeric, textual, or image data). Understanding and capturing the relationships among ER data elements requires a data categorization process. The categorization of each ER data element is not unique and could be context dependent. For the scope of this article, we performed such categorization based on a cause-effect lens (see Figure 1). More specifically, generic assets can be broken down into two elements: its form (i.e., the actual physical entity) and its function[1] (i.e., the emergence property [Borky, 2018]).

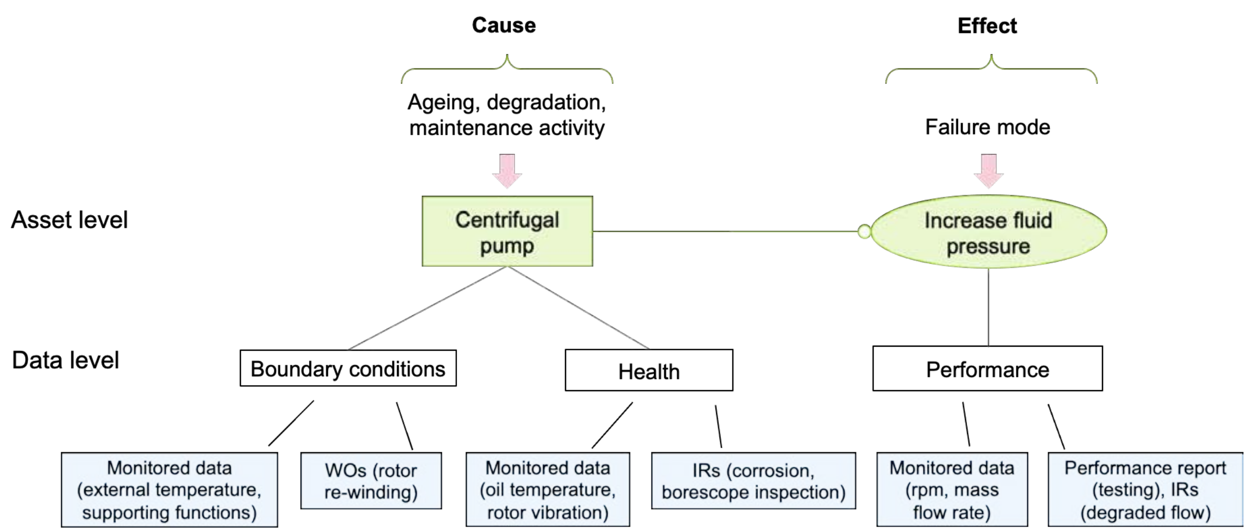

Figure 1. Taxonomy of ER data presented through a cause-effect lens.

For example, when considering a centrifugal pump (see Figure 1), the form element consists of all the components that make up the considered asset (e.g., motor, stator, shaft, and impeller), and the function element indicates its function (i.e., increase fluid pressure). From a reliability standpoint, an asset failure is typically defined in terms of a loss of a function. Aging and degradation (e.g., flow-accelerated corrosion) directly affects the asset form, potentially having a direct impact on its function (e.g., asset failure). Per Figure 1, data associated with either a form or function standpoint can be textual (e.g., WOs) or numeric (e.g., environmental temperature). ER data retrieved from the form node are portioned into two groups—

[1] In many situations, an asset might be supporting multiple functions and might consist of several parts or components that either support or do not support each of these functions depending on the asset architecture. The proposed discussion can be easily extended to these situations.

health monitoring and boundary condition monitoring—and can be either numeric or textual as well. Note here that maintenance operations designed to restore the asset's intended function or form (either by replacement refurbishment or restoration) directly impact asset form, which consequently may affect its function. The objective of this work is to capture the causal relations between ER numeric and textual data elements in order to assist system engineers on the identification of anomalous behaviors.

3 Considered Example: Circulating Water System (CWS)

3.1 System Description and Available Monitoring Data

The CWS system of an existing U.S. nuclear plant site was selected as the target for our analysis methods. The CWS is an important non-safety-related system. As the heat sink for the main steam turbine and associated auxiliaries, the CWS is designed to maximize steam power cycle efficiency (Agarwal, 2021a; Agarwal, 2021b). A CWS consists of the following major equipment (Agarwal, 2021a; Agarwal, 2021b):

- Vertical, motor-driven circulating water pumps (CWPs), each with an associated fixed trash rack and traveling screen at the pump intake to filter out debris and marine life
- Main condenser
- Condenser waterbox air removal system
- Circulating water sampling system
- Screen wash system
- Necessary piping, valves, and instrumentation and controls to support system operation.

The selected plant site (a two-unit pressurized-water reactor) features six circulators at each unit. Schematic representations of the main condensers for Plant Site Unit 2 are shown in Figure 2. Each pair of waterboxes is named using the following convention: Unit #, Condenser #A and Unit #, Condenser #B. In this research, the project team focused on optimizing the maintenance strategy for the CWS. To differentiate between motor and pump maintenance activities for each circulator, those assets are hereafter referred to as the CWP motor and the CWP, respectively. Figure 3 shows different locations on the CWP motor where measurements are continuously collected as part of the plant historian.

The Unit 1 and Unit 2 CWS process data are collected once per minute and stored in the Plant Site 1 historian system. Due to file size restrictions, the project team received hourly CWS process data for both units, from 2009 to 2019. The process data include:

- Gross load (MWe)
- River level (ft)
- Ambient air temperature (°F)
- CWP inlet river temperature (°F)
- CWP outlet water temperature (°F)
- CWP motor status (ON or OFF)
- CWP motor stator winding temperature (°F)
- CWP motor inboard-bearing temperature (°F)
- CWP motor outboard-bearing temperature (°F)

- CWP motor current (amps).

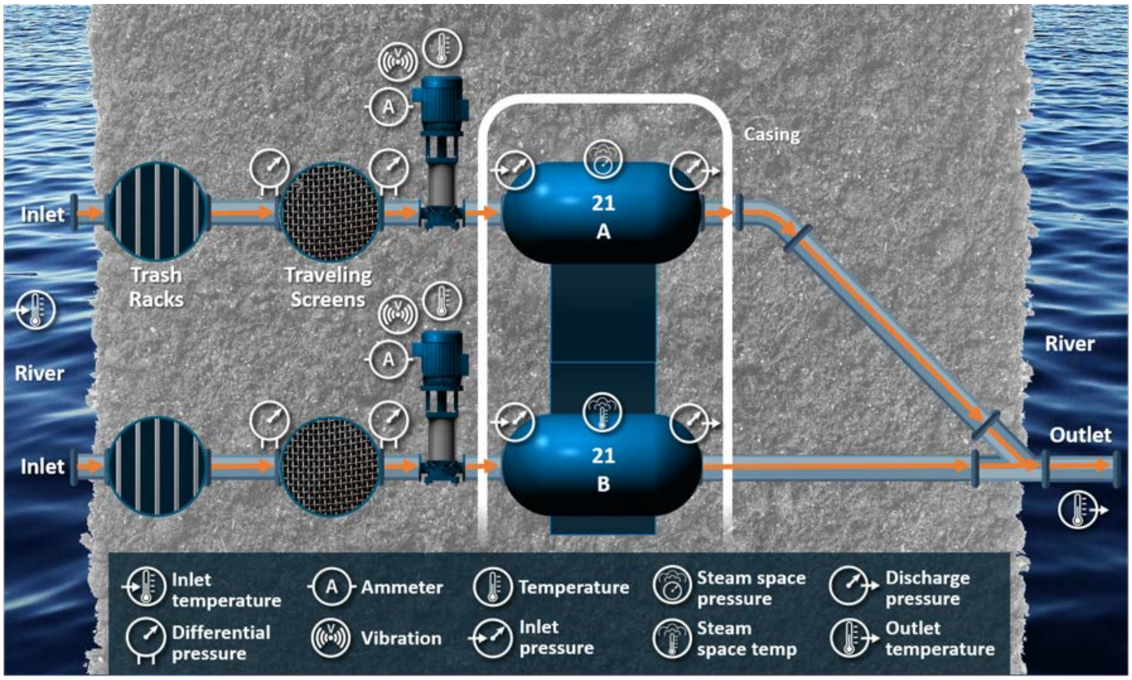

Figure 2. Plant Site Unit 2 CWP combination of 21A and 21B, with sensors and instrumentation.

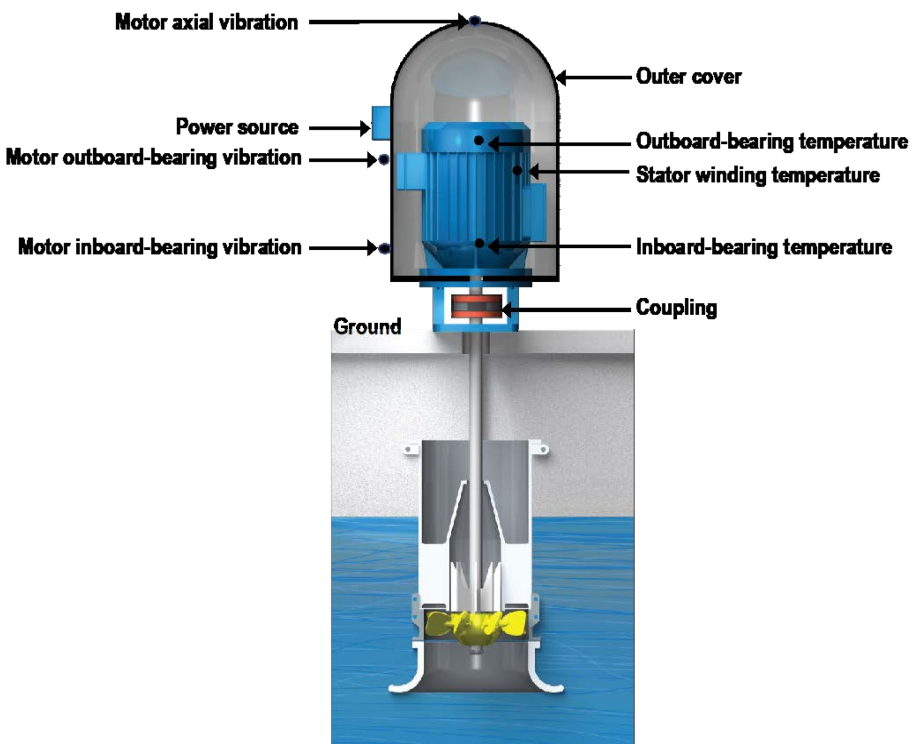

Figure 3. Schematic representation of a CWS motor and pump, along with measurement locations.

3.2 CWS Textual Data

WO data is available from plant Asset Suite Enterprise Management System (EMS) in order to make informed decisions about the historical maintenance performed NPP assets (including the CWS system). Utility information varies in completeness and accuracy across different customers and within individual plants. This variation in data quality is related to many factors, including but not limited to the criticality of work, age of the EMS, number of switches between different EMSs, site procedures, and a utility's ability to extract data. Typically, NPP personnel lack the time or resources for extracting the proper information and performing detailed analytics on their own systems.

The provided database consists of about 2,000 IR and WO data records related to the CWS system, which reports the following data elements: task type, event description, and actual start and completion date (if available). A few relevant observations to be highlighted here are:

- The event description provided in the available set of IRs and WOs are usually very short and not well posed from a grammatical standpoint, i.e., they are not following the standard structure:

  Subject + Predicate + Object + Complements

  This poses several challenges since most NLP methods perform much better with well-formed sentences.
- Often words contained in IRs and WOs might contain typos or be reduced in form (i.e., abbreviated); as an example "management" abbreviated into "mngmt".
- Acronyms and component IDs are often used to indicate specific entities such as systems (e.g., ACC to indicate the accumulator system), assets (e.g., the ID "ACC_pump_001A" indicates the specific pump 001 of train A of the accumulator system), or context-specific elements (e.g., NDE to refer to non-destructive evaluation).

In this respect, Section 5 provides an overview of methods developed to overcome the challenges listed in this section in order to improve the semantic analysis of the provided IRs and WOs.

3.3 CWS Numeric Data

As indicated by (Agarwal, 2021a), the raw data collected from the CWS system are distributed over several data sources and processed by completing the following steps:

1. Logic/binary data elements are converted into numeric form (e.g., the ON/OFF data element is converted into 0/1)
2. New features are generated (e.g., pump differential temperature [DT], pump age since refurbishment)
3. Pump vibration data are processed through a fast Fourier transform algorithm, and the magnitude of the vibration signal for specific frequencies is captured
4. Based on the system operational history (e.g., maintenance records), data elements are labeled (as pertaining to either healthy or faulty state)
5. Missing data entries are resolved
6. Data conflicts between the operational history and recorded numerical values are resolved
7. All data sources are merged into a single time series

8. All features of the time are Z-normalized as each feature x is transformed into $\tilde{x} = \frac{x - mean(x)}{std\_dev(x)}$, where the operators $mean(x)$ and $std\_dev(x)$ correspond to mean value and the standard deviation of the considered variable x, respectively.

The general notation used throughout this paper is that a single observation data element ξ^{obs} is composed of L observed features x_l $(l = 1, \dots, L)$ (i.e., $\xi^{obs} = [x_1, .., x_L]$) and the nature of the observed variables x_l can be heterogenous in nature (e.g., temperature, pressure). A series of plots of the preprocessed time series for the considered five CWS features is shown in Figure 4.

An important element here is that portions of the provided numeric data set has been labeled based on historic system condition. More specifically, five CWS failure modes have been identified:

- Waterbox fouling
- CWP misalignment
- CWP diffuser failure
- Bellmouth fail
- Clogging in air intake screens

Each of these faults has a distinct signature that can be automatically extracted from the CWS process data and vibration data to streamline the diagnosis process. At a minimum, a fault signature is comprised of an asset type, fault type, and a set of one or more observable features that may indicate the occurrence of the associated fault. $\Xi^{obs-healthy}$ indicates the portions of the datasets under healthy conditions while Ξ^{obs-FM_r} $(r = 1, \dots, R)$ indicates the portions of the datasets under faulty conditions for each of the considered five (i.e.,., $R = 5$) failure modes r. Such detection capability is also enabled by the availability of a portion of the numeric data under healthy conditions (which is indicated as $\Xi^{obs-healthy}$). It is here assumed that in the presence of an asset fault, the actual observed data ξ^{obs} can be seen to transition from $\Xi^{obs-healthy}$ to $\Xi^{obs-faulty}$.

4 Digital Representation of Systems and Assets

From a reliability standpoint, it is vital to identify the causal relationships among ER data, maintenance activities, and failure modes. This is typically neglected in the state of practice in current ER data analysis methods based on ML methods. This limitation is due to the fact that data are not enough to identify such causal relationships. Instead, system models are needed to perform such discovery.

In this respect, MBSE (Borky, 2018) methods afford several solutions for modelling systems, assets, and components from both a *form* (i.e., which elements are part of the structures, systems, and components) and a *functional* (i.e., how systems and assets interact with each other, and which functions they support) standpoint. These solutions are based on MBSE languages that represent system and asset form and functional elements via a set of diagrams. The most commonly used languages are Object-Process Methodology (OPM) (Dori, 2002), Unified Modelling Language (William, 2004), and Systems Modelling Language (Friedenthal, 2008).

For the scope of this project, we have chosen OPM since it provides the basic modelling elements we sought, and because—more importantly—digital data structures (i.e., graphs) can be automatically generated from OPM diagrams. Each element of an OPM diagram can be either a *function* (e.g., an action or a transformation) or *form* (e.g., a physical entity) element. In addition, function and from elements in an OPM diagram are connected to each other through a set of *links* designed to convey precise meanings (Dori, 2002). The OPM diagram shown in Figure 5 provides a simple representation of a centrifugal pump which includes the most basic elements found in most OPM models that are used in this paper; in detail, this diagram provides the following information:

- The form element "centrifugal pump" is composed by (through the composition link) four elements: shaft, impeller, bearing, and motor.
- The function "increase fluid pressure" requires the form element "centrifugal pump" (through the instrument link).
- The function "increase fluid pressure" transforms "fluid pressure" from low to high (through the transformation link).
- "Fluid pressure" is an attribute of the form element "fluid" (through the characterization link) that pump is affecting.

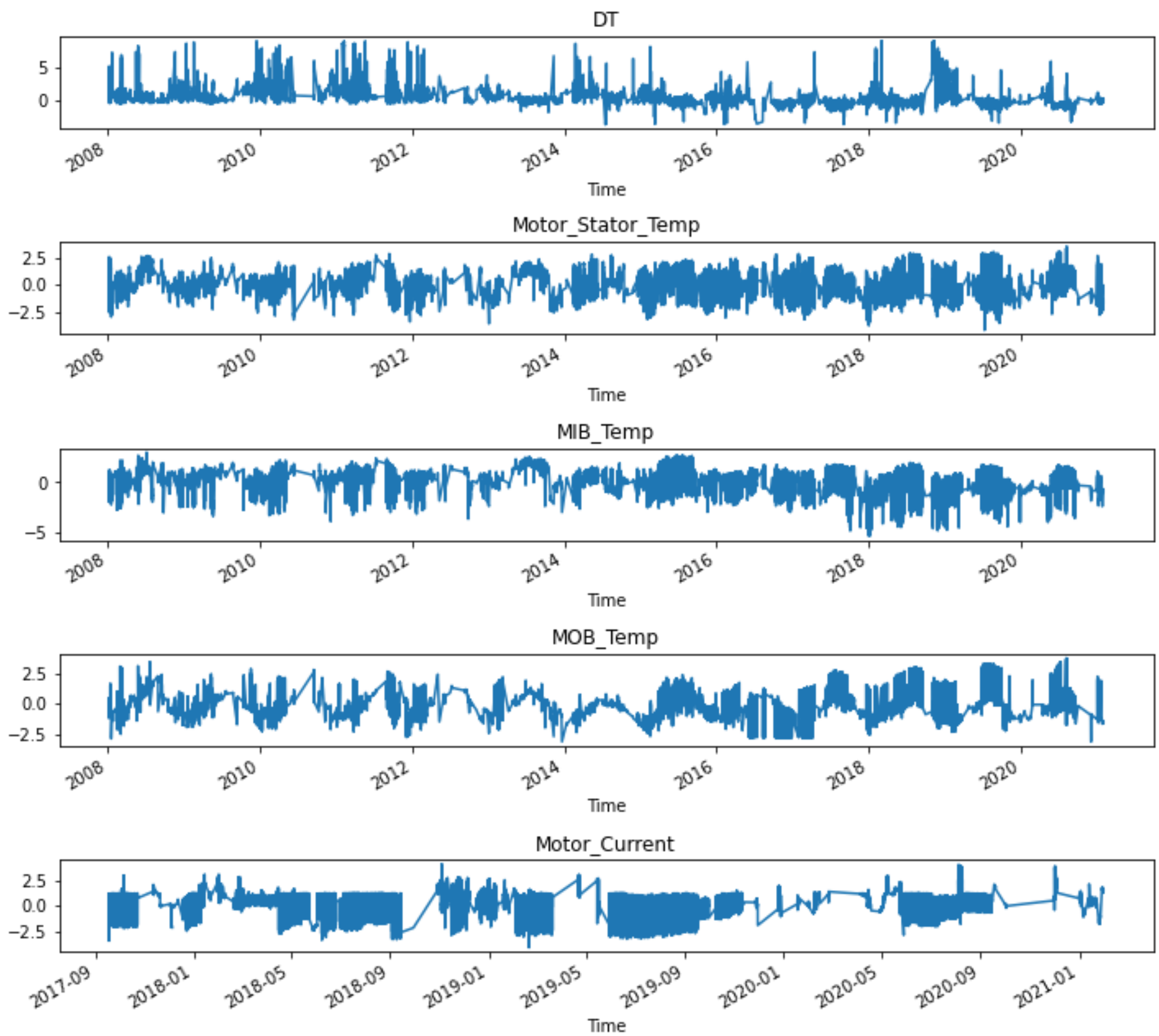

Figure 4. Plot of five features of the preprocessed time series. Note that online motor current data are available from 2017, whereas process variables are available from 2009.

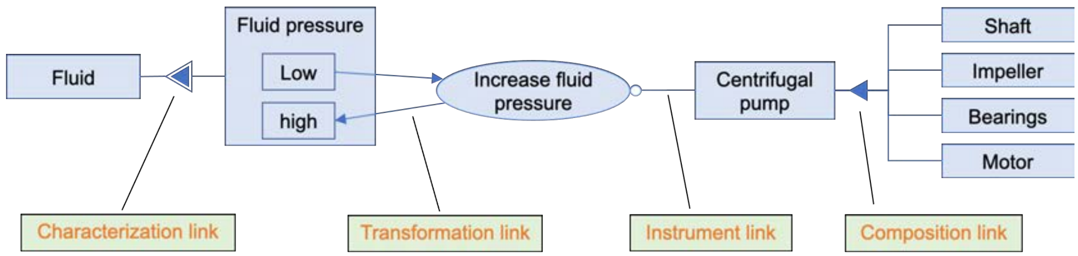

Figure 5. Simplified representation of a centrifugal pump using OPM.

Figure 6 shows the OPM diagram of the considered CWS (Agarwal, 2021a), which includes assets such as trash rakes, waterboxes, and ice barriers. Note that each asset included in the OPM diagram of the CWS may be further described by its own, separate OPM diagram. In other words, a network of OPM diagrams can be constructed to refine and further detail the architecture of the considered system. For example, in the CWS OPM diagram in Figure 6, the centrifugal pumps are indicated as pertaining to a different OPM diagram that represents the pump architecture in greater detail. The corresponding OPM diagram for the centrifugal pump is shown in Figure 7.

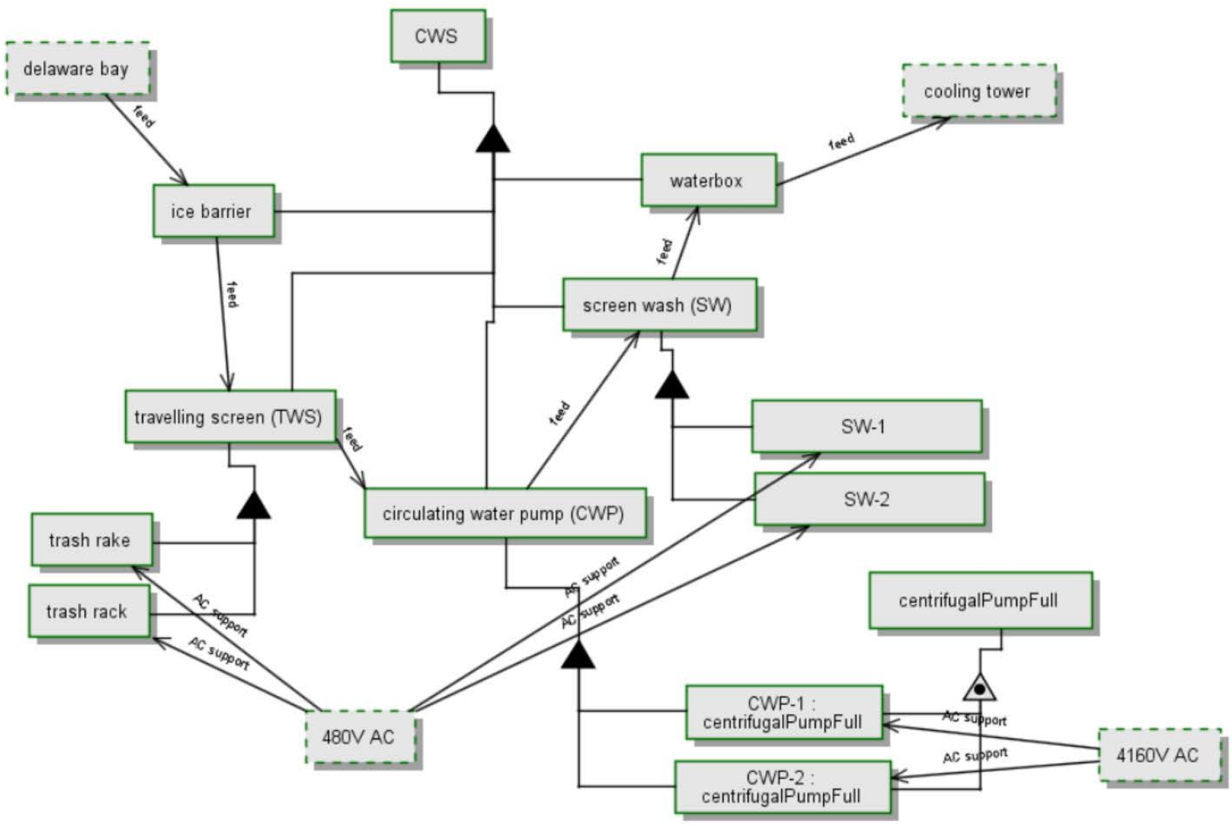

Figure 6. OPM representation of the CWS. Complex assets (e.g., centrifugal pump) are indicated as corresponding to another OPM diagram.

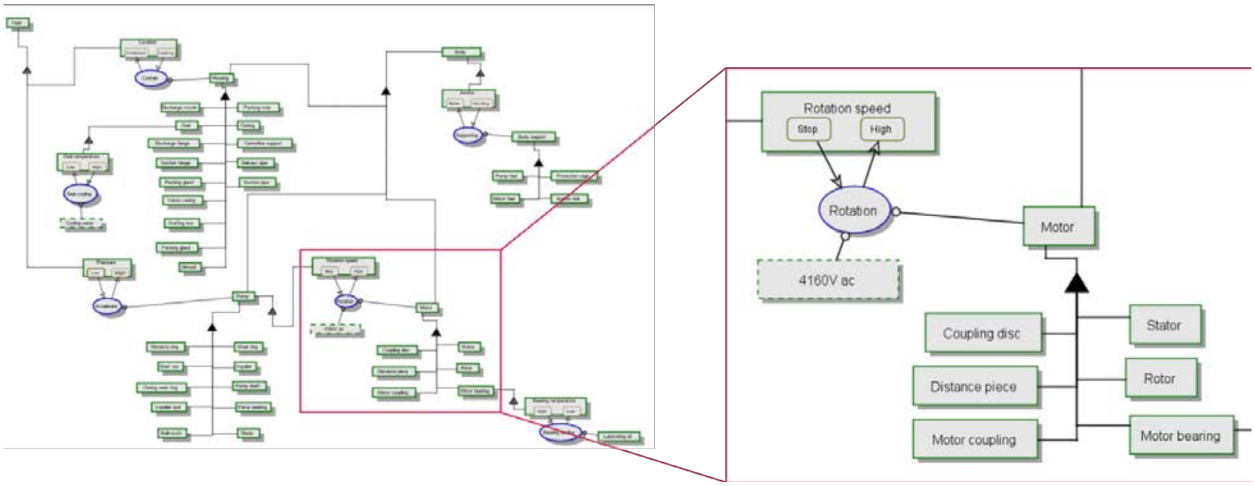

Figure 7. OPM representation of a CWS centrifugal pump (left) which is indicated as CWP-1 and CWP-2 in the CWS OPM diagram (see Figure 6). The portion highlighted in red is expanded (right).

For the scope of the present work, each OPM diagram was translated into a graph structure by using the `NetworkX`[2] library, a database for capturing the different types of OPM elements and edge. The resulting graph made it possible to query all the textual IDs that describe each form or functional element of the diagram. These textual IDs can be employed in our NLP methods to identify and recognize each element based on a given set of sentences (see Section 4). More importantly, if multiple textual IDs are identified, the directed graph can identify the links between them. Note that such links are implicitly causal in nature. In summary, the obtained graph structure is used to accomplish three main objectives: identify OPM entities mentioned in ER textual data elements, identify OPM entities that refer to ER numeric data elements, and identify logical connections between the ER data elements by determining whether a direct causal path exists between the OPM entities associated with each ER data element.

Note that the OPM diagrams can be used for more than simply putting "text into context" via the NLP methods described in Section 5. These diagrams can in fact be employed to identify which OPM elements are being monitored by condition-based, diagnostic, and prognostic systems (e.g., pump shaft vibrations or the rate of mass flow exiting the pump itself) (see Section 5).

5 Analysis of Textual Data

IRs and WOs are valuable data sources for tracking asset health histories, identifying health trends, and performing root-cause analyses. These data sources, typically obtained as text, are usually available in digital repositories. Methods have been developed over the past two decades to enable machine learning (ML) models to analyze textual data and classify textual elements based on their nature (e.g., safety-related vs. non-safety-related natures). In the context of the present work, we are not interested in solving any type of classification problem, but rather in extracting actual knowledge from textual data. This is a harder task, as it requires the development of context-dependent models and vocabularies. The medical field is leading the way in this area by developing methods to extract knowledge from textual data (e.g., for diagnostic purposes or to estimate the performance of specific treatments).

When applied to the nuclear field, knowledge extraction consists of several tasks, including the identification of:

[2] Official website: https://networkx.org/

- Plant-specific entities, such as systems, assets, and components (e.g., centrifugal pump, accumulator system, and pump shaft)
- Temporal attributes that characterize events (e.g., the occurrence, duration, and order of events)
- Phenomena (e.g., material degradation or asset functional failure)
- Causal relations between events.

This process of knowledge extraction is enabled by a series of data, models, and methods. The developed series of methods was designed to identify all the elements listed above, using a mixture of rule-based and ML algorithms. These methods heavily rely on data dictionaries and plant, system, and asset models. Data dictionaries containing a large number of keywords related to the nuclear field were partitioned into several classes (e.g., materials, chemical elements and compounds, degradation phenomena, and electrical, hydraulic, and mechanical components).

The ability of system engineers to analyze textual data is enabled by their knowledge of the architectural scheme of the components and assets that comprise the system. In simpler terms, they know what physical elements comprise a given asset or system, along with their functional relations and dependencies. Without such information, knowledge extraction from textual data is very difficult, as putting the text into context becomes much harder. For the present study, our methods were designed to check whether OPM entities (see Section 4) are mentioned in ER textual data elements.

In more detail, the textual data elements described in Section 3.2 were processed by performing the following steps:

Step 1: Data preprocessing. Here, we perform a series of data cleaning methods designed to improve the clarity of the data elements, including:

a. Text cleaning (e.g., lower all characters, remove punctuation)
b. Identify acronyms
c. Identify and complete abbreviations
d. Spell checking
e. Converting numerical words and quantitative named entities into numeric strings

Step 2: Syntactic analysis. Here, ML methods are used to parse the textual data elements from a syntactic point of view. More specifically, each element is first partitioned into sentences and then part-of-speech methods (e.g., identifying nouns, verbs, adjectives), dependency parsing (i.e., identifying subjects, predicates), and coreference resolution (i.e., identifying textual elements referring to pronouns) methods are employed.

Step 3: Semantic analysis. Here, rule-based methods are employed to identify the following elements from each sentence:

a. Systems, assets, or components that are specific to the considered plant that have been modeled though OPM diagrams
b. Nuclear specific entities and keywords (e.g., degradation phenomena, chemical elements and compounds, or generic electric, mechanical, and hydraulic components)
c. Measured quantities (e.g., pump oil temperature) and temporal occurrence of events (e.g., date and hour)
d. The health nature of a reported event

e. The temporal or location attributes

f. Cause-effect relation between events

g. Conjectures (i.e., events that might have occurred in the past or might occur in the future)

Figure 8 provides an example of knowledge extraction from an ER textual data element. Based on the developed libraries, the asset (i.e., pump) and reactions (i.e., cracking and failure) mentioned in the text are identified, along with a specific pump OPM entity (i.e., shaft). Furthermore, additional elements are captured: the existence of a conjecture and the temporal attribute associated with pump failure.

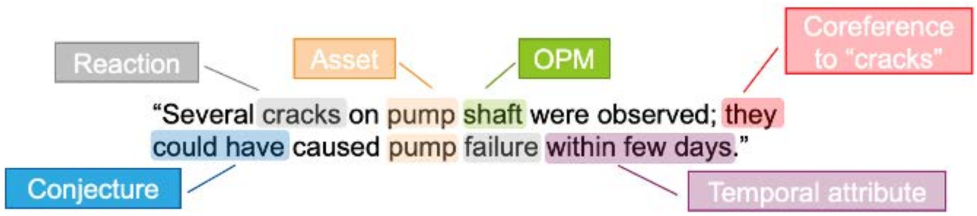

Figure 8. Example of NLP knowledge extraction from an ER textual data element.

6 Analysis of Numeric Data

ER data generated in a numeric format are very common in existing NPPs, in which many assets are continuously monitored (e.g., vibration data, oil temperature, and outlet water pressure) via advanced monitoring and prognostic and health management (PHM) systems in order to identify data trends that may inform system engineers of degraded performance or failure of the considered asset. One challenge with complex systems (not only nuclear) is quantifying the health of each asset, then propagating the assets' health values to the system level. References (Mandelli 2023a; 2023b) provide a margin-based approach that addresses these challenges. Such an approach enables the data generated by condition-assessment, diagnostic, and prognostic systems to be converted into margin values that serve as a quantitative measure of asset health.

An asset's margin value M is defined over the $[0,1]$ interval, where $M = 1$ corresponds to a perfectly healthy asset (requiring minimal to no maintenance attention) and $M = 0$ corresponds to a faulty asset (requiring maintenance attention). Note that margin quantification (see Figure 9) is impacted by the availability of monitoring data and can be defined over heterogenous variables, such as pressure, vibration spectra, and time. For example, when dealing with condition-based monitoring data (both current and archived), margin M is defined here as the distance between actual and past conditions (e.g., oil temperature and vibration spectrum) that lead to failure. Hence, margin-based reliability modeling provides a unified approach to dealing with heterogeneous monitoring data elements.

The margin value of an asset is not static but changes with time, depending on asset conditions. For example, if degradation due to usage is observed from the monitoring data, the corresponding asset margin value decreases. Conversely, if a maintenance operation is performed on that same asset (e.g., restoration of centrifugal pump bearings), the asset margin value increases. This mindset shift regarding the concept of reliability (i.e., margin based instead of probability based) offers the advantage of directly linking the asset health evaluation process with standard plant processes for managing plant performance (e.g., plant maintenance operations and budgeting processes). The transformation also supports decision-making in a form that is more familiar and readily understandable to plant system engineers and decision makers.

So far, margin has been defined for one single asset; the next step is to quantify the system's margin value after obtaining the margin values of its assets. The propagation of margin values from the asset level to the system level is performed through classical reliability models, such as fault trees (FTs) or reliability block diagrams (Lee, 2011), which are solved using different rule sets (Mandelli, 2023a) instead of set theory-based operations.

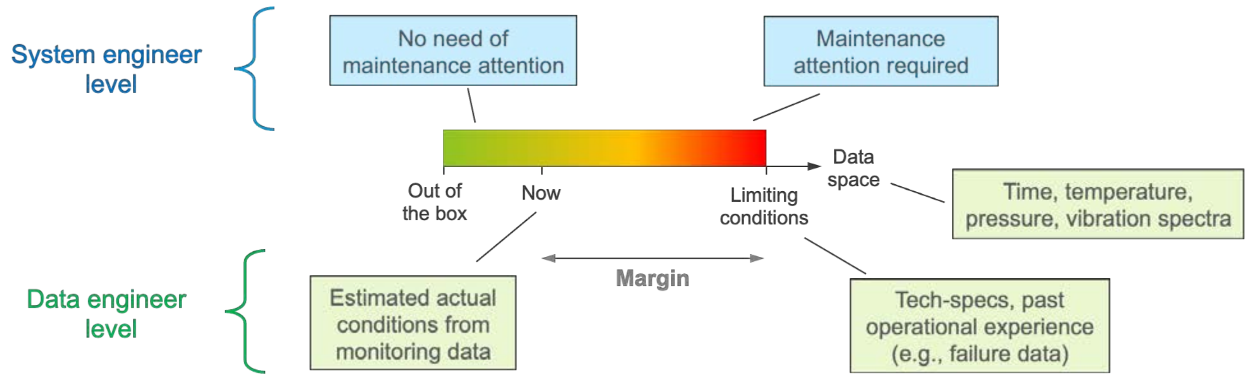

Figure 9. Graphical representation of margin (Mandelli, 2023b).

For the scope of the present article, we are interested in tracking the health history of an asset (if ER data are available, of course) and exploring how reported events (e.g., IRs) might causally relate to health trends. The advantage of margin-based approaches is that complex monitoring data generated by condition-assessment, diagnostic, and prognostic systems can be integrated into a common quantitative health measure.

Here, each numeric data element (i.e., margin temporal profile of a specific monitored variable) can be matched to an exact OPM entity. The example taken from (Mandelli, 2023b) and reflected in Figure 11 shows the margin temporal profile for the CWS waterbox. As indicated by (Agarwal, 2021a) and (Agarwal, 2021b), the following two ML models were generated to perform health and fault classification:

- *Binary classifier*: This module is a XGBoost[3] binary classifier. With CWP data, it predicts whether the CWP is experiencing normal operation or undergoing any degradation at the pump, motor, or system levels. The model is developed by considering time domain features extracted from vibration data, along with the features extracted from monitoring data. Features such as motor current and vibration data are unavailable prior to September 2017 and October 2019, respectively. The missing features are mapped with NaN values. While training and making predictions, the XGBoost model discards all features with NaN values. The left portion of Figure 10 shows an example prediction generated by the binary classifier model.

- *Diagnostic model*: This module is a multiclass classifier. For CWP data, it predicts the type of fault a CWP is currently undergoing. The model is developed by considering frequency domain features extracted from vibration data, along with the features extracted from PI data. Features such as motor current and vibration data are unavailable prior to September 2017 and October 2019, respectively. The missing features are mapped with NaN values. While training and making predictions, the XGBoost model discards all features with NaN values. The right portion of Figure 10 shows an example prediction generated by the diagnostic model.

[3] Official website: https://xgboost.ai/

	Condition	Probability
Time		
2020-04-01	Unhealthy	0.873438

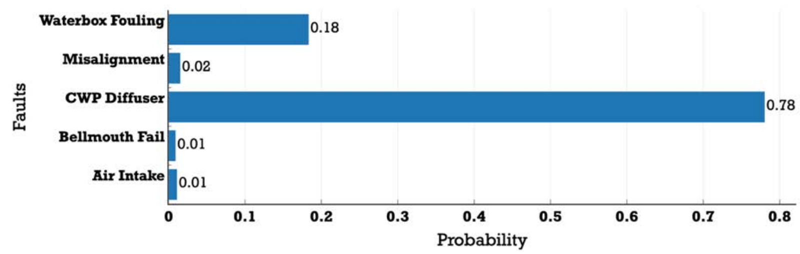

Figure 10. Example prediction by the binary classifier model (left image) by the diagnostic model (right image).

In this setting, a supervised ML model (i.e., a classifier) is trained using both the faulty and healthy datasets ($\Xi^{obs-faulty}$, $\Xi^{obs-healthy}$) and is employed to predict, given ξ^{obs}, the class out (either faulty or healthy) to which ξ^{obs} belongs. Such a prediction can be augmented by also determining the probability estimate $Prob^{detec}$ associated with the prediction out. If the $[0,1]$ margin interval is divided into two equally long segments, we can assign the class "healthy" to the [.5,1] interval and the class "faulty" to the $[0,.5]$ interval. Hence, the predicted class out generated by the ML model determines the margin variability interval (either $[0,.5]$ or [.5,1]). The variable $Prob^{detec}$ is essentially a measure of the prediction accuracy. More precisely, a high value of $Prob^{detec}$ implies a high degree of accuracy in the prediction; conversely, a very low value implies low accuracy. In this context, $Prob^{detec}$ is used to determine the precise margin location in the $[0,.5]$ or [.5,1] intervals. A high value of $Prob^{detec}$ would drive the margin toward the extremes of the intervals (either 0 or 1), whereas a low value of $Prob^{detec}$ would drive the margin toward the common point of the intervals (i.e., 0.5).

Consequently, provided ξ^{obs} and a ML model that can generate both out and $Prob^{detec}$, a margin value can be defined as follows:

$$M(\xi^{obs}) = \begin{cases} 0.5 - \frac{Prob^{detec}}{2} & if\ out = faulty \\ 0.5 + \frac{Prob^{detec}}{2} & if\ out = healthy \end{cases} \quad (1)$$

In this respect, Figure 11 provide a snapshot of the temporal profile of CWS margin associated with the waterbox.

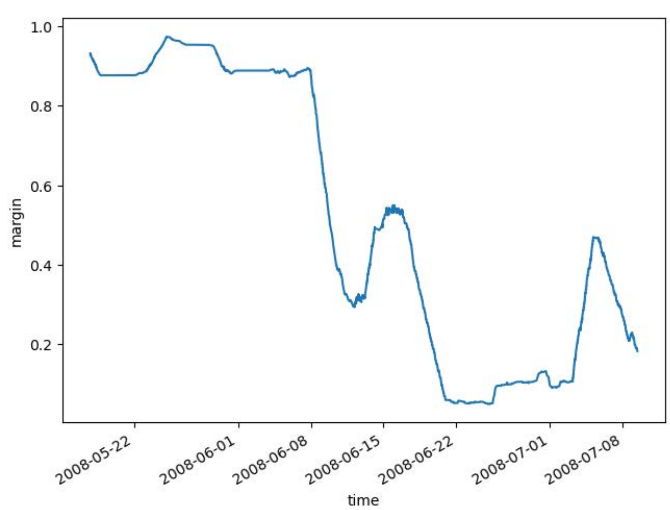

Figure 11. Waterbox margin temporal profile, as determined using a trained classifier (Mandelli, 2023b).

7 Causal Analysis Between ER Data Elements

Once both ER textual and numeric data elements have been processed through the methods described in Sections 5 and 6, the next step is to determine the associations between ER data elements. In the context of this work, this association has a to be interpreted in a causal sense, i.e., the event reported in one data element (the *cause*) has triggered the event reported in the second data element (the *effect*). The determination of the causal association between two data elements is here performed if both of the following conditions hold true:

- A logical association dictated by either inductive reasoning or physics-based considerations between the two events exists (see Section 7.1)
- The temporal correlation between the two events exists (see Section 7.2)

7.1 Logical Analysis

Typically, in order to test if the events reported in two ER data elements have a logical association, system engineers rely on their knowledge about system functional architecture. As an example, if these two events are considered:

"*centrifugal pump is not operating*" , "*4160V AC power system has failed*"

a system engineer might see a logical association between the two events since the considered centrifugal pump requires the 4160V system to operate.

For the scope of this work, the knowledge about system functional architecture is contained in available OPM models described in Section 4 which are translated into graph structures. The logical analysis between two events can then be performed by determining if a logical path (inductive reasoning) in the OPM-induced graph structures between the nodes "*centrifugal pump*" and "*4160V AC*" exists. This path can be visualized by looking at the right portion of Figure 7 where the node "*4160V AC*" directly supports the rotation of pump motor and provide rotation to the impeller responsible to provide water flowrate in the discharge pipe. The existence of this path confirms that the two events have a logical association.

7.2 Temporal Correlation Analysis

Sections 5 and 6 presents methods of analyzing numeric and textual ER data elements, and we explained how OPM diagrams can be employed to identify possible causal relationships between ER data elements (see Section 7.1). The word "possible" is intended to indicate that two events sharing an OPM-based direct relation may in fact exist independently from each other. The first step in testing for such dependence is to observe their temporal correlation. For the scope of this article, we are interested into identifying the existence of a temporal correlation between two events and between an event and a time series.

Regarding the temporal correlation analysis between two events, we are here considering the generic situation where two events (E_1 and E_2) are defined over specific time instances: (E_1, t_1) and (E_2, t_2). Without loss of generality, we assume that $t_2 > t_1$. The assessment of temporal correlation between the events E_1 and E_2 is performed by looking how far, temporally speaking, the two events are. In more detail, we define here a temporal correlation index $I_t(E_1, E_2)$ between the events E_1 and E_2 as follows:

$$I_t(E_1, E_2) = 1 - e^{\frac{-(t_2 - t_1)}{\tau}} \quad (2)$$

where τ represents a decay term that filters out events that are far from each other. The temporal correlation index $I_t(E_1, E_2)$ provides a quantitative measure of the temporal distance among them: if the events E_1 and E_2 are close to each other, then $I_t(E_1, E_2)$ approaches the value of 1. If the events E_1 and E_2 are far from each other, then $I_t(E_1, E_2)$ approaches the value of 0. The parameter τ specifies the scale of the temporal closeness of the two events.

Regarding the temporal correlation analysis between events and time series, our work extends that presented by (Luo, 2014), in which the temporal correlation between time series and events is formulated in terms of a two-sample problem (Gretton, 2006). Our extension includes three relevant items: a modification to the testing process structure, a different two-sample testing algorithm, and the handling of events defined over an interval (as opposed to a time instant).

In its original formulation by (Luo, 2014), the temporal correlation was measured between a set of identical events and the time series. In the scope of the present work, we often deal with single events (e.g., abnormal behavior of an asset) rather than sets of events. The algorithm presented by (Luo, 2014) was based on measuring the statistical difference between the portions of the time series pertaining to both before and after (indicated as l_E^{front} and l_E^{rear}, respectively; see the left-hand plot in Figure 12) the occurrence of an event (as defined over a temporal instant). Our extension, which enables dealing with events defined over a temporal interval (see the right-hand plot in Figure 12) requires the additional time series portion that corresponds to the duration of the event itself: l_E^{dur}.

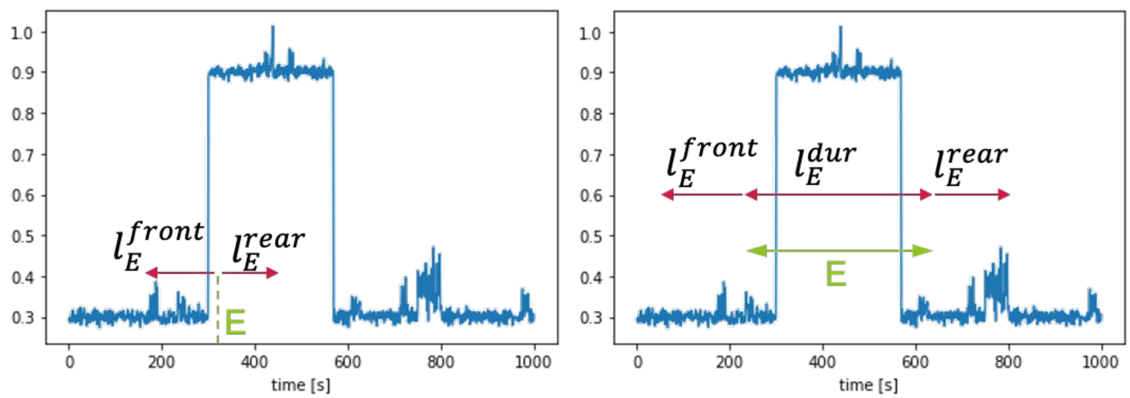

Figure 12. Temporal correlation of a time series with an instantenous (left plot) and interval (right plot) event E.

The plots in Figure 13 were adapted and modified from (Luo, 2014) and provide an overview of the set of cases observable when testing the temporal correlation between time series and events. When indicating the time series with S, we can look at the left-hand plot in Figure 13 and intuitively infer that $E_1 \rightarrow S$, $S \rightarrow E_2$, $E_3 \rightarrow S$, and $S \rightarrow E_4$. Note that the symbol $\rightarrow$ here indicates a temporal relationship between an event E and S but does not necessarily imply a causal relationship between the two. Similarly, by looking at the right-hand plot in Figure 13, we can infer that $E_1 \rightarrow S$, $S \rightarrow E_2$, $S \rightarrow E_3$, and $E_4 \rightarrow S$.

We employ the MMD algorithm (Gretton, 2006) to perform such statistical testing. In its original definition, let S_1 and S_2 be independent random (univariate or multivariate) samples generated from unknown distribution F and G, respectively. The hypotheses of the two-sample test can be stated as follows (i.e., the null hypothesis H_0 and the alternative hypothesis H_1):

$$H_0: F = G$$

$$H_1: F \neq G$$

This is achieved via the following MMD testing with a particular threshold α: if the threshold is exceeded, then the test rejects the null hypothesis (Gretton, 2006). A Type I error (true negative) is made when $F = G$ is rejected based on the observed samples, despite the null hypothesis having generated the data. Conversely, a Type II error (false negative) occurs when $F = G$ is accepted despite the underlying distributions being different. The level α of a test is an upper bound on the probability of a Type I error: this is a design parameter of the test which must set in advance, and it is used to determine the threshold to which we compare the test statistic.

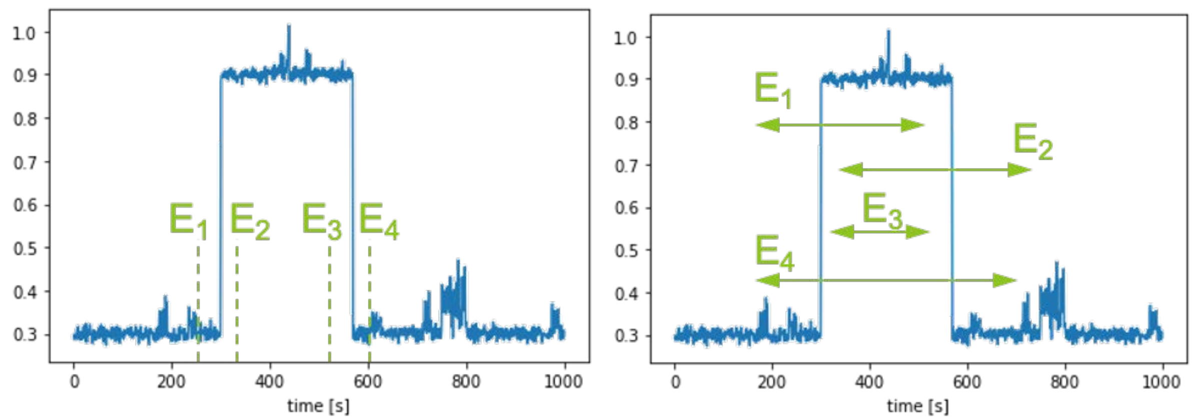

Figure 13. Use cases considered for evaluating the temporal correlation of a time series with a set of instantenous (left plot) and interval (right plot) events.

When dealing with time series, let's consider two time series S_1 and S_2, MMD testing is performed via comparing $MMD_u^2(S_1, S_2)$ to bootstrap random permutations of MMD kernels of S_1 and S_2. If S_1 and S_2 are generated from different distributions (the null hypothesis got rejected), then the probability of $MMD_u^2(S_1, S_2)$ value in the MMD_u^2 distribution (see Figure 14) generated by bootstrap random permutations is lower than a significant level α, i.e.,

$$P(MMD_u^2 > MMD_u^2(S_1, S_2)) < \alpha$$

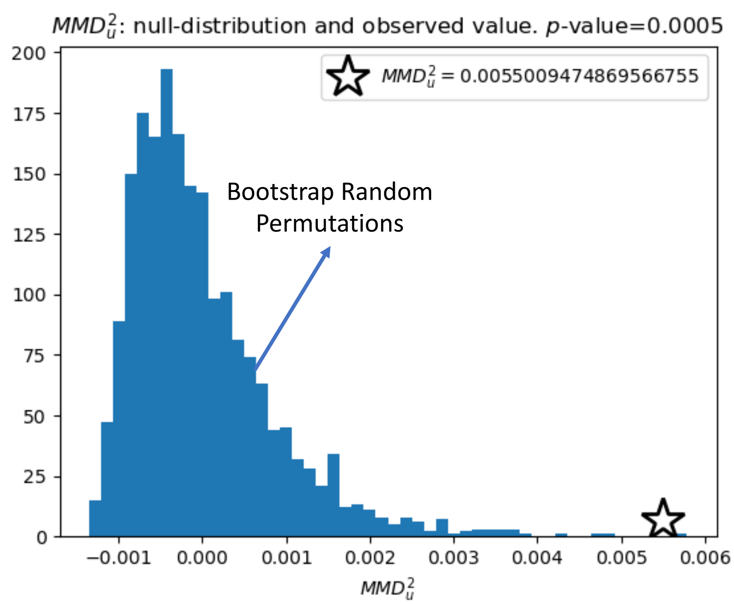

Figure 14. Histogram of MMD_u^2 generated by comparing samples generated by a normal (mean=0, standard deviation=1) and Laplace distribution (location=0, scale=1).

Without a loss of generality, let us consider an event E—defined over either a time instant (E, t_E) or time interval $(E, t_E, \Delta t_E)$—and a time series S (either univariate or multivariate). An identification of the temporal relation between E and S is presented in detail in Algorithms 1 and 2. Both algorithms operate by comparing the statistical distribution l_E^{front} and l_E^{rear} against S. In more detail, this is performed by randomly sampling portions of S that have same duration of l_E^{front} and l_E^{rear}. Such set of portions of S is indicated as Θ.

Given a time series S, and a randomly sampled sub-series Θ from S denoted as $\Theta = (S_1, \dots, S_n, \dots, S_N)$, where each sub-series S_n has the same length of k. Assume S only contains two states, i.e., normal with value of 0 and anomaly with value of 1 with probabilities p_0 and p_1, respectively. We assume $p_1 \ll p_0$. For any sub-series S_E with the length of k, S_E belongs to the anomaly state, if and only if the accept ratio (i.e., the ration between the number of sub-series S_n are similar to S_E based on previous MMD testing over the total number of sub-series of Θ) is below p_1. The null hypothesis, i.e., S_E are generated from the normal states of S, is rejected in this case. In other words, there is a temporal correlation between S_E and the anomaly state of S. As an example, the MMD testing of l_E^{front} vs. Θ and l_E^{rear} vs. Θ is shown in Figure 15.

Algorithm 1: Identification of the temporal relation between event and time series—time instant case

Input: Event (E, t_E), time series $S = (s_1, s_2, \dots, s_m)$
Output: Temporal correlation flag R, direction D

1. Initialize Θ
2. Determine l_E^{front}, l_E^{rear} from t_E
3. Test $T\left(\Theta, l_E^{front}\right)$, and $T(\Theta, l_E^{rear})$
   - Results are denoted as: D_f, D_r
4. Test $T\left(\Theta, l_E^{front} \cup l_E^{rear}\right)$
   - Result is denoted as D_{fr}
5. Test $T\left(l_E^{rear}, l_E^{front}\right)$
   - Result is denoted as d_{fr}
6. If $D_{fr} = False$:
   - Return $R = False$
7. If $D_r = True$ & $D_f = False$: #E1
   - Return $R = True$ and $D = E \to S$
8. If $D_r = False$ & $D_f = True$: #E4
   - Return $R = True$ and $D = S \to E$
9. If $D_r = True$ & $D_f = True$: #E2 and E3
   - Return $R = True$

Algorithm 2: Identification of the temporal relation between event and time series—time interval case

Input: Event $(E, t_E, \Delta t_E)$, time series $S = (s_1, s_2, \dots, s_m)$

Output: Temporal correlation flag R, direction D

1. Initialize Θ
2. Determine l_E^{front}, l_E^{rear}, l_E^{dur} from t_E and, Δt_E
3. Test $T(\Theta, l_E^{front})$, $T(\Theta, l_E^{rear})$, and $T(\Theta, l_E^{dur})$
    - Results are denoted as: D_f, D_r, D_{dur}
4. Test $T(\Theta, l_E^{front} \cup l_E^{dur} \cup l_E^{rear})$
    - Result is denoted as D_{fdr}
5. Test $T(l_E^{dur}, l_E^{front})$
    - Result is denoted as d_{fd}
6. Test $T(l_E^{dur}, l_E^{rear})$
    - Result is denoted as d_{dr}
7. If $D_{fdr} = False$ or $D_{dur} = False$:
    - Return $R = False$
8. If $D_f = False$ & $D_r = True$: #E1
    - Return $R = True$ and $D = E \rightarrow S$
9. elif $D_f = True$ & $D_r = False$: #E2
    - Return $R = True$ and $D = S \rightarrow E$
10. elif $D_f = True$ & $D_r = True$: #E3
    - Return $R = True$ and $D = S \rightarrow E$
11. elif $D_f = False$ & $D_r = False$: #E4
    - Return $R = True$ and $D = E \rightarrow S$

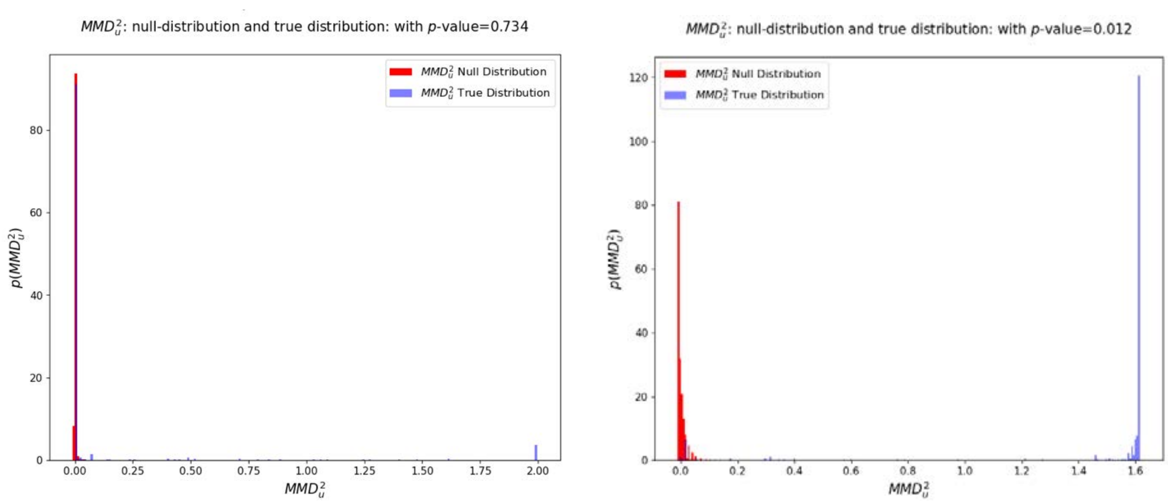

Figure 15. MMD testing of event E_1 shown in the left plot of Figure 13. The left plot shows l_E^{front} vs. Θ, the null hypothesis defined in the previous paragraph is accepted since p value is greater than the significant level value. The right plot shows l_E^{rear} vs. Θ, the null hypothesis previously defined is rejected since p value is smaller than the significant level value.

Table 1. Examples of temporal correlation analysis based on MMD testing for the events E_1, E_2, E_3, and E_4 shown in Figure 13. E_5 is similar to E_4 with 7s temporal shift. E_A, E_B, E_C are specified at location 900s, 500s, 200s respectively.

	l_E^{front} $vs.$ Θ		l_E^{rear} $vs.$ Θ		l_E^{front} $vs.$ l_E^{rear}		$l_E^{front} \cup l_E^{rear}$ $vs.$ Θ		Temporal correlation
	p value	H_0	p value	H_0	p value	H_0	p value	H_0	
E_1 vs. S	0.732	True	0.004	False	0.01	False	0.012	False	E → S
E_2 vs. S	0.012	False	0.052	False	0.01	False	0.02	False	S → E
E_3 vs. S	0.042	False	0.006	False	0.01	False	0.01	False	S → E
E_4 vs. S	0.004	False	0.758	True	0.01	False	0.016	False	S → E
E_5 vs. S	0.008	False	0.048	False	0.01	False	0.004	False	S → E
E_A vs. S	0.726	True	0.728	True	0.63	True	0.72	True	False
E_B vs. S	0.734	True	0.722	True	0.59	True	0.71	True	False
E_C vs. S	0.706	True	0.712	True	0.75	True	0.678	True	False

Lastly, note that the reported time of occurrence of an event is assumed to reflect the actual temporal occurrence of that event. More specifically, the reported occurrence of an event (e.g., sudden bearing failure of a pump) is logged when the event is first observed; however, the actual event may have occurred prior to the logged date (i.e., a temporal delay may exist between the actual and observed occurrence of an event). In such situations, the analysis of the temporal correlation between events and time series may be biased by such delays.

7.3 CWS Testing

An initial testing of Algorithm 1 and Algorithm 2 has been performed on the textual (see Section 4) and numeric (see Section 5) datasets available for the CWS. This testing was set up by initially considering true positive (i.e., a temporal correlation exists) and true negative (i.e., a temporal correlation does not exist) cases. As a reminder, the numeric data described in Section 5 contains not only the temporal profile of several monitor variables, but it also provides information on how those variables are distributed under both healthy (i.e., $\Xi^{obs-healthy}$) and failed conditions (i.e., $\Xi^{obs-faulty}$) for each of the considered failure modes. In this respect, Figure 16 shows the box plot of four considered features for the considered healthy and faulty states. These variables were chosen based on their coverage of all healthy and faulty states. Note that the structure of box plots shifts between healthy and faulty states; in fact, Figure 16 provides a visual comparison between healthy and faulty states by looking at the distribution of a single feature at a time. Such information is precious since the training set Θ required by both Algorithm 1 and Algorithm 2 can be generated by considering only $\Xi^{obs-healthy}$, and as a consequence, the detection performance improves.

The training set Θ was constructed by considering $\Xi^{obs-healthy}$ and it consisted of 2,000 time series characterized by a common length of 24 hours. Then, we applied Algorithm 1 to test the temporal correlation between a known bellmouth failure event and the provided monitored data.

8 Knowledge Graph Construction

Provided the set of processed ER data elements—either textual (see Section 5) or numeric (see Section 6)—we aim to organize each element into a graph structure that captures the health indications of the CWS, and the possible cause-effect relations (logical and temporal) identified in Section 7. The proposed data structure employs the architecture of the system described by its corresponding set of OPM diagram(s).

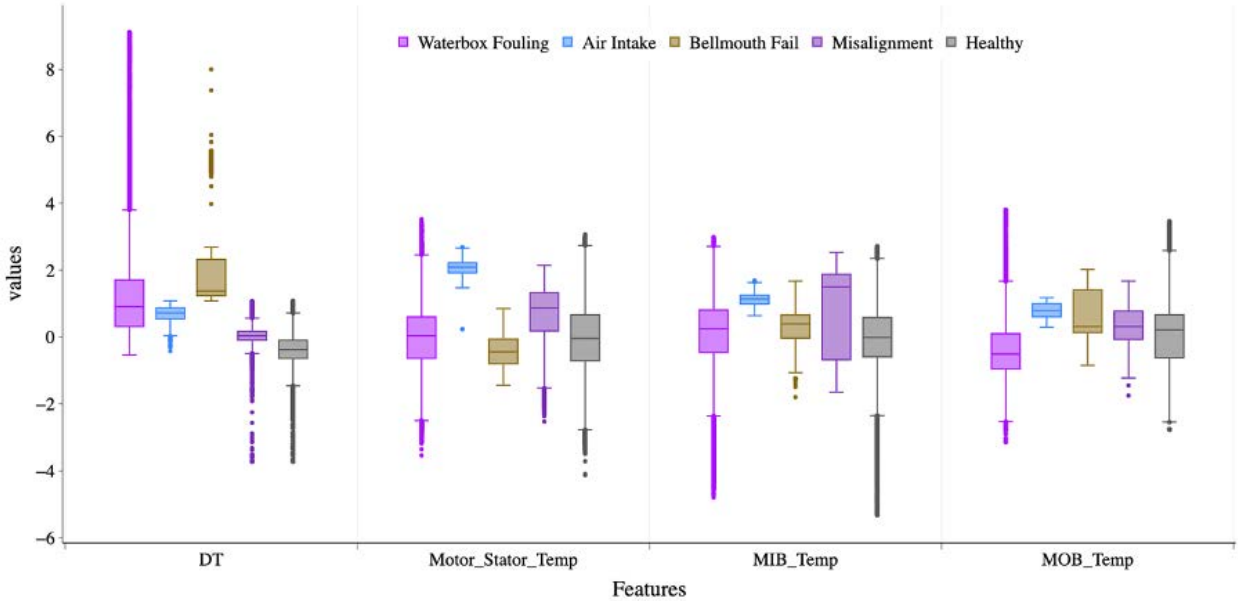

Figure 16. Box plots of four of the considered features (DT, motor stator temperature, motor inboard-bearing, and motor outboard-bearing temperatures) for healthy and failure states.

As indicated in Section 4 the provided OPM diagrams for the considered CWS system and assets are first translated into a digital graph structure and then linked together. OPM entities are represented as nodes in the graph while the directed links intrinsically represent a causal/logical relation between two nodes. Our approach begins with the OPM graph structure of the system and assets under consideration (see Section 4) then it progresses according to the following steps:

1. Associate an ER textual data element with an OPM entity. This step heavily relies on the NLP knowledge extraction method presented in Section 5. In particular, the methods designed to identify the OPM entities from the ER textual data element generate a list of identified entities[4]. This allows us to associate each textual element to a specific OPM entity. If multiple OPM entities have been identified, then the same textual element is associated to the identified entities.
2. Associate each numeric time series to a specific OPM entity. Provided the margin analysis described in Section 6, identify portions of the time series that describe a degradation or abnormal behavior.
3. Identify the causal relationships between data elements; if both temporal (i.e., the coexist in the same temporal window) and logical (i.e., there is a logical path in the OPM graph structure) relations are verified, then create a causal link between the two data elements (see Section 7):
    i. Identify the causal relationships between textual data elements processed in Step 1
    ii. Identify the causal relationships between numeric data elements processed in Step 2
    iii. Identify the causal relationships between textual data elements processed in Step 1 and the numeric data elements processed in Step 2

[4] Given the short length of the available IRs or WOs, normally one single OPM entity is mentioned.

The resulting relational analysis takes the form of a graph structure reflecting the links between the data elements associated with a particular OPM entity. Again, the actual skeleton of the graph structure is directly derived from the OPM diagram of the system and assets under consideration. Note that the obtained graph structure is composed by:

- Nodes that represent either OPM entities or ER data elements (numeric or textual)
- Edges between nodes that can be of three types:
  - Logical/causal relations between OPM entities
  - Associations between ER data elements and OPM entities
  - Causal relations between ER data elements

8.1 CWS Knowledge Graph

As indicated in Section 8, the starting point is the construction of the graphs from the OPM diagrams. In this respect, Figure 17 and Figure 18 show the graph structures of the CWS system and the centrifugal pump that has been directly generated from the provided OPM diagrams. Note that the graph nodes can reflect different data types (form or function), and the same applies to edges.

The available ER dataset for the considered CWS was collected over the past 14 years. Regarding Step 1 indicated in Section 8, knowledge extraction methods presented in Section 5 were employed to analyze all WOs and IRs, enabling us to identify the nature of textual and OPM entities associated with them. In this respect, Figure 19 provide a histogram of the OPM entities that are mentioned in the provided dataset. As an example, Figure 20 shows some elements that are associated with the “waterbox” node indicated in the left portion of Figure 17.

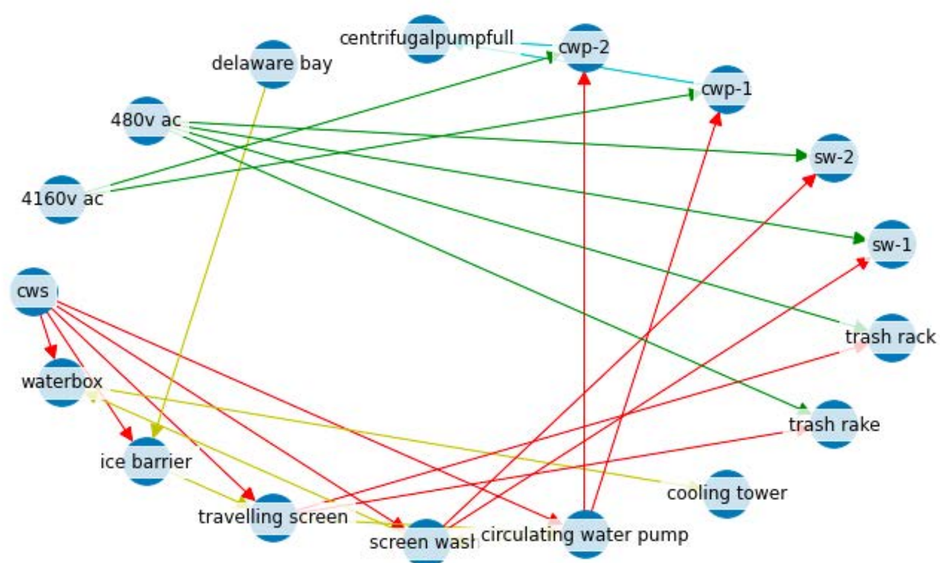

Figure 17. Obtained CWS data graph structure.

Regarding Step 2 indicated in Section 8, we applied the margin analysis originally performed in (Mandelli, 2023b) given the available ER numeric time series. From such margin analysis, we were able to retrieve failure events; as an example, Figure 21 shows the temporal profile of the margin associated with the CWP motor air intake failure where an actual event highlighted in green has been identified.

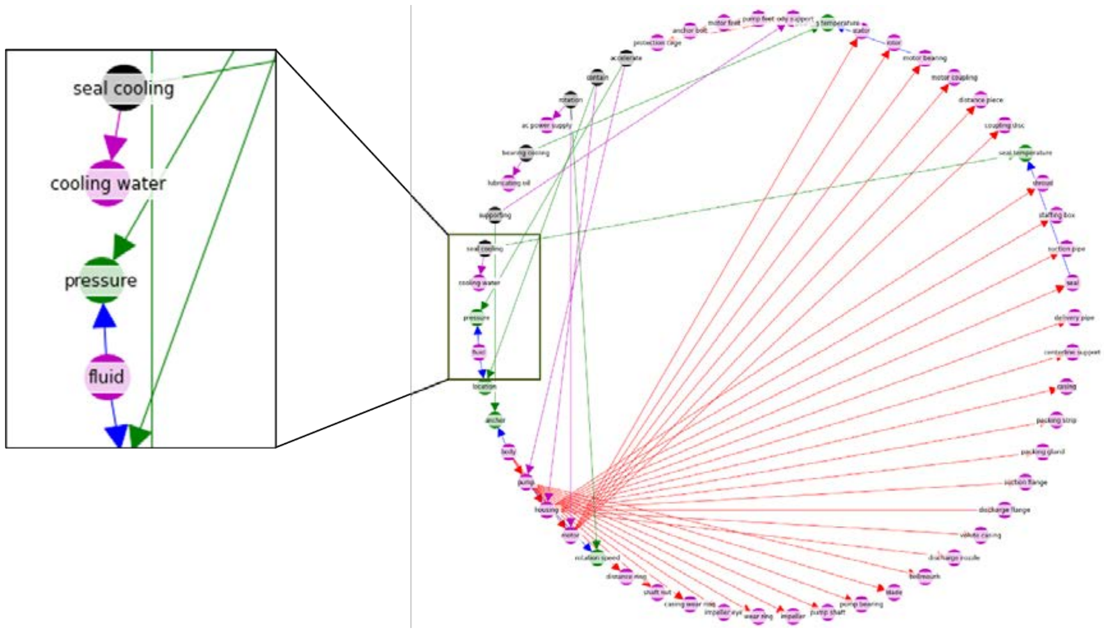

Figure 18. Obtained centrifugal pump data graph structure.

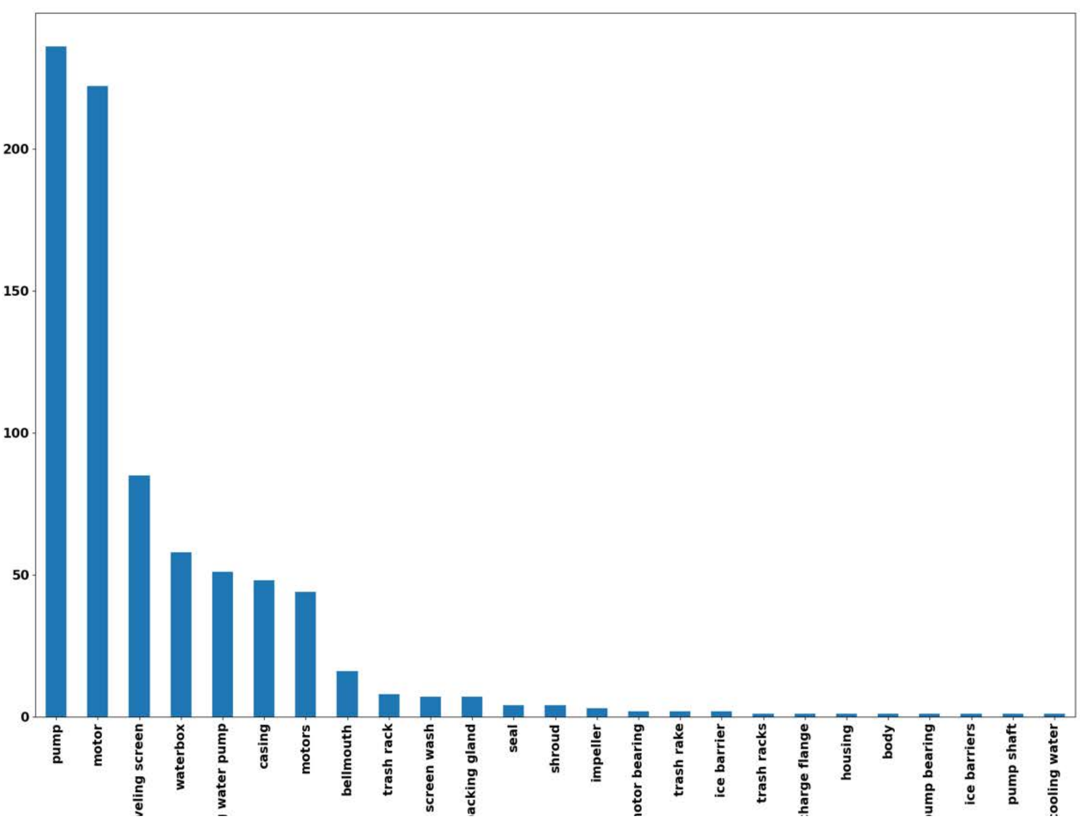

Figure 19. Distribution of OPM entities mentioned in the CWS textual dataset.

	entities	root status	keywords	health statuses	conjecture
1	waterbox	None	None	oc tube leak	False
2	waterbox	None	None	oc tube leak	False
3	waterbox	None	None	tube leak	False
4	waterbox	None	None	tube leak	False
..	...	...	...	...	...
153	waterbox	None	None	oc tube leak	False
154	waterbox	None	None	oc tube leak	False
155	waterbox	None	None	oc tube leak	False
156	waterbox	None	None	outlet leak weld repair	False
157	waterbox	None	None	section	False

Figure 20. Processed IRs and WOs associated with the OPM entity "waterbox."

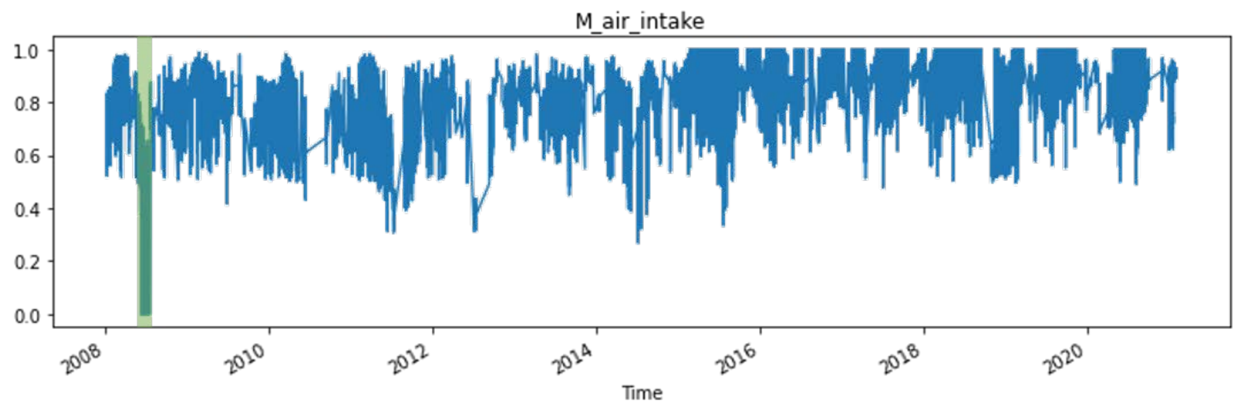

Figure 21. Temporal profile of the margin associated with the CWP motor air intake failure mode.

9 Conclusion

This paper presented a series of methods and algorithms for extracting knowledge from ER data for diagnostic and data management purposes. One main feature of these methods is that they are not based solely on data but are in fact model based. In other words, they also rely on MBSE (OPM) models designed to capture—from a functional point of view—the architecture of the systems and assets under consideration. The main purpose of such models is to digitally emulate system engineers' knowledge of system and asset architecture and to identify dependencies among systems, assets, and components. Provided these models, analyses of textual and numeric ER data can be performed by first identifying the OPM model elements to which the ER data elements are referring. The causal relationships between ER data elements are then identified by checking for any temporal or logical dependencies. The last step is to organize the available ER data elements into a graph structure (i.e., a knowledge graph). In the present work, the OPM model structure served as the skeleton of the graph, with the ER data elements being matched to specific OPM entities. In addition, if both a logical and temporal relationship are identified between two ER data elements, a direct link between them is created. The resulting graph provides a structured, complete, and organized architecture for analyzing and visualizing complex ER datasets to provide an overview of the ER performance of systems and assets.

References

Agarwal, V., K. A. Manjunatha, A. V. Gribok, J. A. Smith, V. Yadav, N. J. Lybeck, M. Yarlett, et al. 2021a. "Machine Learning and Economic Models to Enable Risk-Informed Condition Based Maintenance of a Nuclear Plant Asset," Idaho National Laboratory Technical Report, INL/EXT-21-61984. https://www.osti.gov/servlets/purl/1770866.

Agarwal, V., K. A. Manjunatha, A. V. Gribok, T. J. Mortenson, H. Bao, R. D. Reese, T. A. Ulrich, et al. 2021b. "Scalable Technologies Achieving Risk-Informed Condition-Based Predictive Maintenance Enhancing the Economic Performance of Operating Nuclear Power Plants," Idaho National Laboratory Technical Report, INL/EXT-21-64168. https://doi.org/10.2172/1894498.

Borky, J. M., and T. H. Bradley. 2018. *Effective Model-Based Systems Engineering*. Springer. https://doi.org/10.1007/978-3-319-95669-5.

Dori, D., and E. Crawley. 2002. *Object-Process Methodology: A Holistic Systems Paradigm*. Heidelberg: Springer. https://doi.org/10.1007/978-3-642-56209-9.

Friedenthal, S., A. Moore, and R. Steiner. 2008. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann. https://doi.org/10.1016/C2013-0-14457-1.

Gretton, A., K. Borgwardt, M. Rasch, B. Scholkopf, and A. Smola. 2006. *A Kernel Method for the Two-Sample-Problem*. Advances in Neural Information Processing Systems, MIT Press, vol. 19. https://doi.org/10.48550/arXiv.0805.2368.

Lane, H., H. Hapke, and C. Howard. 2019. *Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python*. Shelter Island, New York: Manning Publications.

Luo, C., J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang. 2014. *Correlating Events with Time Series for Incident Diagnosis*. KDD'14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1583–1592. https://doi.org/10.1145/2623330.2623374.

Mandelli, D., C. Wang, and S. Hess. 2023a. *On The Language of Reliability: A System Engineer Perspective*. Nuclear technology. https://doi.org/10.1080/00295450.2022.2143210.

Mandelli, D., C. Wang, V. Agarwal, L. Lin, K. A. Manjunatha. 2023b. "Reliability Modeling in a Predictive Maintenance Context: A Margin-Based Approach." Under review in Reliability Engineering and System Safety.

William, S. 2004. *The Object Primer: Agile Model Driven Development with UML 2.0*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511584077.