INL/RPT-24-78817

Light Water Reactor Sustainability Program

Automated Knowledge Extraction from Plant Records to Support Predictive Maintenance

June 2024

U.S. Department of Energy

Office of Nuclear Energy

INL/RPT-24-78817

Automated Knowledge Extraction from Plant Records to Support Predictive Maintenance

D. Mandelli, C. Wang, C. M. Godbole, V. Agarwal
Idaho National Laboratory, Idaho Falls, USA

M. Movassat, B. Mori, D. Liang, E. Nur, A. Birjandi, B Lobo, N. Jacome
Ontario Power Generation, Toronto, Canada

June 2024

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

Page intentionally left blank

ABSTRACT

For almost a decade, the U.S. Department of Energy has been sponsoring the Light Water Reactor Sustainability Program with the goal of developing solutions and technologies to improve economics and reliability, sustain safety, and extend the operation of nation's fleet of nuclear power plants. In this respect, the Risk-Informed Systems Analysis pathway of the Light Water Reactor Sustainability Program is focusing on the research, development, and deployment of solutions designed to assist operating nuclear power plants (NPPs) to reduce operating costs, maintain safety standards, and improve plant reliability and availability.

One of the Risk-Informed Systems Analysis research areas is focusing on developing computational methods and tools to optimize plant operations (e.g., maintenance operations and aging management of plant structures, systems, and components) such that plant operational cost can be reduced while system reliability and availability are maximized. Such optimization can be realized when adequate monitoring data is available where such data can be employed to adequately assess the health status of assets and components. Such data can provide system engineers with valuable insights and information regarding the presence of anomalous behaviors or unexpected degradation phenomena, and guide engineers into identifying the possible causes behind such behaviors and phenomena.

However, a current challenge that NPP system engineers are experiencing is that the amount of equipment reliability (ER) data being continuously generated is not only extremely large, but it also comes in different forms: textual (e.g., condition or maintenance reports) and numeric (e.g., generated by monitoring systems). On top of it, such data is often kept in physically different databases (e.g., plant operation servers and plant monitoring and diagnosis servers) with little or no possibility to cross-reference the information contained across different databases to assess the reliability history of plant assets and components.

The advanced modeling and data analytics project tackles this plant operational challenges with the goal of building a robust bridge between plant ER data and system engineer decision-making regarding maintenance activity scheduling and aging management. Such a bridge is built on top of a set of computational tools designed to analyze ER data, perform system modeling, and optimize plant resources (personnel, time, and money).

This report summarizes the activities performed within the advanced modeling and data analytics project during Fiscal Year 2024 in collaboration with a nuclear

power utility. Our work activities focused on a fairly unique task: the analysis and integration of ER data in all its forms, numeric and textual.

Our approach takes inspiration from the latest research and development in the medical field where integrating several data sources is vital to assist medical practitioners achieve correct diagnoses and indicate optimal treatments. Even though the operational context is different, there are a lot of similarities; the goal is to guarantee system and asset operation based on actual and historic condition-based data.

One of the unique aspects of our approach is that it is strongly based on model-based system engineering (MBSE) models of systems and assets; these models are designed to graphically capture their architecture from a form and functional standpoint. These models basically emulate system engineer knowledge about systems and assets architecture and dependencies between systems and assets. We think that this is the key to "put data into context." Context is here intended as the information required by ER data analysis tools to understand what these data elements are referring to, that is, which kind of knowledge they are generating.

Once this MBSE models have been developed, ER data elements are processed by identifying first which elements of the developed MBSE elements they are referring to. When dealing with numeric ER data, each condition-based monitoring data time series is associated with a unique MBSE entity. As an example, the time series generated by the sensors designed to monitor the winding temperature of a centrifugal pump is associated with a specific pump (or the actual component of the considered pump, i.e., the pump stator). In this work, we focused our work on the preprocessing of the time series obtained by a nuclear power utility and on identifying anomalous behavior from the processed time series.

When dealing with textual data we aim as well to associate a textual element to one (or more) MBSE entity. This association process requires the text to "be understood" by a computational tool: we refer this process as "knowledge extraction." In the medical field, several approaches designed to extract knowledge from textual data have been developed in the past decade. By following the paths developed in the medical field, we have adapted such approaches to system reliability operations applied to an NPP context. From the nuclear power utility, several databases of issue reports, maintenance reports, and shift logs were provided, and they were processed by the developed knowledge extraction methods.

The process of associating MBSE entities to the set of anomalies obtained from numeric ER data and the set of anomalous behaviors inferred from textual ER data constitutes the first-of-its-kind knowledge graph. A knowledge graph is a digital structure designed to capture system architecture (derived from the system MBSE model) and historic performance of its constituent assets and components (obtained from the processing of numeric and textual ER data elements).

The design of the knowledge graph allows our data analytics methods to identify the possible correlations and cause-effect relations among anomalous behaviors. This is performed by observing if a logical connection through the MBSE models exists and if there is a temporal correlation among them. The logic and temporal are the two main ingredients to perform the first-of-its-kind "machine reasoning" from ER data.

This report presents a summary of the research and development activities performed during Fiscal Year 2024 in tight collaboration with an NPP for a specific system. First, we show how the digital modeling of NPP systems and assets is performed using state-of-the-art MBSE models. Then, we provide details about developing computational methods designed to process and analyze numeric and textual ER data elements and show how our innovative data integration is performed in a MBSE context. We then provide details on how the proposed development can support actual system engineer decisions in terms of maintenance operation optimization and asset aging management.

Page intentionally left blank

CONTENTS

FIGURES

TABLES

Page intentionally left blank

ACRONYMS

API	application program interface
CWS	circulating water system
DACKAR	digital analytics, causal knowledge acquisition and reasoning
ER	equipment reliability
FY	fiscal year
INL	Idaho National Laboratory
MBSE	model-based system engineering
ML	machine learning
M&D	monitoring and diagnostic
NPP	nuclear power plant
NLP	natural language processing
OPG	Ontario Power Generation
PHM	prognostic and health management
S-ML-AI	statistical, machine learning, and artificial intelligence
TLP	technical language processing

Page intentionally left blank

AUTOMATED KNOWLEDGE EXTRACTION FROM PLANT RECORDS TO SUPPORT PREDICTIVE MAINTENANCE

1. INTRODUCTION

The past two decades have seen the emergence of advanced prognostic and health management (PHM) computational tools for anomaly detection, diagnostic, and prognostic purposes, which are helping system engineers and plant operators monitor the performance of several assets and optimize plant resources (personnel, time, and money). In the same direction, nuclear power plants (NPPs) are now in the process of digitizing operation and maintenance activities to track trends and events at the system or plant level (e.g., plant planned shutdown or system taken out of service) and, more importantly, observed abnormal conditions. In this respect, NPPs generate large amounts of equipment reliability (ER) data, which record the historic performance of a large number of components and assets. As a drawback, engineers and operators are now facing the challenge of processing the ER data being continuously generated, which is not only extremely large but also appears in different forms: textual and numeric.

This report directly tackles this challenge by providing computational methods to assist system engineers and operators with the means to extract knowledge from ER data (see Figure 1). The first point we claim here is that all the ER data elements described earlier provide equally-important indications about asset and system performance and, hence, cannot be analyzed separately. The second claim is that generating knowledge from data requires the ability to put data into "context." Here, context is the additional information needed by ER data analysis tools to understand what these data elements are referring to.

To support these two claims, our approach deviates substantially from state-of-practice methods where the main focus is almost exclusively on numeric data analysis. Instead, we employ model-based system engineering (MBSE) approach to represent systems and assets to capture their architectural and functional (i.e., cause-effect) relations. Textual and numeric 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 provided system design documents which indicate a precise association between asset and monitored time series. On the other hand, this task is more challenging when dealing with textual data; we employ technical language processing (TLP) methods to "extract knowledge" from textual elements.

Filtering abnormal behaviors can then be performed from numeric data through anomaly detection methods and textual elements (by understanding their semantic nature). Such abnormal instances that are associated with a specific MBSE element are then stored in a relational database. This database takes the form of a graph where the main skeleton is the actual system MBSE model and abnormal instances are "linked" to such skeleton. At this point, both numeric and textual data elements are integrated and put into context. From here, graph-based analysis methods can be employed to perform "machine reasoning" including identifying abnormal patterns and the root-cause behind them.

This report is structured as follows:

- Sections 2 provides the operational context of our work and Ontario Power Generation (OPG) vision in terms of plant modernization while Section 3 summarizes the collaboration that occurred during Fiscal Year (FY) 2024 between LWRS and OPG to support OPG vision;
- Section 4 provides an overview of the activities performed during FY24 in terms of the analysis of numeric and textual ER data and their integration to assist system engineers identify degraded performance and the correlation between events;

- Section 5 summarizes how the work shown in Section 4 support the utility vision of plant modernization and, in general, can support nuclear industry predictive maintenance decision-making;
- Appendix A provides more technical details of this activity in the form of a journal paper that will be submitted shortly after the release of this report;
- Appendix B provides technical details about the developed TLP methods in the form of a journal paper that has been published during FY2024 for the Energies journal.

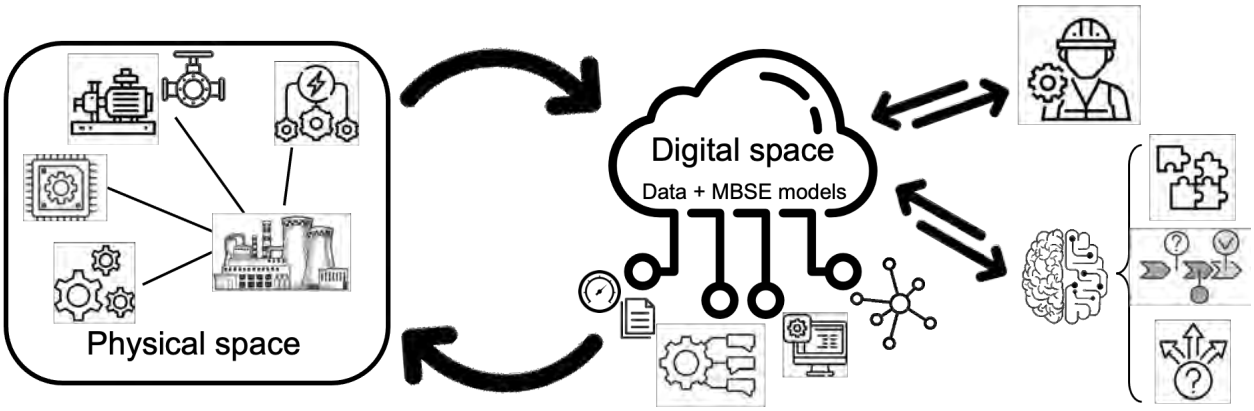

Figure 1. Operational context of this report: developing computational tools (digital space) to support the maintenance decision-making of complex systems (physical space).

2. UTILITY VISION

To achieve operational excellence, OPG Enterprise vision is to be an industry leader in plant reliability driven by proactive and innovative solutions across our fleet. This will ensure that safe and reliable electricity is available for the citizens of Ontario for the coming decades. Knowing the expected growth of demand due to electrification, and our commitment to a net zero business model by 2040, underlines the importance of reliable running of our current fleet, and leveraging our experience for the upcoming new builds, i.e. SMRs.

To support our plant reliability, OPG adopted establishing Monitoring and Diagnostics (M&D) Centre in 2018. Industry benchmarking proved the value of having M&D Centre for two main purposes: early detection of equipment degradation compared to traditional static alarms, and supporting the move from time base maintenance to condition base maintenance (CBM) by leveraging models developed by M&D. Since its establishment, M&D has proved its business value through achieving both above purposes. OPG also added thermal performance (TP) modelling capabilities in M&D Centre which has avoided generation losses due to cycle isolation and secondary side inefficiencies.

The value of online instrumentation data goes beyond early degradation detection, CBM, and TP. The data can be used to gain efficiencies through automating repetitive tasks, to perform customized calculations to assist with decision making and to provide dashboards for reporting purposes. To achieve these added values, OPG M&D Centre added an enhanced data analytics platform to its software fleet. The platform, having a module for open-source programming, has opened substantial opportunities for the data to be used to assist with decision making and gaining efficiency.

OPG M&D Centre has gone through various benchmarking either through direct working with peer utilities to exchange best practices or by participating in EPRI studies. Through these benchmarking, OPG has learned from operational experience and has enhanced its monitoring practices to improve plant

reliability. M&D has been also recognized by various organizations for its innovative solutions to support OPG operations.

The next step for OPG M&D Centre is to provide explainability to the instrumentation trend changes that are observed. The current M&D reliability models are all mathematics-based models. The models identify a change in the trend by comparing the historical pattern of the specific instrumentation with other correlated points. If deviation passes a threshold, the model alarms and M&D analysts will investigate the case. The models do not provide any diagnostics or explainability to why the trend has changed.

To provide explainability, physics-based modelling needs to be added to the current monitoring strategy. By this addition, the changes in the trends will be attributed to the drivers behind the change. M&D would be able to answer the question of "why there is a change" compared to current situation of informing "there is a change". This will increase the actionability of alarms M&D is communicating with Operations. In the past, there were cases where due to lack of explainability, the alarm about a change in trend had gone under radar until equipment failures happened.

The complexity of physics-based models depends on what is the expected outcome form the model. The models can range from zero-dimensional models, to reduced order models, to full three-dimensional models. With currently available computational resources, zero-dimensional and reduced order models are achievable to run on live data. Currently, there is no standardized solution for the utility industry to provide these types of models. OPG in collaboration with INL has started developing its first hybrid model, a combination of mathematics and physics-based models. The collaboration would expand as OPG starts deploying and using these hybrid models.

In addition to developing these hybrid models, OPG and INL are collaborating to leverage the available text data for further Explainability. Data sources such as operators shift logs convey important information about the operation of the plant that is currently underutilized for plant reliability purposes. With the recent development in natural language processing (NLP) and large language models (LLMs), unlocking the value from these text information sources has become practical.

The combination of hybrid models and the information from text data would give a real time picture of the condition of the asset and the associated risks for its operation. This model would be a digital twin for the asset which can be used for calculating remaining useful life of the asset as well. The benefits of developing these digital twins are beyond plant reliability and can be leveraged for long term asset planning and asset management. OPG M&D Centre vision is to develop digital twin models and leverage them for data-driven decision making across the business.

3. LWRS-UTILITY COLLABORATION

In collaboration with the nuclear utility partner, OPG, we tested our methods on an NPP specific system: the circulating water system (CWS). The CWS system is used in many types of thermal power plants (e.g., coal, gas, oil) and it is designed to remove the residual heat from the turbine-condenser system and release it into the environment. Water is collected from a water body (e.g., lake or river) in service gates. Then, using traveling screens, it is cleaned of debris, water life, and foreign bodies that might damage CWS components. Screen wash pumps provide spray water to remove debris accumulation on the screens. Then, water is pumped through heat exchangers located in the plant secondary loop, which removes the heat from the turbine-condenser system. Lastly, the warm water is then released downstream of the same water body. From an operational standpoint, even though the CWS system does not directly support a plant safety function, any degradation of its performance or abnormal behaviors may directly affect power generation (either in a power derate or power shutdown) and, consequently, plant economic revenues.

In this respect, the industry partner has provided a large amount of proprietary data that has been provided through secure channels. Such data included:

- *Condition-based monitoring data* of several CWS assets, supporting systems, and environmental variables collected during the 2012–2022 time frame;
- *Reactor operator shift logs* of events related to the CWS system;
- *CWS condition reports* of abnormal events in the CWS system;
- *CWS work orders* for maintenance operations performed to CWS system;
- *Plant outage data* of time instances where the plant was shut down for either planned or unplanned outages;
- *Designed documents* gave us precise information about the architecture and functional relations between the CWS system, the rest of the plant, and the assets that are part of the CWS system.

Given the proprietary nature of data used in this project, this report does not provide details about the plant and system. Similarly, the outcome of each analysis step reported here has been digitally edited, obscured, or hidden to secure the provided proprietary information. A large amount of information has also been shared throughout monthly meetings between INL and the industry partner. In such meetings, current INL progress was shown and initial results were checked by plant personnel (managers, data scientists, and system engineers). In addition, INL model and data assumptions were validated by the plant personnel.

4. INL RESEARCH, DEVELOPMENT, AND DEMONSTRATION ACTIVITIES

Based on the interactions that occurred during FY-24 with the industry partner, we have pursed several research and development activities were conducted to build a bridge between the available ER data and knowledge (see Section 3) and the utility digitalization vision (see Section 2). These activities were:

- A digital representation of systems, assets, and components through MBSE diagram-based representation (see Section 4.1)
- An analysis of condition-based (numeric) ER monitoring data (see Section 4.2)
- An analysis of condition-based (textual) ER data (see Section 4.3)
- ER data fusion and integration with a system digital model (see Section 4.4).

4.1 System Modeling

A unique element of our approach that differentiates us from state-of-art methods is that we rely on system and asset models to "put data into context." Simply speaking, we aim to emulate system engineer knowledge about system architecture. While the term system architecture is sometimes not well defined and might differ from context to context, we use the following system engineering definition which includes several aspects:

- The decomposition of systems into its constituent assets and assets into constituent components
- The functional representation of systems, assets, and components
- The operands that the defined function acts upon
- The dynamic behaviors of the interactions between systems, assets, and components.

The past decades have seen the emergence of a model-based approach to system engineering: the MBSE approach. Such an approach allows users to systematically decompose system architecture into form and functions using a precise diagram-based language.

Figure 2 shows how a generic asset commonly found in NPPs, a centrifugal pump, is translated into a (very) basic diagram where its architecture is captured, both functional (increased fluid pressure) and form (its constituent elements, such as shaft and bearings). It is important to note that system engineers possess the same structural and functional information of the same asset. In our approach, such information is digitized.

From the CWS system design documentation, we have developed an MBSE model of the CWS system (see Figure 3) that captures all assets and components (including their corresponding ETAG ID) and the supporting systems (e.g., AC and cooling systems).

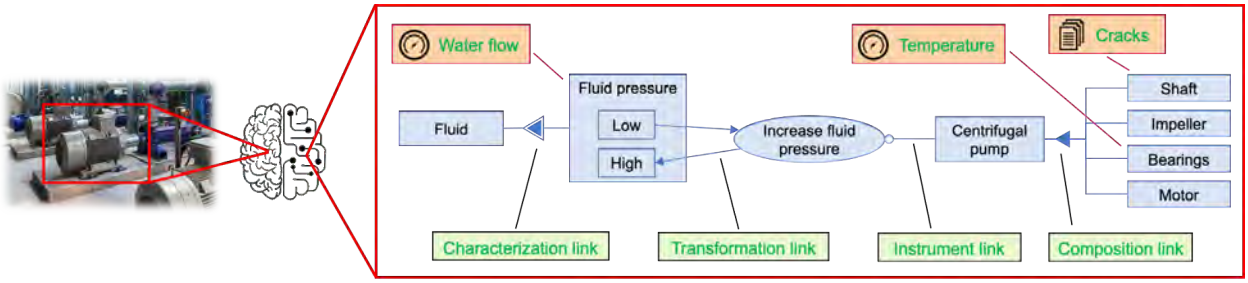

Figure 2. Digital representation of the architecture of a centrifugal pump using a basic MBSE diagram.

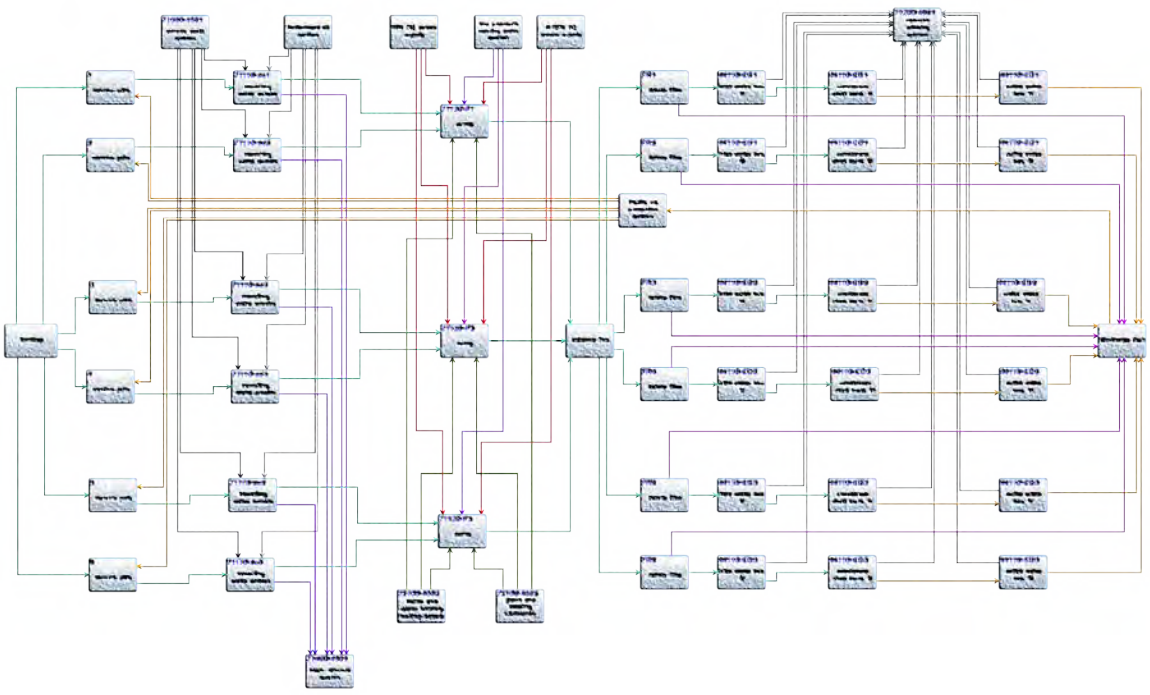

Figure 3. MBSE model of the considered CWS (intentionally edited to obscure proprietary information.

At this point, a question may arise from the reader: what is the role of ER data in an MBSE context? Using the simple example shown in Figure 2, condition-based numeric data, such as the water flow generated by the pump and bearing oil temperatures, can be assigned to specific entities of the MBSE generated diagram: fluid pressure and bearing MBSE entities.

Similarly, textual ER data elements, such as an issue report of the presence of cracks in the pump shaft, can be associated with a specific MBSE, i.e., entity shaft, where the identified degradation element (i.e.,

cracks) is recognized as a possible future cause of pump's failure or degraded performance. Section 4.3 provides details about how ER textual elements are processed to extract quantitative information from them.

4.2 Analysis of Numeric Data

NPPs are continuously monitoring the functional and health performance of several assets that are relevant from a safety and reliability point of view (e.g., vibration data, oil temperature, water pressure). Collected data is continuously processed via advanced PHM systems in the utility monitoring and diagnostic centers. The goal is to (see Figure 4):

- *Detect* data trends and anomalies that may inform system engineers of the degraded performance or failure of the considered asset (i.e., Sensing)
- *Identify* which type of failure mode is being observed such that adequate replacement parts can be procured (i.e., Diagnostics)
- *Predict* the temporal occurrence (i.e., prognostic) of the loss of performance of the considered asset such that maintenance activities can be scheduled to prevent failure (i.e., Prognostics)

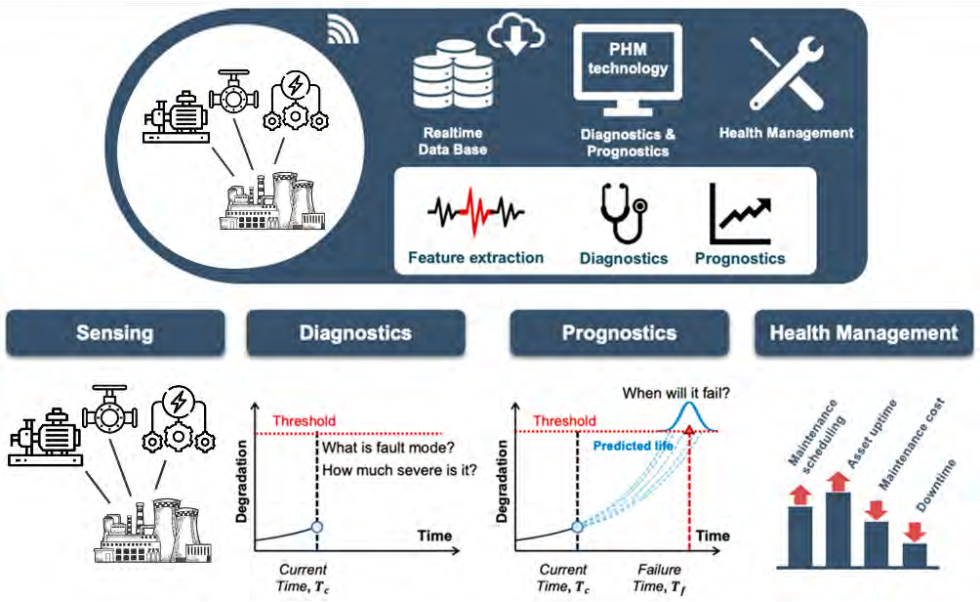

Figure 4. PHM computational technologies to assess asset health status through diagnostic and prognostic methods, adapted from Kim (2021).

During FY24, we have focused on developing methods to identify anomalies in the CWS system that integrate state-of-art machine learning (ML) models and tested them on the CWS condition-based monitoring data described in Section 3. As an example, Figure 5 shows an anomaly detection analysis based on a matrix profile (Yeh, 2016) using training data from the 2012–2017 (blue temporal profile of the upper plot of Figure 5). The anomalies highlighted in red were detected in the 2017–2022 time window (orange temporal profile of the upper plot of Figure 5) by identifying temporal regions characterized by high values of matrix profile (shown at the bottom plot of Figure 5).

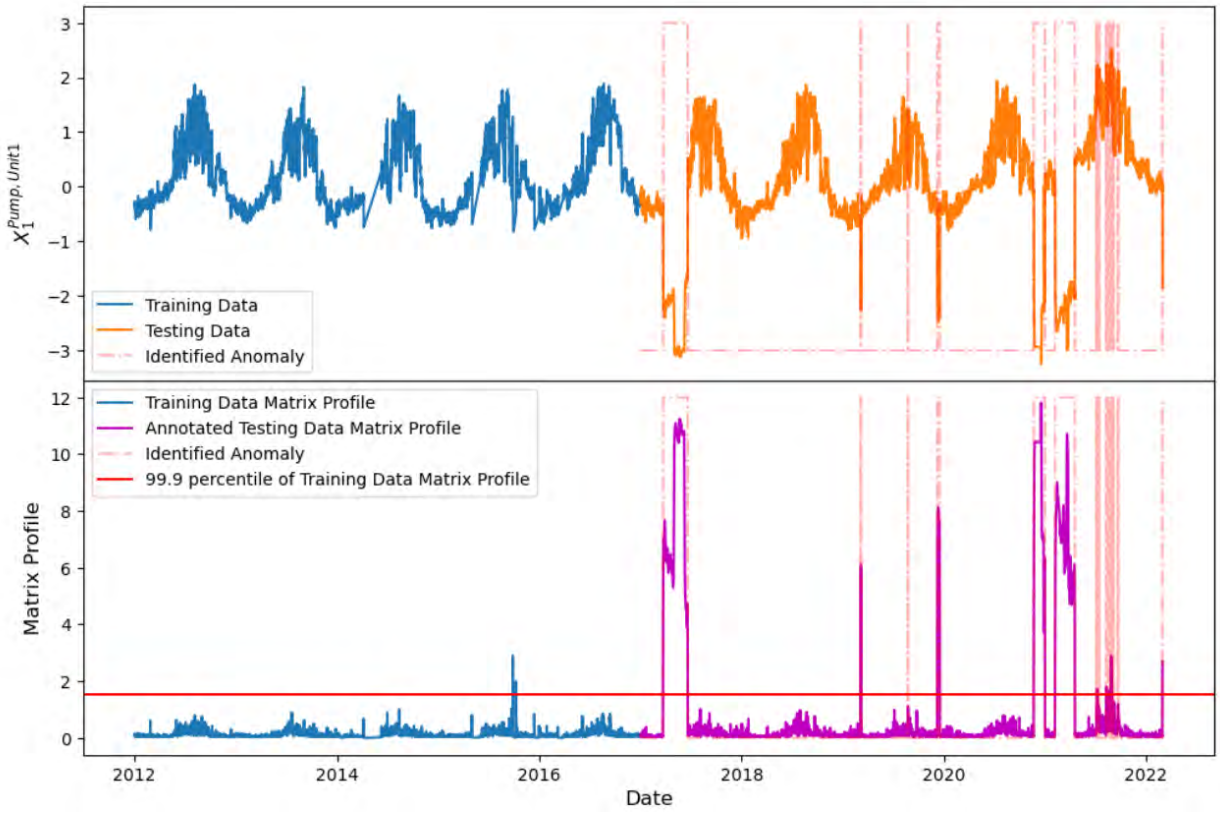

Figure 5. An example of anomaly detection methods applied to CWS data that has been intentionally edited to hide any proprietary information.

4.3 Analysis of Textual Data

ER textual data elements, such as issue reports and work orders, are valuable data sources for tracking asset health histories, identifying health trends, and performing root-cause analyses. These data sources, typically obtained in text form, are usually available in digital repositories. Natural language processing methods (Lane, 2019) have been developed over the past two decades to enable ML models to analyze textual data and classify textual elements based on their nature (e.g., safety related vs. non-safety related).

Here, 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 developing 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 identifying plant-specific entities (such as systems, assets, and components), the temporal attributes that characterize events (e.g., the occurrence, duration, and order of events), specific phenomena (e.g., material degradation or asset functional failure), and causal relations between events.

This knowledge extraction is enabled by a series of data, models, and methods. The developed series of TLP methods was designed to identify all elements listed above, using a mixture of rule-based and ML algorithms. These methods (Wang, 2024) 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).

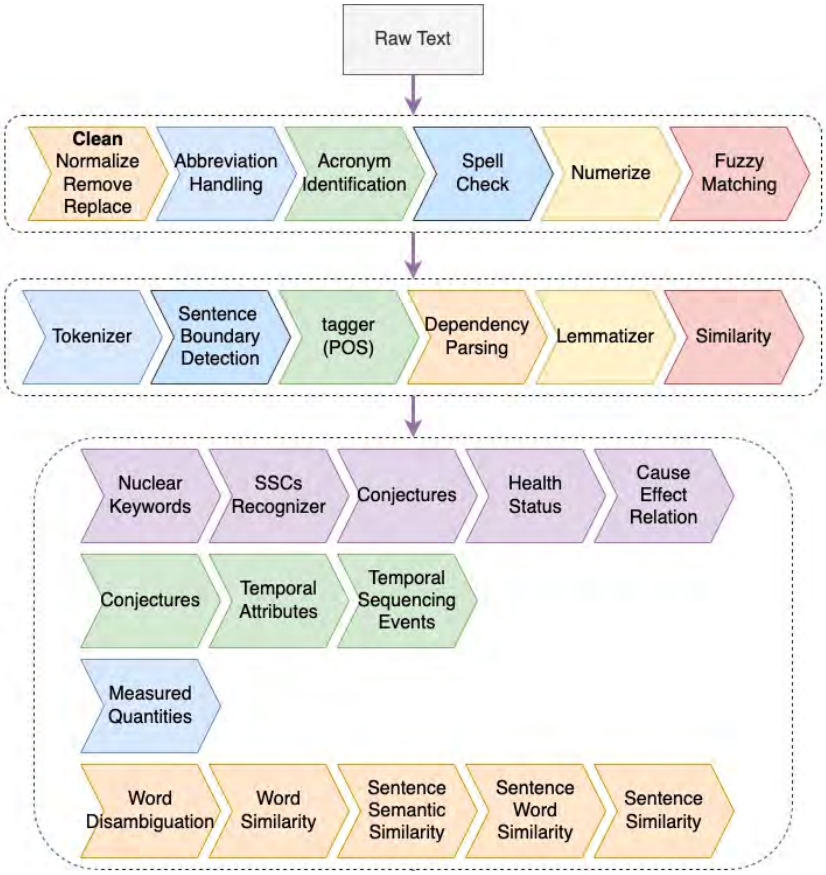

Figure 6. Graphical representation of the NLP elements that comprise the knowledge extraction workflow (Wang, 2024).

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 MBSE entities are mentioned in ER textual data elements.

Figure 6 provides an overview of the NLP methods that together constitute the knowledge extraction workflow. These methods are grouped into three main categories:

- *Text preprocessing*: The raw text is cleaned and processed to identify specific nuclear entities and acronyms (e.g., HPI in reference to a high-pressure injection system), and to identify and correct typos (i.e., through a spell check method) and abbreviations (e.g., “pmp” meaning “pump”).
- *Syntactic analysis*: This analysis identifies the relationship between words in a sentence, focusing on understanding the logical meaning of sentences or parts of sentences (e.g., subjects, predicates, and complements).
- *Semantic analysis*: This analysis identifies the nature of the event(s) described in the text, along with their possible relationships (temporal or causal).

An example of a TLP analysis that we have developed and tested is shown in Figure 7 where specific elements are identified in the issue report (e.g., degradation [cracks] and specific component [shaft]). The same figure is also displaying a relevant analysis step that we have developed in FY24 to summarize the

nature of the text, which is condensed in this case as: a degradation (of the shaft) was observed from the inspection which could have led (conjecture) to the asset (pump) failure.

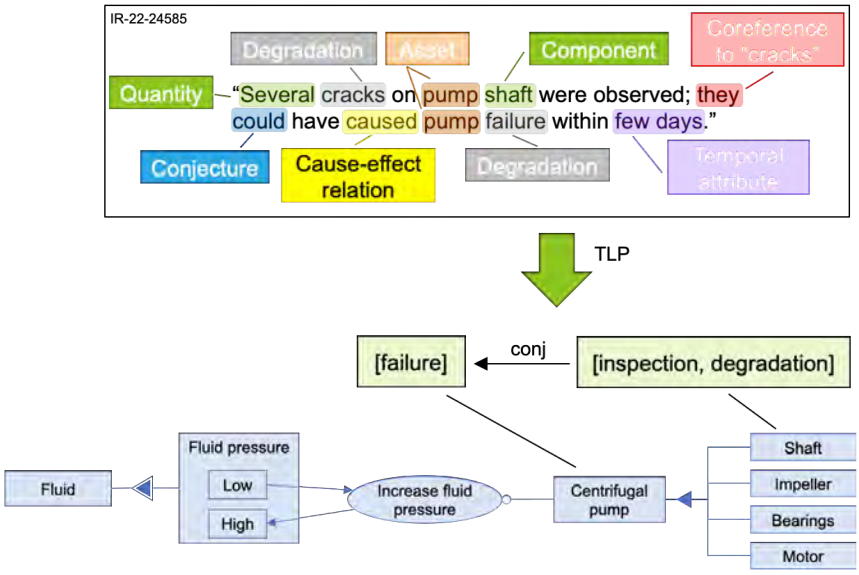

Figure 7. Example of a TLP analysis of textual data and association with MBSE entities.

4.4 Data Fusion

Lastly, after processing ER numeric and textual data (see Sections 4.2 and 4.3), the last step is constructing the knowledge graph. As indicated in Section 1, the purpose of a knowledge graph is to capture system architecture and system historic performance in a human-explorable data structure. Here, human-explorable refers to the capability that the data structure, rather than being a black box, can be visualized and explored. In addition, original ER data is preserved, and it is actively part of the knowledge graph. The methods shown in Sections 4.2 and 4.3 are employed to identify abnormal behaviors (i.e., anomalies from numeric data), or specific features (e.g., degradation phenomena) from textual elements.

A unique feature of our approach is that the system MBSE model is used as the main skeleton of the knowledge graph (see Figure 8) where nodes and edges refer to specific elements and links of the MBSE diagram. A relevant observation to be highlighted here is that each edge in this MBSE-based graph implicitly contains a cause-effect information. Then the processed ER data elements shown in Sections 4.2 and 4.3 are as follows:

- Anomalies detected from ER nuclear data elements (see Section 4.2) are associated with the corresponding MBSE entity being monitored;
- Textual data elements processed using the methods shown in Section 4.3 are associated with the MBSE entities mentioned in the original textual data element.

At this point, provided a system knowledge graph, plant system engineers can now perform the following analyses:

- *Discover patterns behind repetitive occurrences of abnormal events*: the ability to capture the full performance history of an asset, rather the using a one-event-at-a-time mindset, is vital to identify the most appropriate corrective actions.

- *Identify cause-effect relations between events*: 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. Logical relation between event sis here captured through MBSE structure while temporal relation is verified through statical testing methods (see Appendix A for more detail).

Note that the proposed approach is not bound to a specific anomaly detection or knowledge extraction method; we in fact provide well defined application programing interfaces (APIs) such that currently employed methods in plant monitoring and diagnostic (M&D) centers can be easily interfaced. The methods shown in Section 4.2 and 4.3 can be considered as state-of-the-art since they rely on recent data analytics advancements designed to overcome some limitations of current state-of -practice methods.

Note also that the already built knowledge graph (e.g., for the CWS system) can be easily expanded by adding or merging the knowledge graphs developed for supporting systems (e.g., the 4160V AC system for the CWS system). The developed knowledge graph construction process is in fact modular in the sense that a knowledge graph can be constructed for each system, but then these graphs can be merged once the cross-system dependencies are capture in the system MBSE models.

Modularity can be also achieved from a data point of view; additional data sources can be found in each utility such as: outage data (i.e., maintenance and surveillance operations performed periodically during plant outages), asset usage data (e.g., historic number of hours an asset has been running), regulatory related data (e.g., the basic event ID of an asset as part of the plant risk model, or the set of risk-informed plans associated with that asset), and economical data (e.g., procurement and maintenance costs). These data sources can be added to knowledge graph provided a well-defined label to the node in the graph that contains such data. This feature allows multiple stakeholders (e.g., system engineers, plant risk analysts, financial teams) to provide their own perspective of an asset (i.e., operational, regulatory, economical) into a unique and coherent structure designed to overcome current data limitations of nuclear utilities: missing, redundant, or contradictory information.

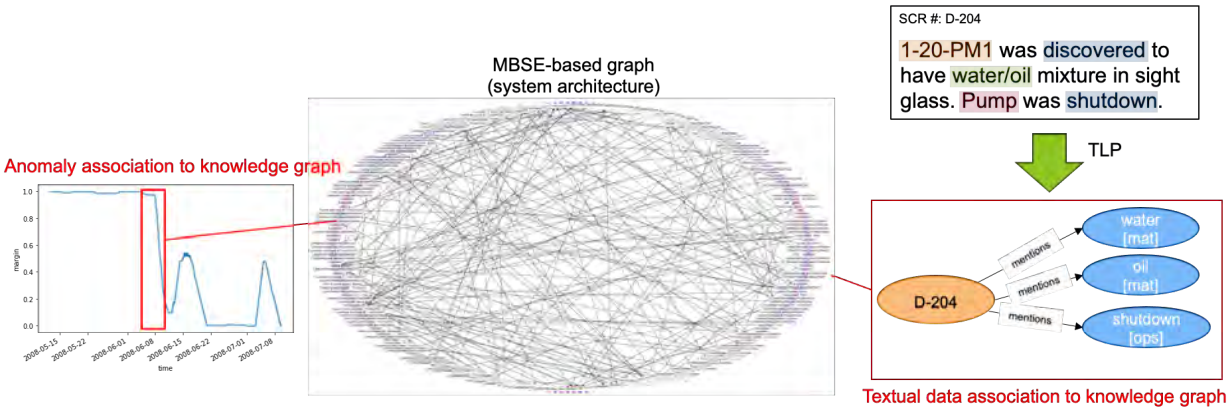

Figure 8. Graphical representation of a knowledge graph where system architecture and system historic performance data is captured in a single graph-based data structure.

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

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

4.5 Methods Development

For sharing the developed methods with the industry partner and extending such sharing to other industry stakeholders, we have framed our development in a set of plant-agnostic workflows. As indicated in Table 1, each of the four developed workflows has been designed with precise specifications in terms of the format of input data. These workflows are being developed and released in the digital analytics, causal knowledge acquisition and reasoning (DACKAR) repository that will be released opens-source soon.

Table 1. List of developed plant-agnostic workflows.

#	Workflow	Input	Output
1	MBSE modeling (see Section 4.1)	Design documents describing system architecture (form and functional description)	MBSE models for the considered system and derived graph structure
2	Anomaly detection (see Section 4.2)	System monitoring data (labeled or unlabeled)	List of inferred anomalies
3	TLP processing (see Section 4.3)	Textual event data (e.g., operator shift logs, condition reports, maintenance reports)	Graph representation of event
4	Knowledge graph construction (see Section 4.4)	Data elements generated in Workflows 1, 2, and 3	Graph structure

5. BENEFITS TO NUCLEAR INDUSTRY

In the few past decades, many high-tech industries (e.g., automotive, aerospace, medical) have demonstrated that the integration of advanced statistical, ML, and artificial intelligence (S-ML-AI) methods with plant data has been able to optimize plant resources (e.g., operating costs, time, personnel), improve the effectiveness of plant operations, and provide robust and informed decisions to decision makers.

Recently, the nuclear industry, including the industry partner that has collaborated with us in this work, has started to explore and evaluate such potentials, and several operational areas may benefit from S-ML-AI methods. Operational areas that would benefit from application of novel data analytics and decision-making technologies include plant operations (e.g., maintenance and outage), aging management, fuel management, and risk and reliability analyses.

The project described in this report directly tackles work reduction opportunities identified in the Light Water Reactor Sustainability Integrated Operation for Nuclear plan (Reemer, 2023) focusing on the direct application S-ML-AI methods on plant-specific areas that would benefit from automation to optimize plant resources. These computational methods are designed to “copilot” (rather than “autopilot”) with plant analysts, system engineers, and maintenance crews to discover the causes behind anomalous behaviors (i.e., automated troubleshooting) and optimize maintenance resources to restore plant and system health.

During FY-24, this project tackled a challenging problem that the nuclear industry is currently facing: the ability to fully analyze all ER data elements (e.g., condition-based monitoring data and anomalous events reported in textual form). Solving this challenge is essential to correctly assess the current and historic performance of systems and assets to enable proactive solutions to maintain highest levels of reliability and availability.

From an economic standpoint, the impact of our work can be quantified in terms of prevention of costly equipment failures and avoidance of potentially long equipment unavailability and even plant outages that could be caused by equipment being out of service. There are also cost savings in day-to-day tasks and activities since the hours required to parse large amounts of data either manually or using current state-of-

practice methods can be significantly reduced when our method is employed by the utility. Since current state-of-practice methods are prone to inefficiencies, the economic impact of this work can be quantified in terms of the loss of resources (e.g., time and money) when anomalous behaviors are not promptly detected or are wrongly diagnosed.

Lastly, we envision that our S-ML-AI methods can directly support the automation of specific plant activities, such as the planning and scheduling of plant operations based on current and historic plant and system performance information. The impact of such a feature can be quantified in terms of better utilized plant resources (e.g., maintenance crews, spare parts) and the hours required for planning and scheduling repetitive plant operations.

6. CONCLUSIONS

This report summarizes activities performed within the advanced modeling and data analytics project focused on he analysis and integration of ER data in presented in both numeric and textual formats. The context for this research is the analysis of ER data to assess system and asset health to support efficient and effective equipment maintenance and health management. A system common to any NPP, the circulating water system, was selected and the collaborating utility and provided a large amount of data and information to support the research. Such data included system schematics, monitoring data, issue reports, maintenance reports, and operator shift logs. The work has been structured in four different directions: analyzing anomalies from monitoring data, analyzing textual data, digital modeling of system architecture, and integrating ER data through a knowledge graph.

The outcome of the research and development activities is the developed knowledge graph: a relational database that captures system architecture (i.e., system design) and integrates all collected data elements (both numeric and textual) to support understanding of the system structure and behaviors, both normal and emergent. The knowledge graphs can support system engineers and other plant personnel with the analysis of the historic equipment performance which in turn enables informed decision-making. This is the first step to support decision-making in a predictive maintenance context where the health status of assets and components needs to be precisely quantified through available data.

Details of the technical approach and methodologies developed within this project are described in a journal article which is presented in Appendix A.

The future plan is to further improve and test the computational tools developed during FY-24. Once this task is complete, developed tools and methods will be released to the industry with an open-source license such that additional nuclear utilities can test and eventually deploy them in their information technology systems. Lastly, our goal is to complete the interfaces between the other computational tools developed under the Light Water Reactor Sustainability Risk-Informed Systems Analysis Pathway designed to perform system reliability modeling (SR^2ML[3]) and optimize plant resources (LOGOS[4] and RAVEN[5]).

This final task will generate a complete suite of computational tools designed to bridge ER data and decisions in a plant operation context.

[3] SR^2ML (Safety Risk Reliability Model Library) repository: https://github.com/idaholab/SR2ML

[4] LOGOS repository: https://github.com/idaholab/LOGOS

[5] RAVEN (Risk Analysis and Virtual ENvironment) repository: https://github.com/idaholab/raven

REFERENCES

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.

Kim S., J.-H. Choi, and N. H. Kim. 2021. "Challenges and Opportunities of System-Level Prognostics." *Sensors*, 21(22): 7655. https://doi.org/10.3390/s21227655

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*, 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, 209(11): 1637–1652. 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: 158–163. https://doi.org/10.1016/j.procir.2014.02.006.

Pecht, M., and M. Kang. 2019. "Introduction to PHM." In *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*, edited by Michael G. Pecht and Myeongsu Kang, 1–37. https://doi.org/10.1002/9781119515326.ch1.

Reemer, J. et al. 2023. "Integrated Operations for Nuclear Business Operation Model Analysis and Industry Validation." Idaho National Laboratory Technical report INL/RPT-22-68671.

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.

Yeh, C.-C., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. Dau, D. Silva, A. Mueen, E. Keogh. 2016. "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets." *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1317–1322. https://doi.org/10.1109/ICDM.2016.0179.

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

Page intentionally left blank

Appendix A

Applying Knowledge Graphs to Track System Reliability Performance

Applying Knowledge Graphs to Track System Reliability Performance

C. Wang, D. Mandelli, C. M. Godbole, V. Agarwal
[a] *Idaho National Laboratory, Idaho Falls (USA)*

M. Movassat, B. Mori, D. Liang, E. Nur, A. Birjandi, B Lobo, N. Jacome
[b] *Ontario Power Generation, Toronto (Canada)*

ABSTRACT

With the goal of maximizing plant reliability and availability, complex systems such as nuclear power plants continuously monitor and record the performance and health status of many components, assets, and systems. Such data may take the form of online monitoring data, condition reports, and maintenance reports, and they can provide system engineers with insights into anomalous behaviors or degradation trends as well as the possible causes behind them and predict their direct consequences. Analyzing such data however poses a few challenges. While some of these challenges are technical in nature (i.e., data are often distributed over several physical servers or databases), others are conceptual (i.e., data elements come in different formats, numeric or textual), and measured values have different scales (e.g., vibration spectra and oil temperature). This paper directly tackles these challenges and focuses on integrating all these data elements to assist plant system engineers in analyzing component, asset, and system performances and optimizing maintenance activities. This integration is performed by extracting knowledge from textual data via technical language processing methods and quantifying system, asset, and component health from numeric condition-based data. We rely on model-based system engineering (MBSE) models of systems and assets to identify their architecture and functional (i.e., cause and effect) relations. Numeric and textual data elements are then associated with an 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 plant 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: Technical language processing, predictive maintenance, MBSE, data fusion

Target journal: Reliability Engineering and System Safety

List of acronyms

API	application programing interfaces		NLP	natural language processing
CWS	circulating water system		NPP	nuclear power plants
DACKAR	digital analytics, causal knowledge acquisition and reasoning		OPM	object-process methodology
DBSCAN	density-based spatial clustering of applications with noise		PHM	prognostic and health management
ER	equipment reliability		RLM	robust linear models

IR	incident report		SSC	system, structure, and component
LML	lifecycle modeling language		SysML	systems modeling language
MBSE	model-based system engineering		TLP	technical language processing
ML	machine learning		UML	unified modeling language
MMD	maximum mean discrepancy		WO	work order
M&D	monitoring and diagnostic			

1 INTRODUCTION

The rapid development and deployment of advanced condition-based monitors and data analytics techniques (e.g., anomaly detection, diagnostic, prognostic methods) is helping system engineers and plant operators monitor the performance of several assets that constitute complex systems. Similarly, digitizing operation and maintenance activities allows the engineers and operators to track events at the system or plant level (e.g., plant planned shutdown or system taken out of service) and, more importantly, observe asset abnormal conditions and operations that have been performed on such assets (Coble, 2015; Xingang, 2021). As a drawback, engineers and operators are now facing the challenge of processing the amount of equipment reliability (ER) data being continuously generated, which is not only extremely large but also appears in different forms: textual and numeric.

This paper addresses this challenge by presenting methods to assist engineers and operators in extracting knowledge from ER data. The first point we claim here is that all the ER data elements described earlier equally provide indications about asset and system performance and, hence, cannot be analyzed separately. The second claim is that generating knowledge from data requires the ability to put data into "context." Here, context is the additional piece of information needed by ER data analysis tools to understand what these data elements are referring to.

Here, we employ model-based system engineering (MBSE) models of systems and assets to capture their architectural (i.e., physical) and functional (i.e., cause-effect) relations. With that, ER data elements (both textual and numeric) 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, but it is more intricate for the text-based data. We employ technical language processing (TLP) methods to "extract knowledge" from textual elements. Filtering abnormal behaviors can then be performed using numeric (through anomaly detections and diagnostic and prognostic methods) and textual elements (by understanding their semantic nature). The abnormal behavior instances, which are associated with a specific MBSE element, are then stored in a relational database. Such a database takes the form of a graph where the main skeleton is the system's MBSE model and abnormal instances are "linked" to the modeled system elements. At this point, both numeric and textual data elements are integrated and put into context. From here, graph-based analysis methods can be employed to perform "machine reasoning," which includes identifying abnormal patterns and the root cause behind such patterns.

For clarity purpose, throughout this paper, we employ the term *system* to indicate a collection of assets designed to provide a specific function (e.g., to generate alternating current power or provide high-pressure injection during a loss-of-coolant accident). The term *asset* indicates a system element designed to support the system function (e.g., a diesel generator, motor-operated valve, or centrifugal pump). A *component* denotes an asset sub-element (e.g., a transmission gear in a diesel generator, the drive sleeve of a motor-operated valve, or the impeller of a centrifugal pump). Components are subject to degradation/aging and may require maintenance to guarantee proper operation of the asset.

Since this work was performed in collaboration with a nuclear utility, the data elements and corresponding figures reported in this paper have been intentionally altered to hide proprietary information.

However, the computational methods and algorithms described in this paper can be found in the Digital Analytics, Causal Knowledge Acquisition and Reasoning (DACKAR) GitHub repository[1]. In addition to the source code, this repository also contains the full workflows (shown as Jupyter notebooks) described in this paper.

2 ER DATA TAXONOMY

As indicated in Section 1, NPP ER data can be heterogenous in nature (e.g., numeric, textual, sound, 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[2] (i.e., the emergence property [Borky, 2018]).

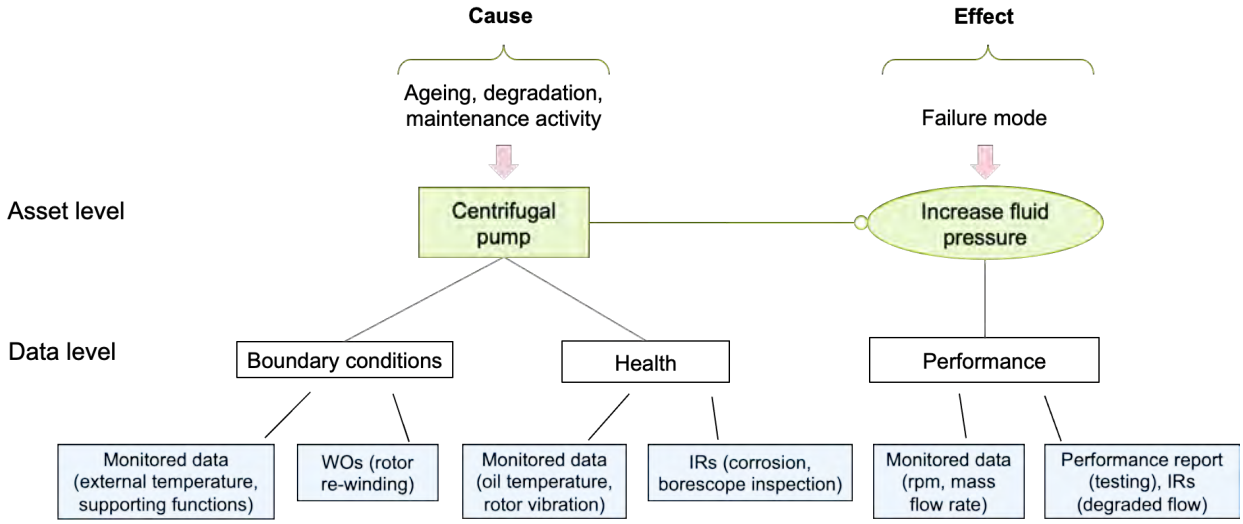

Figure 1. Taxonomy of ER data presented through a cause-effect lens.

For example, when considering a centrifugal pump (see Figure 1), the form element *Centrifugal pump* consists of all the components that make up the considered asset (e.g., motor, stator, shaft, and impeller), and the function element *Increase fluid pressure* 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—health monitoring and boundary condition monitoring—and can be either numeric or textual as well. Note 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 affects 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 with the identification of anomalous behaviors.

---

[1] DACKAR GitHub repository link: https://github.com/idaholab/DACKAR

[2] In many situations, an asset might be supporting multiple functions and it 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.

3 CONSIDERED SYSTEM

The system under consideration in this paper is the circulating water system (CWS) of an existing nuclear power plant. Typically, this system is used in many types of power plants (e.g., coal, gas, oil) and is designed to remove the residual heat from the turbine-condenser system and release it into the environment. In our case, water is collected from a body of water (e.g., lake or river) through service gates. Then, using traveling screens, the water is cleaned of debris, aquatic life, and foreign bodies that might damage CWS components. Screen wash pumps provide spray water to remove accumulated debris on the screens. The CWS also contains a vacuum priming system that removes any air from the system. Then, water is pumped through heat exchangers located in the plant secondary loop and removes heat from the turbine-condenser system. Lastly, warm water is released downstream of the same body of water. Depending on the environmental conditions, a portion of warm water is released back into the service gates to avoid ice formation that would block water flow. Several systems support the CWS, such as alternating current (AC) systems (4,160 and 480 V) and water-cooling systems. From an operational standpoint, even though the CWS does not directly support a plant nuclear safety function, any performance degradation or abnormal behaviors may directly affect power generation (either in terms of power derate or power shutdown) and, consequently, plant economic revenues.

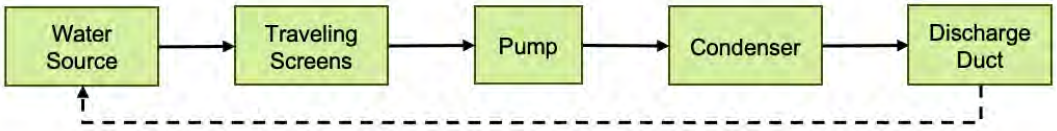

Figure 2. Simplified schematics of the CWS structure.

4 ER DATA

This section describes the large amount of data that has been collected throughout the 2012–2022 time frame used in this research. This data consisted of both condition-based monitoring data (numeric in nature) and a large set of condition-based report data (textual in nature). Sections 4.1 and 4.2 provide details about numeric and textual data elements, respectively.

4.1 MONITORING DATA

CWS operation has been continuously monitored to detect early signs of degradation and proactively perform maintenance to restore system operations and guarantee system availability. In this respect, Table 1 provides a list of the available monitoring variables collected over the past decade; note these variables not only provide indications of the performance of the CWS pumps and condenser but also of systems interfacing with the CWS. Note that plant environment variables are also available (water body and air temperature); Section 7 provides considerations about the importance of environment variables to remove seasonal (i.e., periodic) trends from CWS plant monitoring variables when performing anomaly detection.

Table 1. List of CWS monitoring variables.

Variable IDs	Description
$x_1^{pump,unit}, \ldots, x_4^{pump,unit}$	Monitoring variables associated with CWS pumps of a specific plant unit
$x_5^{cond,unit}, \ldots, x_9^{cond,unit}$	Monitoring variables associated with the condenser of a specific plant unit
$x_{10}^{unit}, \ldots, x_{12}^{unit}$	Monitoring variables associated with systems interfacing with the CWS
T_{water} and T_{air}	Plant environment variables

To ensure the raw data were organized, processed, and cleaned, several steps were conducted. All missing data points for the CWS pump, condenser, system, and environment were filled with the previously available data point. All monitoring variables and plant variables were further normalized using z-score. Normalizing using z-score helps protect sensitive plant information and easily capture any noisy data points that do not directly correspond to CWS anomalies. Figure 3 shows the temporal profile of two of the monitored variables listed in Table 1 over the considered 10-year lifespan.

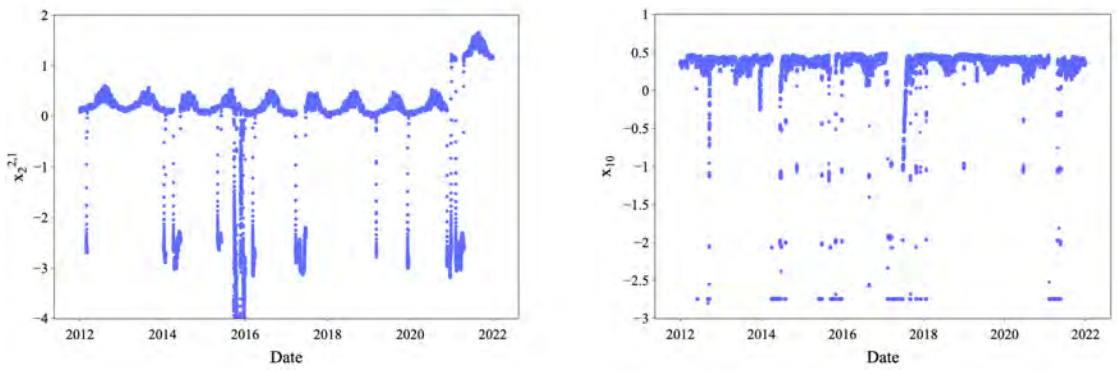

Figure 3. Temporal profile of two monitored variables after the cleaning process.

4.2 TEXTUAL DATA

In addition to the numeric data described in Table 1, the considered nuclear power plant has also recorded in its databases all operational events as follows:

- *Reactor operator shift logs* of events related to the CWS system;
- *CWS condition reports*: abnormal events that occurred in the CWS;
- *CWS work orders*: maintenance operations performed on the CWS;
- *Plant outage data*: time instances where the plant was shut down for either planned or unplanned outages.

Note that all these events recorded in textual form (while the data elements described in Table 1 are in numeric form) provide indications not only about the historical reliability performance of the CWS but also precise information about the nature of the recorded abnormal events and corresponding operations

performed to restore CWS operations. Additionally, the textual information helps in capturing durations of any anomalies to understand correlations with numeric data distributions.

In addition, a set of design and plant operations documents were provided, which gave us precise information about the architecture and functional relations between the CWS, the rest of plant, and the assets that are part of the CWS. Lastly, plant staff provided us with a list of acronyms and abbreviations typically used in textual data along with the ETAG list which provides an indication of the unique ID associated with each CWS asset and component.

5 ANALYSIS OF ER DATA

Figure 4 shows our approach to process and analyze CWS historical performances provided the ER data elements (numeric and textual) described in Section 4. Constructing the knowledge graph starts by performing four different workflows:

- *Step 1: MBSE workflow.* System architecture information provided by the plant and CWS design documents is translated into MBSE models (see Section 6).
- *Step 2: Numeric workflow.* CWS anomalies are inferred from CWS numeric monitoring data (see Section 7).
- *Step 3: Textual workflow.* CWS-related events reported from operator shift logs, conditions, or maintenance reports are processed using TLP methods (see Section 8).
- *Step 4: Event to time series correlation analysis.* Based on the temporal occurrence of the inferred anomalies (see Step 2) and reported events (see Step 3), we test whether the occurrence of these events had a cause-effect relation with observed monitoring data (see Section 9).
- *Step 5: Knowledge graph construction.* The construction of the knowledge graph (see Section 10) starts by translating the system MBSE model into a graph structure where each node of this graph is a physical entity of the CWS (e.g., pump, traveling screen). Each edge in such a graph represents a physical connection between two entities where the nature of such a connection can be of different types (mechanical, electrical, hydraulic, digital). Then, the set of anomalies derived from Step 2 and the events processed in Step 3 are digitally associated with one (or more) node of the graph derived from the MBSE model (see Section 6).

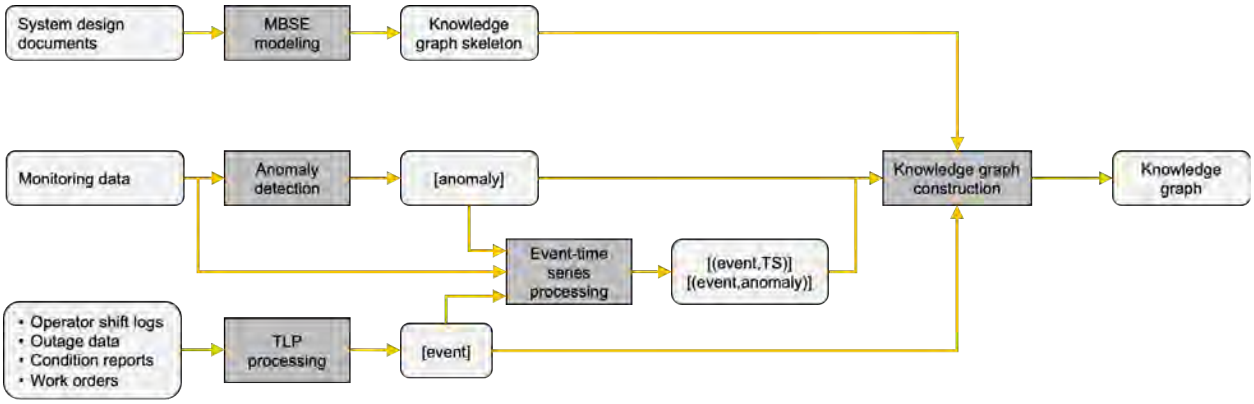

Figure 4. From ER data to knowledge graph: a graphical description of the workflows. Analysis methods are highlighted in dark grey while generated and input data are highlighted in light grey.

Table 2. Functional description of workflows designed to construct a knowledge graph from ER data.

#	Workflow	Input	Output
1	MBSE modeling (see Section 6)	Design documents describing system architecture (form and functional description)	MBSE models for the considered system and derived graph structure
2	Anomaly detection (see Section 7)	System monitoring data (labeled or unlabeled)	List of inferred anomalies
3	TLP processing (see Section 8)	Textual event data (e.g., operator shift logs, condition reports, maintenance reports)	Graph representation of event
4	Event to time series correlation analysis (see Section 9)	System monitoring data, anomalies identified in Workflow 2, events processed in Workflow 3	List of events correlated to time series variations; list of events correlated to identified anomalies
5	Knowledge graph construction (see Section 10)	Data elements generated in Workflows 1, 2, 3, and 4	Knowledge graph structure

6 SYSTEM DIGITAL REPRESENTATION

The ability of system engineers to effectively 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 via a set of diagrams. The most commonly used languages are: Unified Modeling Language (UML) (Booch, 2005), Object-Process Methodology (OPM) (Dori, 2002), Lifecycle Modeling Language (LML) (LML, 2022), and Systems Modeling Language (SysML) (Friedenthal, 2008).

For the scope of this project, we have chosen LML and OPM since they provide the basic modeling elements we sought and because—more importantly—digital data structures (i.e., graphs) can be automatically generated from LML and OPM diagrams. Each element of an OPM and LML 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 form elements in an OPM diagram are connected to each other through a set of *links* designed to convey precise meanings (Dori, 2002).

Figure 5 shows the LML diagram of the considered CWS. Note that each asset included in the LML diagram of the CWS may be further described by its own separate LML or OPM diagram. In other words, a network of LML and OPM diagrams can be constructed to refine and further detail the architecture of the considered system. For example, in the CWS LML diagram in Figure 5, 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 6.

Once the MBSE models (LML or OPM) have been developed, they are saved into file. While OPM models can be saved in human readable files (textual form), LML diagrams (which are here developed using the MBSE tool Innoslate) can be saved into `xml` files. The files containing the OPM and LML diagrams are then converted into graphs using the methods available in DACKAR; here, we rely on the `Neo4j`[3] library to construct these graphs.

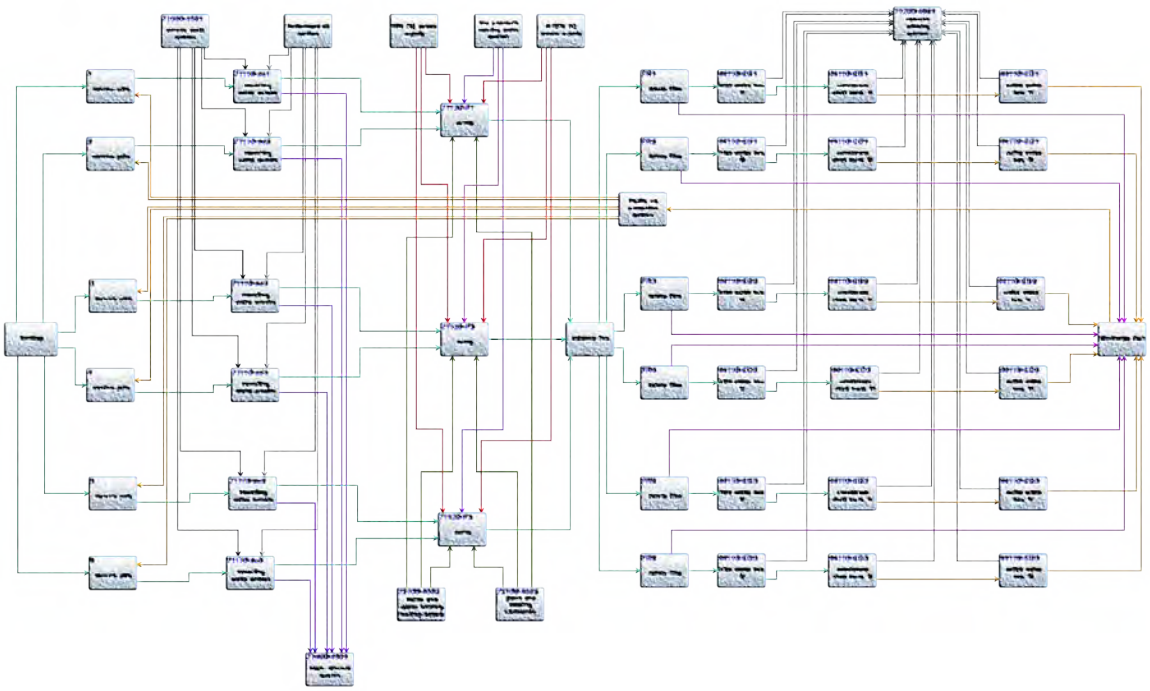

Figure 5. LML model of the considered CWS, which has been intentionally edited to hide any proprietary information.

7 ANALYSIS OF NUMERIC DATA

The amount of anomaly detections (applied to any scientific or technological context) available in the open literature is vast, and it is not within the scope of this paper to provide an exhaustive overview of such methods to compare performances for the considered system. Such methods can rely on classical statistical, ML, or deep learning methods with different pros, cons, and ranges of operability. The main requirements for the choice of anomaly detection methods were fast computation, ability to deal with periodic patterns and missing data, ability to identify anomalies defined over time instance or time intervals, scalability, and interpretability.

[3] Neo4j official site: https://neo4j.com

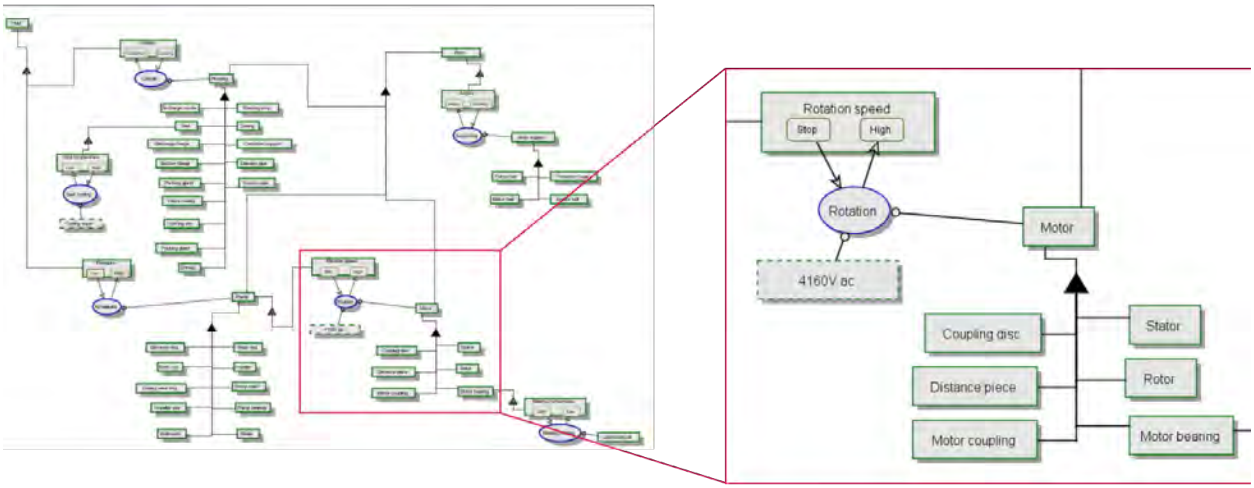

Figure 6. OPM representation of a generic centrifugal pump (left), which also includes supporting systems (the 4,160 V AC system).

7.1 ANOMALY DETECTION THROUGH ML MODELS

ML has seen an exponentially rise in use in a large variety of fields, including the nuclear industry, and has been widely used for real-time monitoring from predictive maintenance, thermal hydraulic computations and nuclear design (Zhao, Shivran, Salko, & Guo, 2020) (Godbole, Delipei, Wu, Avramova, & Rohatgi, 2022). ML algorithms can behave as a universal approximator for both linear and nonlinear relations when the physics may be unknown or complicated to model and have high computational speed even for large quantities of data. ML can be classified as either supervised, semi-supervised, or unsupervised based on the model training regime. Supervised learning algorithms are used when the data available has both input and output features and the ML algorithm learns the relation between input and output. Unsupervised learning algorithms are used when there is no known target or output feature, and the ML algorithm works on learning structures, clusters, and patterns in the input data. A semi-supervised learning algorithm is a combination of both supervised and unsupervised learning algorithms used when there is a lot of unlabeled data and some labeled data.

This work uses DBSCAN, which belongs to the unsupervised ML category, to detect anomalies in multivariate data. DBSCAN is used on a large amount of data containing four system variables and data features. DBSCAN algorithm is used to detect anomalies and irregularities in the data by clustering all normal data points together and capturing and clustering all anomalies and irregularities as an anomaly cluster. The entire data used for DBSCAN is further broken down into smaller chunks to ensure optimal clustering by DBSCAN and to accurately capture all anomalies within the data without the need for hyperparameter tuning over each individual year. DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) is a commonly used clustering algorithm belonging to the unsupervised ML category and is widely used for detecting outliers and anomalies in data (Çelik, Dadaşer-Çelik, & Dokuz, 2011). DBSCAN uses proximity parameters to cluster data points that are close together. The DBSCAN algorithm requires two parameters, epsilon, which specifies the distance between two points for them to be considered neighbors, and minimum points, which states that that there should be at least those set number of minimum points at epsilon to be considered part of cluster. Epsilon can be considered as a radius for two-dimensional data, minimum points then finds all the data points that have data points equal to minimum points within the radius epsilon and categorizes them together. Figure 7 shows the methodology of clustering by DBSCAN when the minimum points is set at four. All points belonging to a cluster are termed core points, shown by red dots in Figure 7. A core point is any point that has minimum data points with a radius epsilon around it. All data points belonging to a cluster and that are around a core point but cannot themselves be termed as a core point are

called border points, as depicted by yellow circles in Figure 7. All border points belong to the same cluster as the core point surrounding it at distance epsilon. Any data point not satisfying the minimum data points at distance epsilon criteria are then categorizes as an anomaly data point by the DBSCAN algorithm as detected by the blue circle in Figure 7.

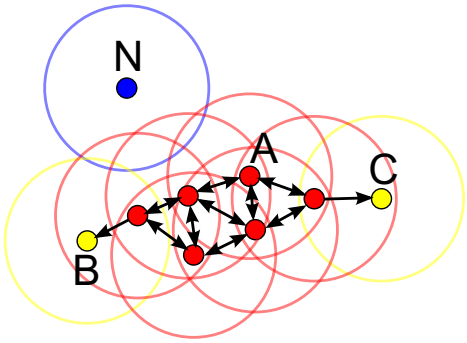

Figure 7. DBSCAN methodology for clustering data points.

This is how DBSCAN is utilized in this work to capture anomaly data points in multivariate monitoring variables for a CWS pump. Epsilon and minimum data point parameters are manually tuned to capture the optimal values that work for data for a large duration of data. Data is split into smaller chunks to ensure the robustness and generalizability of the DBSCAN without the requirement for hyperparameter tuning, and DBSCAN is applied to capture anomalies and normal data points. After manual hyperparameter tuning, the optimal values of epsilon and minimum points that work on various sections of the dataset containing four system variables are 1 and 2,000, respectively. Pump 1 CWS monitoring variables for a particular subsection are shown in Figure 8 with the x-axis depicting a variable of time.

Similarly, Figure 9 shows the application of DBSCAN for a different subset of monitoring variables for CWS Pump 2, These anomalies match accurately with the true anomalies seen in the system variable data for the CWS, as can be seen through system textual information and visually as these anomalies have a clear trend that is out of the normal ranges, behavior, and distribution of the system variable trends.

Thus, DBSCAN was successfully able to predict the anomalies without needing hyperparameter tuning for different sections of data and for different pumps, showing the robustness of the algorithm, as seen in Figure 8 and Figure 9, DBSCAN was successfully able to capture and categorize all anomalies in the original dataset as well as the dataset with all seasonal variations removed based on environment variables, which coincides with the anomalies seen in textual information on the CWS.

7.2 DATA SEASONALITY DETRENDING

From the initial look of the temporal profile of the obtained monitored variables (see Figure 3), it is possible to observe periodic patterns that are due to seasonality effects. In particular, monitored data variance is higher during the summer season and lower during the winter season. This can negatively impact the ability of the anomaly detection method to correctly identify equipment anomalies and distinguish anomalies that are actually due to environmental conditions. Thus, we have started to look at ways to remove the seasonality effect from plant monitoring data.

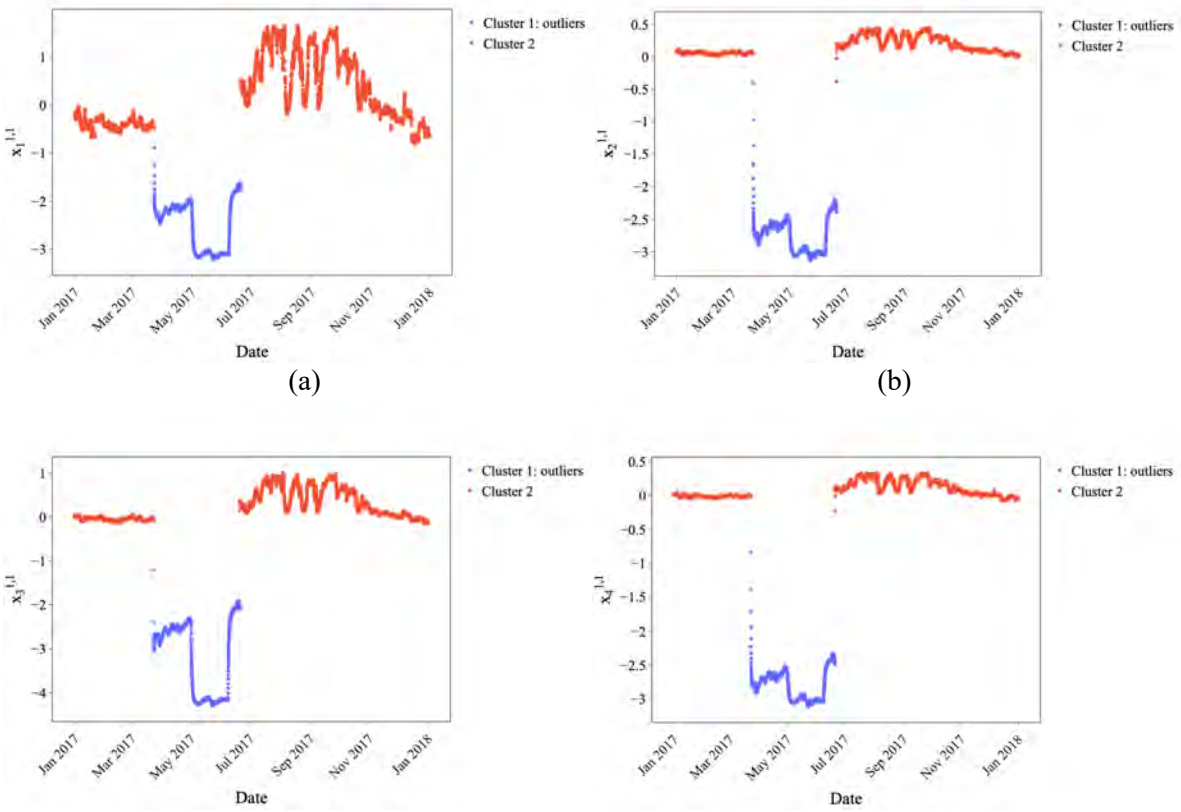

Figure 8. DBSCAN results for CWS pump monitoring variables denoted by (a) $x_1^{1,1}$, (b) $x_2^{1,1}$, (c) $x_3^{1,1}$ and (d) $x_4^{1,1}$.

A favorable element here is that air and water body temperature have been monitored, and their temporal profiles are also available (see Table 1). Then, we have explored the statistical correlation between CWS monitoring variables and environmental variables; such an analysis has highlighted that a strong correlation exists between CWS monitoring variables and water lake temperature. An example is shown in Figure 10 where the temporal profiles of a CWS monitoring variable and lake water temperature are compared. More importantly, the same figure shows that a linear correlation exists between these two variables. Given this, we have explored methods to remove data seasonality from CWS monitoring data using environmental variables using regression models.

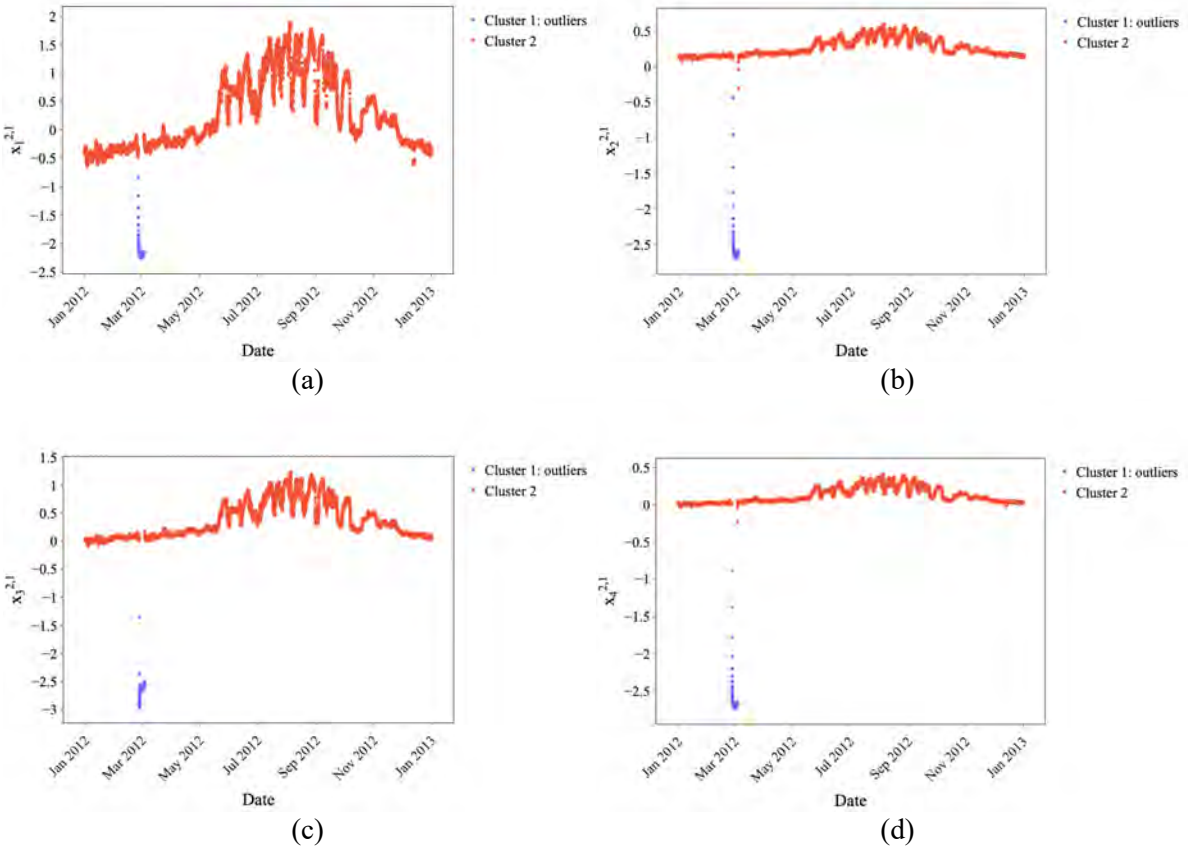

Figure 9. DBSCAN results for CWS pump monitoring variables denoted by (a) $x_1^{2,1}$, (b) $x_2^{2,1}$, (c) $x_3^{2,1}$ and (d) $x_4^{2,1}$.

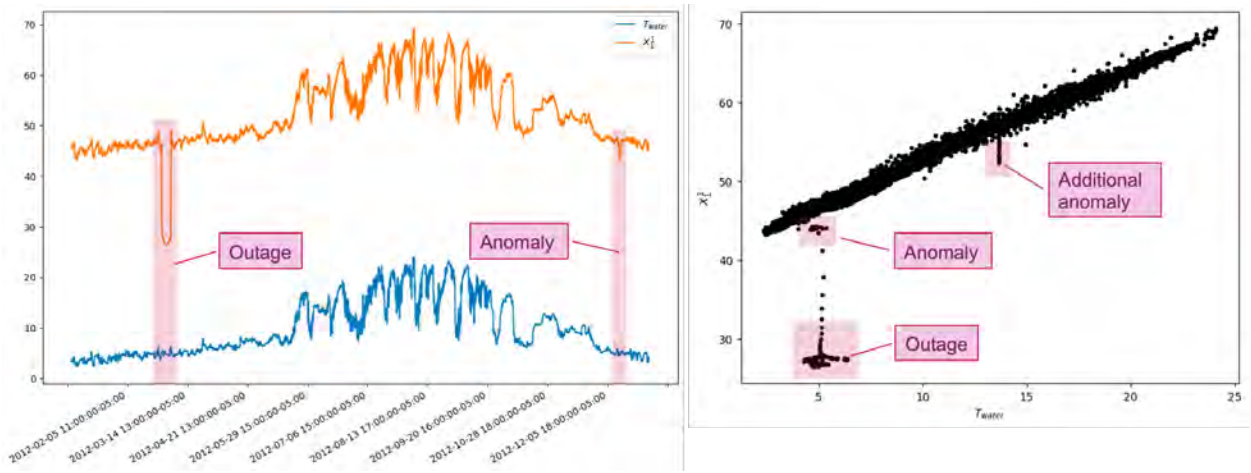

Figure 10. Comparison of the temporal profile of a CWS monitored parameter and lake water temperature (left plot) and correlation analysis of the same two parameters.

Traditional regression models aim at finding a relationship between an independent variable and a dependent variable. Robust linear models (RLMs) aim at overcoming the drawbacks of linear models by

being able to handle the outliers present in the data. This work uses an RLM with the Huber loss (Huber, 1981) to find the linear model that computes monitoring variables of the CWS pump as a function of T_{water}. The goal is to remove all the seasonal effects in system variable data distributions, which are easily captured in the T_{water} distributions using the RLM model to compute the function shown in Equation 1. $x_i^{pump,unit}$ indicates the pump monitoring variables with i ranging from 1 to 4, as computed by the RLM model that computes the linear function as a function of T_{water} as shown in Equation 1.

$$x_i^{pump,unit} = RLM(T_{water}) \tag{1}$$

To remove the seasonal variations in the system variables, Equation 2 is applied. $x_i^{pump,unit"}$ indicates the system variables without any seasonal variations and is computed by subtracting $x_i^{pump,unit}$, which is the output from the RLM model that computes the relationship between pump monitoring variables and T_{water}, from Y_{system}, which denotes the original values of system variables including seasonal variations:

$$x_i^{pump,unit"} = Y_{system} - x_i^{pump,unit} \tag{2}$$

Figure 11 shows the RLM prediction results as shown in Equation 1 along with the residual on the secondary y-axis as shown in Equation 2. The RLM prediction is shown by the red dotted line, and the residual prediction on the secondary y-axis is shown in green. This shows that the RLM results match accurately with the true variations of system variables and environment variables except for the anomalies. Thus the residuals are a good measure to capture the anomalies present in the data.

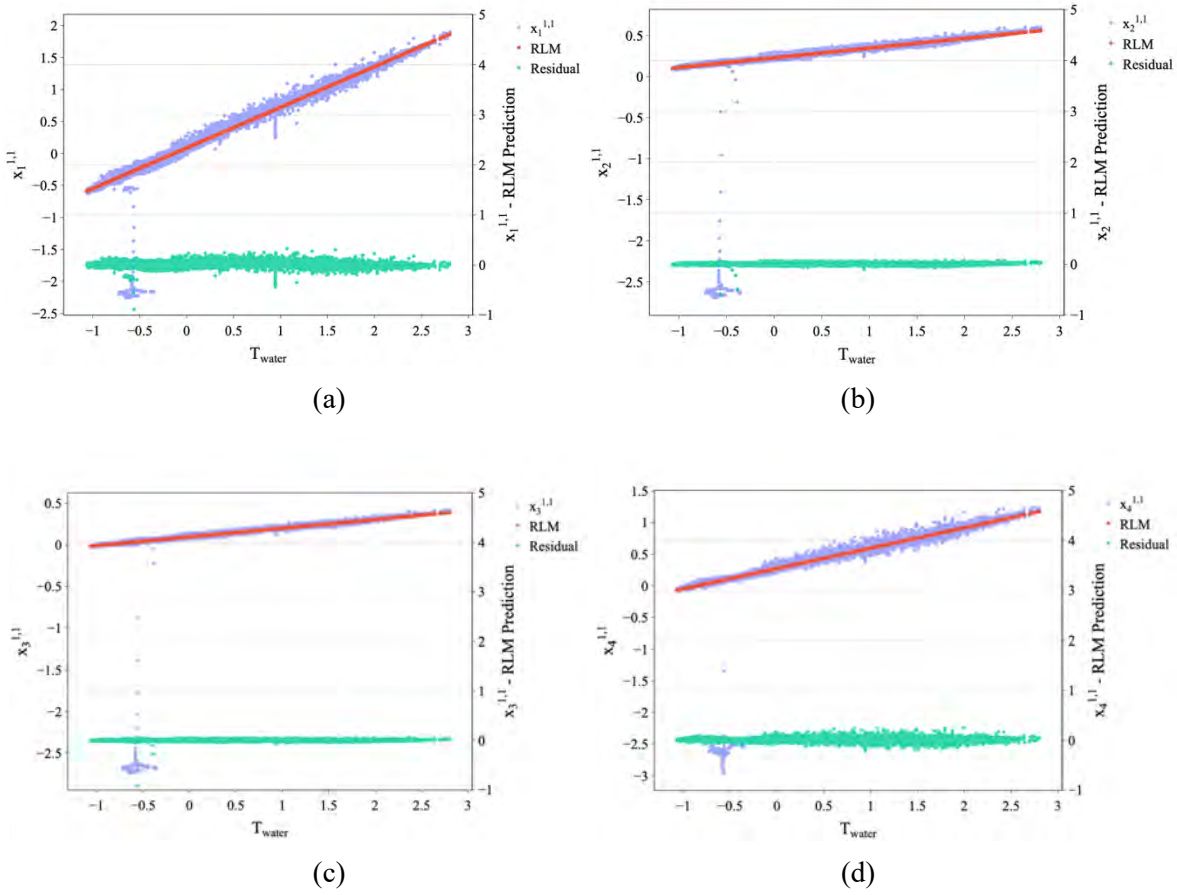

Figure 11. RLM predictions and residuals pump monitoring variables.

DBSCAN is then applied on the residual values to capture the anomalies. The DBSCAN used for original data is used without any hyperparameter tuning needed with epsilon 1 and minimum points as 2,000. Figure 12 shows the DBSCAN results on the residuals computed using RLM for all four system variables.

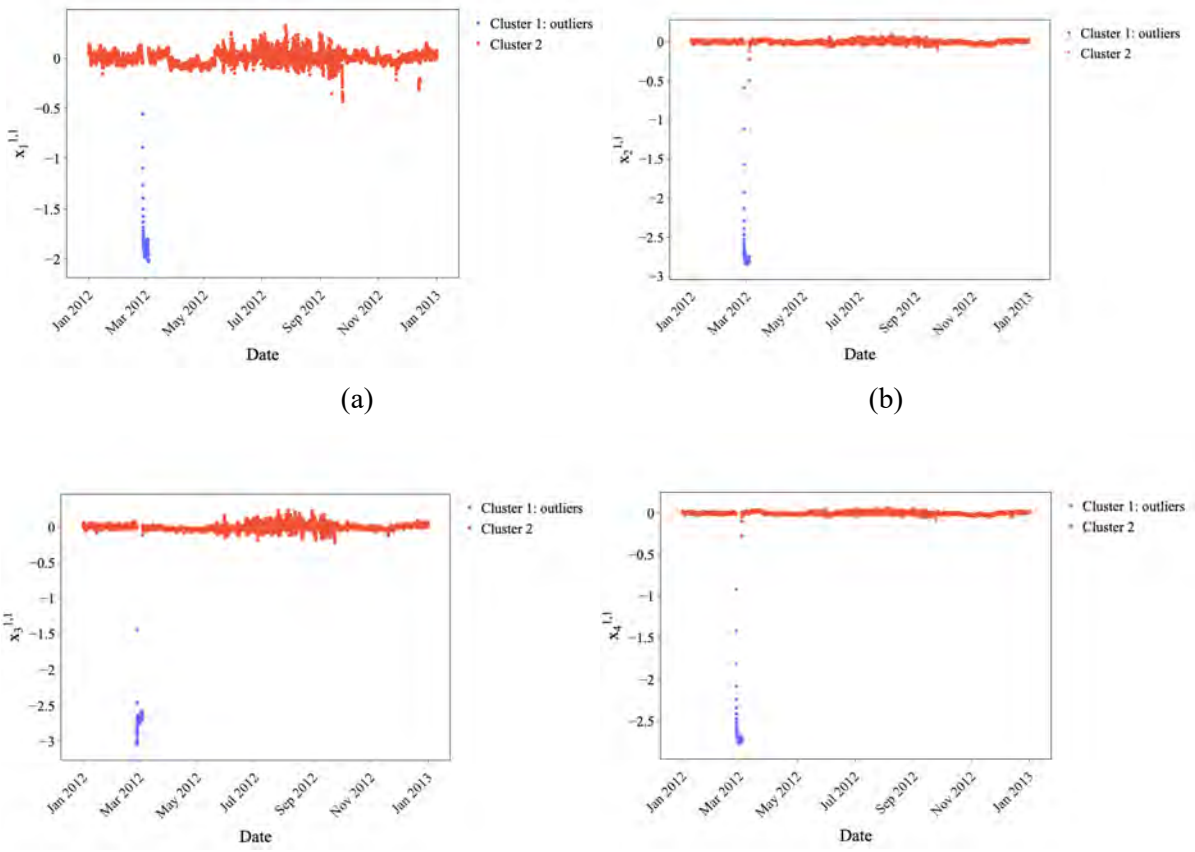

Figure 12. DBSCAN results for residuals computed by RLM to remove seasonal variations for pump monitoring variables.

Thus, DBSCAN was successfully able to predict the anomalies without the need for hyperparameter tuning for different datasets and different distributions, showing the robustness of the algorithm. DBSCAN was successfully able to capture and categorize all anomalies in the original dataset as well as the dataset with all seasonal variations removed based on environment variables, which coincides with the anomalies seen in textual information on the CWS.

7.3 ANOMALY DETECTION THROUGH MATRIX PROFILE

The matrix profile is a time series annotation computed from a time series that can be used to identify motifs and discords (anomalies), corresponding to recurring patterns (or similar subsequences) and outliers, respectively (Yeh, 2016). For example, the lowest points on the profile correspond to the time series motifs, and the highest points correspond to the time series discords. In simple terms, this algorithm is a distance-based approach over a sliding window; here, the considered time series is progressively scanned by

identifying the smallest distance between the portion of the time series limited within the considered time window and the set of time windows previously processed. In summary, the steps for computing the matrix profile from a time series $T \in R^n$ with length n are:

1. Select a time window or subsequence $T_{i,m} \in R^m$ with length m where $m < n$ from T, which is a sequence of m contiguous elements with indices from position i to $i + m - 1$ in T;
2. Compute the Euclidean distance profile D_i^T between a given subsequence $T_{i,m}$ and every other subsequence $T_{j,m}$ from T, i.e., $D_i^T = dist\left(T_{i,m}, T_{j,m}\right)_{j=1}^{n-m+1}$;
3. Compute the matrix profile value for $T_{i,m}$, which value is the minimal distance between $T_{i,m}$ and its nontrivial neighbors $\left\{T_{j,m}\right\}_{j=1,j\neq i}^{n-m+1}$ using D_i^T (i.e., the distance between $T_{i,m}$ and its nearest nontrivial neighbor, denoted as $nn_{nt}\left(D_i^T\right)$);
4. Identify the matrix profile index value I_i^T for $T_{i,m}$ (i.e., the index of the nearest nontrivial neighbor for $T_{i,m}$);
5. The matrix profile for time series T becomes a vector that stores $nn_{nt}\left(D_i^T\right)_{i=1}^{n-m+1}$, and the matrix profile index becomes a vector that stores $\left(I_i^T\right)_{i=1}^{n-m+1}$.

The matrix profile technique can identify unusual patterns caused by unexpected events or deviations from the normal behaviors in time series. An example of anomalies detected is shown in Figure 13 where the matrix profile algorithm from Law (2019) has been applied to the time series of two monitored variables of the CWS. Here, two time series are considered (shown in blue in Figure 13), T_{water} and X_1^{pump}, and the corresponding temporal matrix profiles are shown in red in Figure 13. Anomalies are identified by looking at the regions characterized by high values of the matrix profiles.

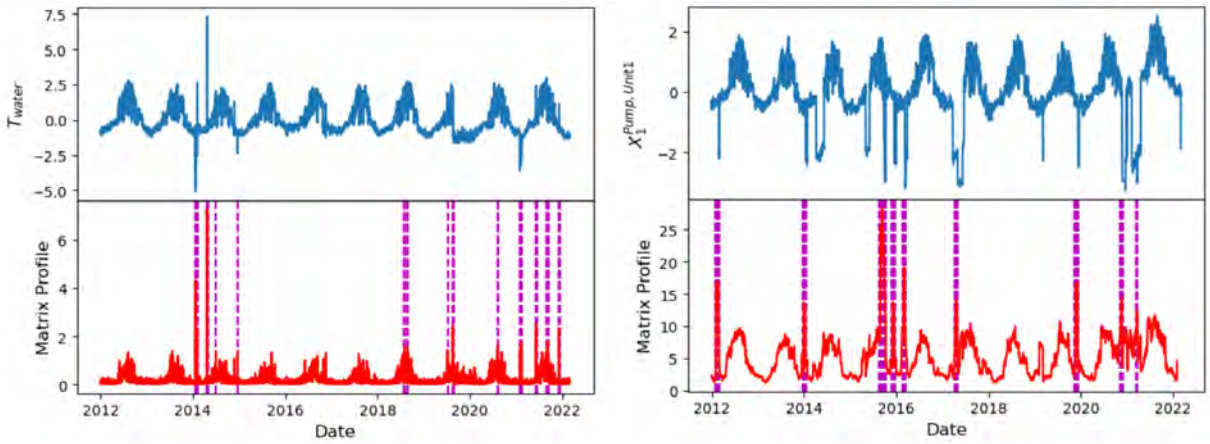

Figure 13. Example of anomalies detected by the matrix profile algorithm when applied to the time series of the monitored variable X_N of the CWS.

Moreover, matrix profile algorithms from Law (2019) also support for finding outliers and anomalies in streaming data. As illustrated in Figure 14 and Figure 15, the preprocessed (i.e., remove outliers) normalized normal monitoring data from 2012 to 2016 is used to train the matrix profile and identify the matrix profile threshold value for anomalies. For T_{water}, the maximum value of training data matrix profile is used as the threshold, while for X_1^{pump}, the 99.9 percentile value is used as the threshold instead to counteract the large matrix profile values introduced to remove the larger size of subsequence anomalies.

We have computed the matrix profile of test data (i.e., monitoring data from 2017 to 2022) against training data, and we have highlighted the identified anomalies using dash dotted line in the given time series. As observed from both Figure 14 and Figure 15, matrix profile can be used to identify both large periods of anomalies and small period spikes and dips.

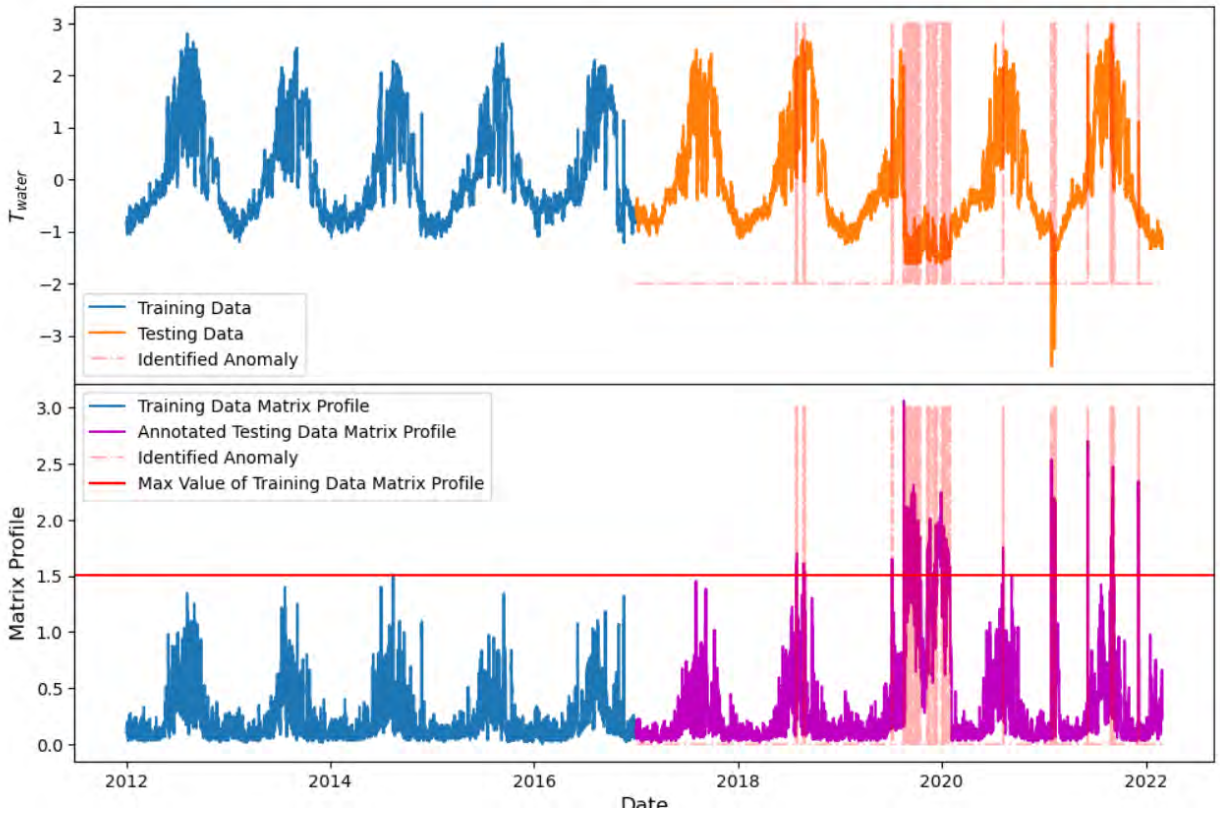

Figure 14. Apply matrix profile for streaming data of T_{water} to detect anomalies.

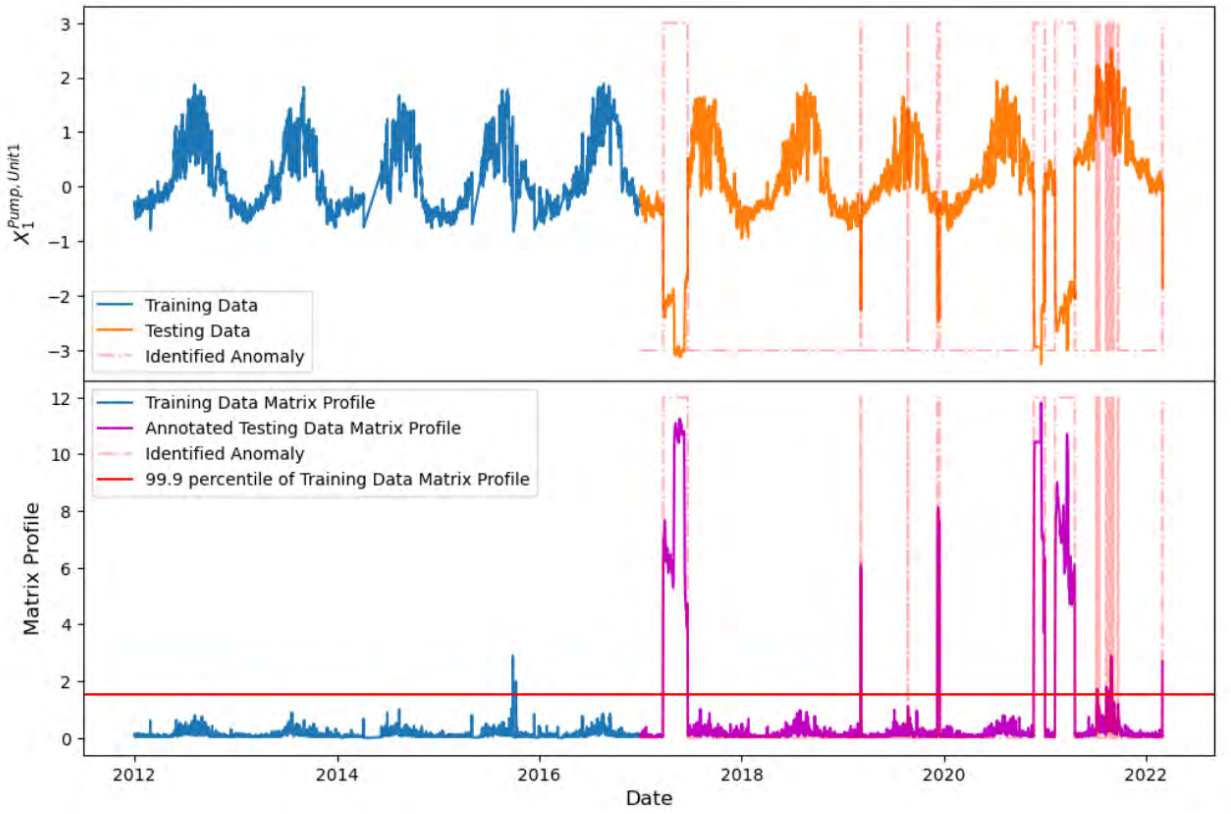

Figure 15. Apply matrix profile for streaming data of X_1^{pump} to detect anomalies.

Apart from selecting the window size, the matrix profile can be used in a simple manner without any tuning.

Once an anomaly is detected, it is digitally recorded by observing the time interval under which it is detected and the set of variables employed to detect it. More precisely, a generic anomaly An is defined as a specific entity, which is defined as:

$$\mathtt{An} = (\mathtt{[vars]},\ t_{in},\ t_{fin}) \tag{3}$$

where [vars] corresponds to the list of variables under which the anomaly was observed and t_{in} and t_{fin} define the temporal duration of such an anomaly.

Table 3. Temporal duration for identified anomalies via matrix profile.

	t_{in}	t_{fin}		t_{in}	t_{fin}
	2018-07-28 01:00:00	2018-07-28 03:00:00		2017-03-22 20:00:00	2017-06-20 14:00:00
	2018-07-30 14:00:00	2018-07-31 07:00:00		2019-03-05 23:00:00	2019-03-08 15:00:00
	2018-08-23 09:00:00	2018-08-23 17:00:00		2019-12-08 05:00:00	2019-12-17 16:00:00
	2018-08-27 19:00:00	2018-08-27 19:00:00		2020-11-20 23:00:00	2020-12-31 12:00:00
	2019-07-06 23:00:00	2019-07-07 02:00:00		2021-02-05 23:00:00	2021-04-18 14:00:00
	2019-08-17 08:00:00	2019-08-18 03:00:00		2021-07-07 16:00:00	2021-07-08 16:00:00
	2019-08-24 15:00:00	2019-08-28 19:00:00		2021-08-09 21:00:00	2021-08-10 20:00:00
	2019-08-29 19:00:00	2019-09-01 06:00:00		2021-08-18 16:00:00	2021-08-18 23:00:00
	2019-09-02 03:00:00	2019-09-05 13:00:00		2021-08-21 07:00:00	2021-08-21 09:00:00
	2019-09-05 17:00:00	2019-09-05 23:00:00		2021-08-21 14:00:00	2021-08-22 03:00:00
	2019-09-06 06:00:00	2019-09-09 03:00:00		2021-08-26 22:00:00	2021-08-30 12:00:00
	2019-09-10 02:00:00	2019-09-14 01:00:00		2021-08-30 14:00:00	2021-08-30 17:00:00
	2019-09-15 21:00:00	2019-09-24 02:00:00		2022-02-28 20:00:00	2022-03-01 01:00:00
	2019-09-24 09:00:00	2019-09-24 11:00:00			
	2019-09-25 18:00:00	2019-09-28 00:00:00			
	2019-09-28 06:00:00	2019-09-28 07:00:00			
	2019-09-28 11:00:00	2019-09-29 04:00:00			
	2019-09-30 20:00:00	2019-10-01 19:00:00			
	2019-10-03 16:00:00	2019-10-04 17:00:00			
	2019-10-05 21:00:00	2019-10-09 02:00:00			
	2019-10-13 08:00:00	2019-10-15 01:00:00			
	2019-11-10 10:00:00	2019-11-11 00:00:00			
	2019-11-13 00:00:00	2019-11-13 22:00:00			
	2019-11-15 12:00:00	2019-11-15 14:00:00			
T_{water}	2019-11-15 23:00:00	2019-11-16 21:00:00	X_1^{pump}		
	2019-11-17 20:00:00	2019-11-20 07:00:00			
	2019-12-02 16:00:00	2019-12-02 22:00:00			
	2019-12-05 07:00:00	2019-12-06 16:00:00			
	2019-12-10 09:00:00	2019-12-13 03:00:00			
	2019-12-15 07:00:00	2019-12-31 06:00:00			
	2020-01-04 12:00:00	2020-01-05 13:00:00			
	2020-01-05 20:00:00	2020-01-08 04:00:00			
	2020-01-14 08:00:00	2020-01-16 18:00:00			
	2020-01-18 20:00:00	2020-01-26 05:00:00			
	2020-01-28 21:00:00	2020-01-30 14:00:00			
	2020-08-07 15:00:00	2020-08-07 16:00:00			
	2020-08-07 23:00:00	2020-08-08 00:00:00			
	2021-01-27 13:00:00	2021-01-28 12:00:00			
	2021-02-07 11:00:00	2021-02-08 10:00:00			
	2021-02-09 15:00:00	2021-02-10 14:00:00			
	2021-06-06 10:00:00	2021-06-07 10:00:00			
	2021-08-30 17:00:00	2021-08-31 00:00:00			
	2021-08-31 07:00:00	2021-08-31 13:00:00			
	2021-08-31 22:00:00	2021-08-31 23:00:00			
	2021-09-01 03:00:00	2021-09-01 05:00:00			
	2021-09-04 09:00:00	2021-09-04 11:00:00			
	2021-09-04 13:00:00	2021-09-05 11:00:00			
	2021-09-05 19:00:00	2021-09-05 21:00:00			
	2021-09-08 01:00:00	2021-09-08 01:00:00			
	2021-12-05 13:00:00	2021-12-06 12:00:00			

8 ANALYSIS OF TEXTUAL DATA

The analysis of the textual data presented in Section 4.2 follows two paths: knowledge extraction using TLP methods and text summarization. These paths are discussed in detail in Sections 8.1 and 8.2, respectively.

8.1 KNOWLEDGE EXTRACTION FROM TEXTUAL DATA

Issue reports (IRs) and work orders (WOs) are valuable data sources for tracking asset health histories, identifying health trends, and performing root-cause analyses. These data sources, typically obtained in text form, are usually available in digital repositories. Natural language processing methods (Lane, 2019) have been developed over the past two decades to enable ML models to analyze textual data and classify textual elements based on their nature (e.g., safety related vs. non-safety related). 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 identifying:

- 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)
- Measured quantities (i.e., a numeric value followed by unit of measure)
- 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 TLP methods was designed to identify all elements listed above, using a mixture of rule-based and ML algorithms. These methods (Wang, 2024) 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 6) are mentioned in ER textual data elements.

Figure 16 provides an overview of the NLP methods that together constitute the knowledge extraction workflow. These methods are grouped into the following three main categories:

- Text preprocessing: The provided raw text is cleaned and processed to identify specific nuclear entities and acronyms (e.g., HPI in reference to a high-pressure injection system) and to identify and correct typos (i.e., through a spell check method) and abbreviations (e.g., “pmp” meaning “pump”).
- Syntactic analysis: The goal of this analysis is to identify the relationship between words contained within a sentence, focusing on understanding the logical meaning of sentences or parts of sentences (e.g., subjects, predicates, and complements).
- Semantic analysis: We rely on the results of this analysis to identify the nature of the event(s) described in the text, along with their possible relationships (temporal or causal).

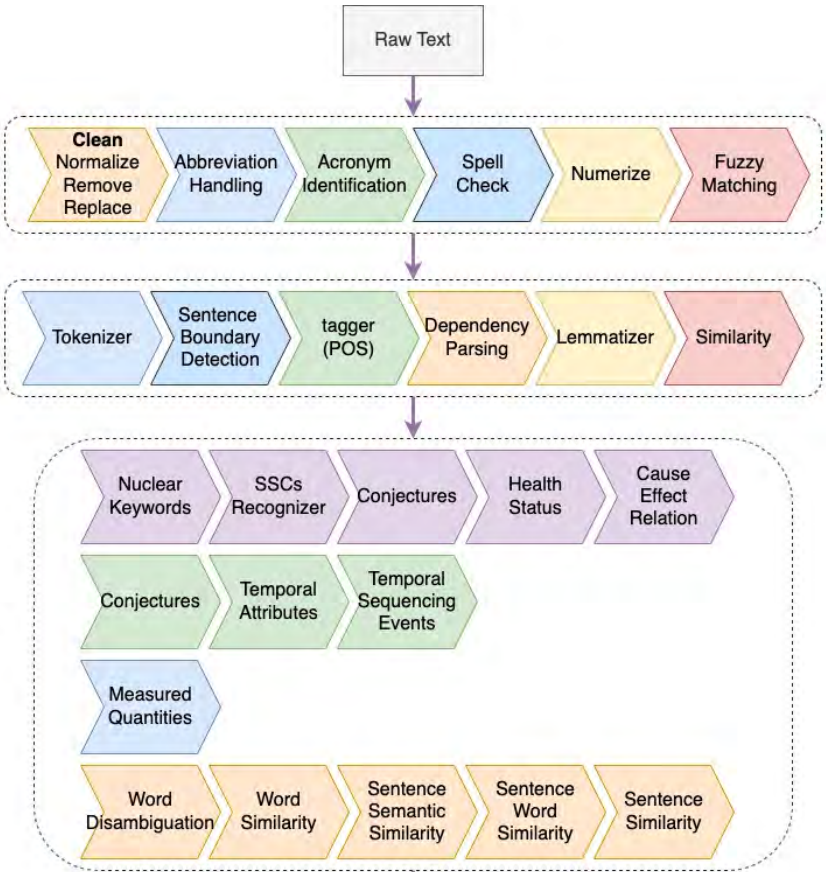

Figure 16. Graphical representation of the NLP element that comprise the knowledge extraction workflow (Wang, 2024).

In this respect, the sequence of steps applied to each textual data element is:

1. *Tokenization and Lemmatization*. The first step in text processing is tokenization, for which we employed the SpaCy tokenizer (i.e., to segment the text into a list of words, punctuation marks, etc.) by applying rules specific to raw text. First, the raw text is split on whitespace characters. The tokenizer then processes the text from left to right. On each substring, it performs the following two checks:
    - 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."
    - Can a prefix, suffix, or infix be split off (e.g., based on punctuation such as commas, periods, hyphens, or quotation marks)?
2. *Sentence Segmentation*. The next important step is to determine the sentence boundaries—that is, segment the text into a list of sentences. This is a key underlying task in the NLP process. For the present work, we employed PySBD, a rule-based sentence boundary disambiguation Python package, to detect the sentence boundaries. With it, we developed a custom pipeline that can be employed in tandem with SpaCy to divide up text into a list of sentences.
3. *Part of Speech* (POS). We used the SpaCy tagger to parse each sentence and tag every token found within. Both TAG and POS attributes were generated for each token after the SpaCy tagger process. POS, the simple universal part-of-speech tag, does not include information on any morphological

features, only the word type (https://universaldependencies.org/u/pos/). The morphology is the process by which the 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 via 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 employed these POS and TAG tags to determine the nature of the asset health status estimate (conjecture or qualitative observations).

4. *Dependency Parsing*. POS provides information on word types and morphological features but not on the dependencies between words. Thus, we employed the SpaCy parser to label dependencies uncovered during parsing. Among such dependencies are nominal subject (nsubj), direct object (dobj), and indirect object (iobj). The parser utilizes a variant of the nonmonotonic arc-eager transition system described in (Honnibal and Johnson, 2014). The parser uses the terms "head" and "child" to describe words connected by a single arc in the dependency tree. The dependency labels, listed in https://v2.spacy.io/api/annotation, are used in determining the arc label, which describes the type of syntactic relation connecting the child and the head.

5. *Spellchecking and the Handling of Acronyms and Abbreviations*. NPP IRs and WOs are usually written out in short sentences that often contain abbreviations, making it harder to accurately extract knowledge. Thus, we developed an NLP pipeline for identifying abbreviations and replacing them with the complete words they represent. The starting point is a library of word abbreviations collected from documents available online. This library is basically a dictionary that contains the corresponding set of words for each identified abbreviation, the inherent challenge being that a single abbreviation can be associated with multiple different words. Similarly, a word might also have different ways of being abbreviated. To handle the abbreviations found in each sentence, all misspelled words are first identified. Each misspelled word is then searched for in the developed library. If an abbreviation in the library matches the misspelled word, it is replaced by the corresponding complete word. If no abbreviation is found, we proceed by searching for the closest match. If multiple words match the obtained abbreviation, the one that best fits the context of the sentence is chosen. Acronyms are often present in ER textual data, and typically refer to specific NPP assets or systems. Also, in this case, a library of acronyms was developed based on publicly available NRC, EPRI, and NEI documents. Any remaining misspelled words are parsed through the developed library in search of an exact match. After the abbreviations and acronyms have been handled, the remaining misspelled words are parsed through our spellchecking methods for one final attempt at correction.

6. *Identification Temporal Attributes*. Temporal attributes indicate the time instances at which specific events occurred. Time of occurrence is important from a causal point of view, as the emergence of an effect is always preceded by its cause. Hence, temporal information can be valuable in identifying the causal links between recorded events. Temporal attributes are identified by looking at specific prepositions and relations.

7. *Identification of Measured Quantities*. We now wish to identify the presence of a precise observation (i.e., a measured point value or delta estimate) of a measured variable. Such an observation requires a numeric value followed by its unit; however, the unit is often missing.

8. *Identifying Nuclear Keywords*. In the medical field, NLP knowledge extraction methods require the capability to identify specific entities. This is similarly the case in the nuclear field, in which such entities are, among other things, systems, assets, and components found in any NPP. A library for the nuclear field was developed in past years by using available NRC and EPRI textual data.

Entities contained in this library (about 5,000 and growing) have been grouped into different categories (e.g., mechanical, hydraulic, electric, and electronic components and assets, degradation phenomena, and architectural entities).

9. *Identification of Conjectures*. A relevant element of knowledge extraction is the ability to distinguish between information pertaining to future predictions (e.g., an event that can occur in the future) or hypotheses about past events (e.g., a failure that potentially occurred). Future predictions are characterized by present- and future-tense verbs, whereas hypotheses about past events are typically characterized by past-tense verbs. Also, for these kinds of reports, we identified specific keywords and relations that may indicate we are dealing with a conjecture observation.
10. *Identification of Cause-Effect Relations*. A common pattern in ER textual data is the reporting of multiple events that all share a causal relationship. In its simplest form, such a paragraph refers to an event (i.e., cause) that triggered a second event (i.e., effect). However, this type of paragraph can be structured in different ways: an event that has been identified as not being the cause of another event, multiple causes that trigger a single effect, or a single cause that triggers multiple effects. In the present work, our methods did not employ ML algorithms, such as for supporting classification methods (Mohri and Rostamizadeh 2012), but were instead rule based (Doan et al. 2019), as we aimed to extract actual quantitative information from textual data, rather than "classifying" the nature of the raw text. These rules were based on identifying the following: keywords that indicate the possibility of a causal relation between a subject and an object or NLP structures (or constructs) that indicate a casual transition between clauses in a sentence (e.g., the word consequently).
11. *Identification of Temporal Sequencing of Events*. Temporal relations can be either quantitative (e.g., an event that occurred two hours after another event) or qualitative (e.g., an event that occurred prior to another event). Note that a temporal relation does not necessarily imply a causal relation. In this paper, we build on the work in (Moerchen, 2010), which lists the major temporal relations between events: order (sequential occurrence of events), concurrency (nearly simultaneous occurrence of events from beginning to end), and coincidence (temporal intersection of events).
12. *Identification of Health Status*. Often, IRs reflect qualitative information on abnormal observed events (e.g., failures or precursors to a degradation phenomenon). From a reliability standpoint, identifying the nature of the reported event plays a major role, with the goal being to track the health performance of a single SSC or multiple SSCs operating in similar operating conditions. Based on the large number of IRs and WOs gathered from operating NPPs, we collected and extracted a list of keywords (nouns, verbs, adverbs, and adjectives) for indicating the health status along with the underlying grammatical structures and converted them into relations. These keywords have been partitioned into three main classes (negative, positive, and neutral) based on sentiment analysis and then expanded using the WordNet synonym search capabilities. Thus, identification of the health status of the textual clause can be assessed by searching in the text for the developed lists of relations and keywords.

8.2 TEXT SUMMARIZATION ANALYSIS

A second path that we follow to extract knowledge from text is here referred as text summarization. The goal is to extract the nature of a clause or a sentence. In a system reliability context, the nature of clause (or a sentence) can be of a different nature; common examples are:

- The report of a surveillance or a maintenance activity
- The observation of a degradation phenomena

- The diagnosis performed from an observed anomalous behavior.

This task requires two main elements; the first one is a model that describes system reliability operations and physical elements that might affect performance of assets and components that are part of that system. Also in this case, we have developed such model through a MBSE diagram developed using the OPM language, which is shown in Figure 17. Such a model can be interpreted as follows[4]: the key element of this model is an *asset* designed to support a *specific* function. Through its lifetime, such an asset can be either in a *degraded* or *OK* state. The transition from the OK to degraded state is caused by *a degradation mechanism*, which is driven by either *external agents* or chemical and physical *reactions*. *Inspection* operations are typically performed to assess asset state (degraded or OK) through *surveillance tools* where quantitative *measurement* can be reported. If the asset is found in a degraded state, *diagnosis* operation can be performed to trigger a *maintenance* operation designed to restore asset operation (from a degraded to OK state) by employing specific *maintenance tools*.

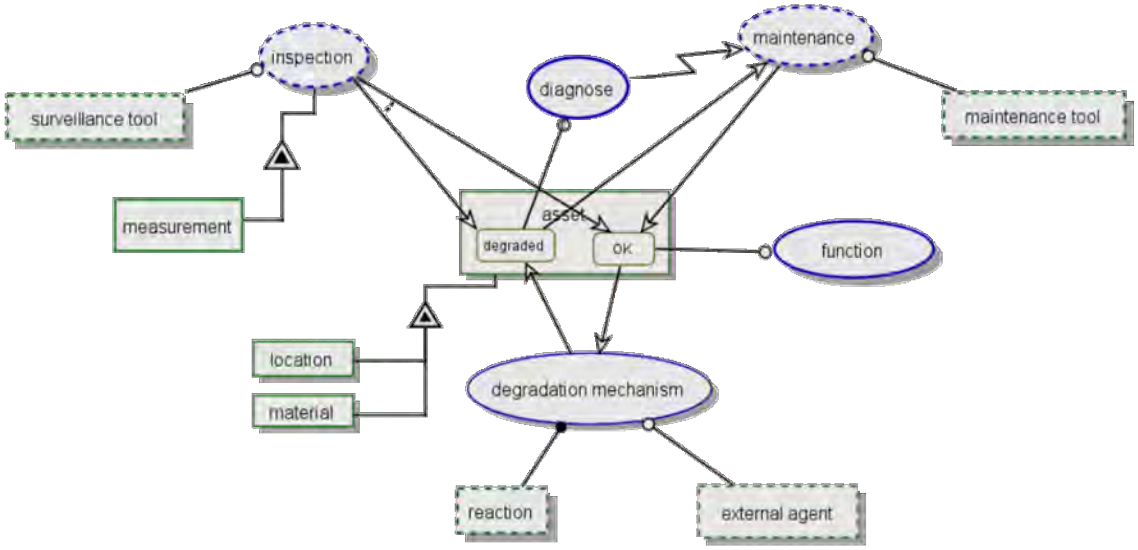

Figure 17. Graphical representation of the developed ER schema for text summarization.

The second element required to perform text summarization is a vocabulary of terms (i.e., nouns or verbs) that are semantically associated with each entity of the model described above. Such vocabulary has been developed using openly available documents and tested against plant data to check vocabulary completeness. Currently, this vocabulary contains about 5,500 terms (either verbs or nouns), which are arranged into eight main classes and subclasses and is continuously being updated. Table 4 lists the various classes and subclasses available so far, along with examples of entities corresponding to each group. Each subclass is naturally associated with specific entity of the MBSE model shown in Figure 17.

Our method for text summarization employs `SpaCy` name entity recognition functions to identify the terms of our database (summarized in Table 4) within a textual data element. Identified entities are flagged with their associated class and subclass and saved as part of the metadata associated with the textual data. Then, using the correspondence between class and subclass and the ER schema entity (of the diagram shown in Figure 17) shown in Table 4, a list of identified MBSE entities is ordered based on the connections shown in Figure 17. Figure 18 provide a few examples of text summarization.

[4] Note that the elements in the MBSE graph shown in Figure 17 are highlighted in italics in the text.

Table 4. List of classes and subclasses for the developed dictionary of nuclear related entities and the associated entity on the schema shown in Figure 17.

Class	Subclass	Examples	ER Schema Entity
Mechanical components	Fasteners Rotary elements Structural Purpose specific	Anchor bolt, cap screw, latch Cam, shaft, gear, pulley Beam, column, sleeve, socket Filter, manifold, blade	Asset
Nonmechanical components	Electrical/electronic Hydraulic/Pneumatic	Amplifier, relay, capacitor Coupler, filter, pipe	Asset
Assets	Mechanical Electrical Hydraulic/Pneumatic Electronic I&C Nuclear fuel	Engine, vessel AC bus, alternator, generator Pump, valve, condenser, fan Computer, tablet, controller FPGA, transmitter, sensor Fuel rod, control blade	Asset
NPP elements	Systems Architectural	Feedwater, switchyard Containment, pump house	Asset Location
Tools and treatments	Maintenance tools Maintenance operations Surveillance tools Surveillance operations Diagnosis	Jigsaw, solder gun, tape, crane Bolting, riveting, grinding Inspection, leak test, infrared test Sample, verify, inspect Require, need, demand	Maintenance tool Maintenance Surveillance tool Surveillance Diagnose
Operands	Electrical Hydraulic/Pneumatic	AC current, electromagnetic Compressed air, steam, gasoline	
Materials	Chemical compounds Chemical elements Materials	Ammonia, ethanol, methane Plywood, concrete, polyethylene Fiberglass, lumber, cement	Material
Reactions	Chemical reaction Degradation mechanism Failure type	Combustion, oxidation Corrosion, dissolution, fatigue Leak, rupture, brittle fracture	Reaction Degradation Mechanism Anomalous state

Original Text	Text Summary
Pump was inspected	[inspection]
CWS P1 pump shut-down due to vibrations	[degraded mechanism, asset(anomalous)]
FG1T isolated	[maintenance]

Figure 18. Example of text summarization using the ER schema.

9 TEMPORAL CORRELATION ANALYSIS

Sections 7 and 8 have presented methods of analyzing numeric and textual ER data elements, and we explained how MBSE diagrams can be employed to identify possible causal relationships between ER data elements. 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 such dependence is to observe their temporal correlation.

9.1 EVENT-EVENT TEMPORAL CORRELATION ANALYSIS

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 a 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:

$$I_t(E_1, E_2) = 1 - e^{\frac{-(t_2 - t_1)}{\tau}} \tag{4}$$

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, $I_t(E_1, E_2)$ approaches the value of 1. If the events E_1 and E_2 are far from each other, $I_t(E_1, E_2)$ approaches the value of 0. The parameter τ specifies the scale of the temporal closeness of the two events.

9.2 EVENT-TIME SERIES TEMPORAL CORRELATION ANALYSIS

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 two relevant items: a modification to the testing process structure and a different two-sample testing algorithm.

In its original formulation (Luo, 2014), the temporal correlation is 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 in (Luo, 2014) is based on testing 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 19]) an event defined over a temporal instant has occurred.

The right plot in Figure 19 is adapted and modified from Luo (2014), and it provides 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 right plot in Figure 19 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 $\rightarrow$ indicates a temporal relationship between an event E and S but does not necessarily imply a causal relationship between the two.

Here, we employ the Maximum Mean Discrepancy (MMD) algorithm (Gretton, 2006) to perform such statistical testing between the portion of the time series before and after the event temporal occurrence (i.e., l_E^{front} and l_E^{rear}). In its original definition, the MMD algorithm has been developed to test if two observed stochastic variables are characterized by the same probabilistic distribution function: 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):

$$H_0: F = G$$
$$H_1: F \neq G \tag{5}$$

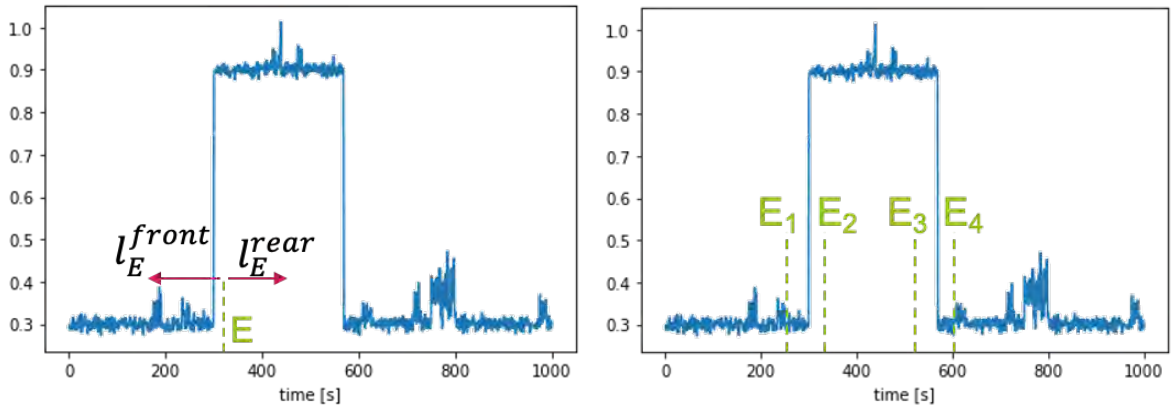

Figure 19. (left) Elements designed to test the temporal correlation of a time series with an instantaneous event E, and (right) use cases considered for evaluating the temporal correlation of a time series with a set of instantaneous (left plot) and interval (right plot) events.

This is achieved via the following MMD testing with a particular threshold α; if the threshold is exceeded, 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 that must set in advance, and it is used to determine the threshold to which we compare the test statistic.

Here, we are directly applying the MMD algorithm to test time series. In this respect, let us consider an event E defined over a time instant t_E and a time series S (either univariate or multivariate). With the term $T\big((E, t_E), S\big)$, we indicate the testing algorithm that determines the temporal correlation between an event E that occurred at time t_E and a time series S; the possible set of outcomes generated by $T\big((E, t_E), S\big)$ are described in Table 5 provides indication of the list of outcomes. Note from Table 5 that we are only not considering the clear cases where a temporal correlation does (i.e., $E \rightarrow S$ and $S \rightarrow E$) or does not exist (i.e., $S\,!\,E$) but also these two additional situations:

- $S; E$: E occurs in an abnormal (statically speaking) transient of S
- $S?\,E$: The temporal correlation testing is negative; however, the obtained p value indicates a statistically significant difference between the time windows before and after the occurrence of event E in the times series S

As a notation, we indicate with the term $T^{MMD}(S_1, S_2)$ as the MMD-based algorithm designed to test whether time series S_1 and S_2 have the same statistical distribution that returns the outcome of the test (Boolean logic value). The identification of the temporal relation between E and S is presented in detail in Algorithm 1. This algorithm operates by comparing the statistical distribution l_E^{front} and l_E^{rear} against S; this is performed by randomly sampling portions of S that have same duration of l_E^{front} and l_E^{rear}. Such a set of portions of S is indicated as Θ.

Given a time series S, a randomly sampled subseries Θ generated from S is here denoted as $\Theta = (S_1, \dots, S_n, \dots, S_N)$, where each subseries S_n has the same length of k. Assume S only contains two states:

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 subseries 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 subseries S_n are similar to S_E based on previous MMD testing over the total number of subseries 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 20.

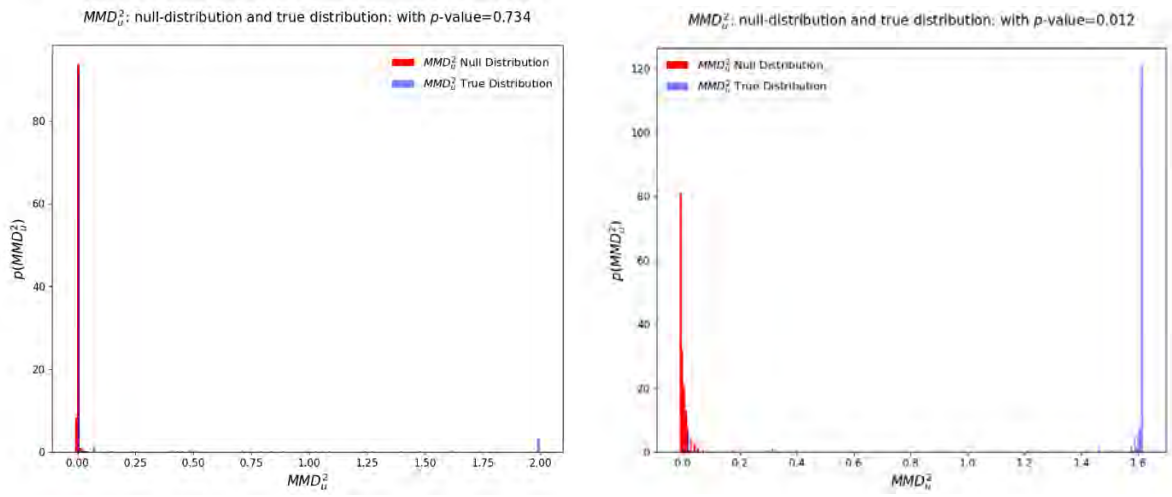

Figure 20. MMD testing of event E_1 shown in the right plot of Figure 19. The left plot shows l_E^{front} vs. Θ, and the null hypothesis defined in the previous paragraph is accepted since the p value is greater than the significant level value. The right plot shows l_E^{rear} vs. Θ; the null hypothesis previously defined is rejected since the p value is smaller than the significant level value.

Using the example shown in Figure 19 (right), we have tested Algorithm 1 with the time provided time series against eight events.

Table 5. Notation employed to indicate temporal correlation between time series S and an event E.

Temporal Notation	Meaning
$E \rightarrow S$	The occurrence of event E is followed by a change in the time series S
$S \rightarrow E$	A change in the time series S is followed by the occurrence of event E
$S; E$	The occurrence of event E is during the change in the time series S
$S? E$	There is no correlation identified between time series S and the occurrence of event E. However, within the testing time window, there is a significant difference between the time windows before and after the occurrence of event E in the times series S
$S! E$	There is no correlation identified between time series S and the occurrence of event E

Algorithm 1: $T\big((E, t_E), S\big)$ - Temporal testing between event and time series

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

Output: Temporal correlation D

1. Initialize Θ
2. Select portion of time series S before and after t_E: l_E^{front}, l_E^{rear}
3. Determine:
   - $D_f = T^{MMD}\big(\Theta, l_E^{front}\big)$
   - $D_r = T^{MMD}(\Theta, l_E^{rear})$
   - $D_{fr} = T^{MMD}\big(\Theta, l_E^{front} \cup l_E^{rear}\big)$
   - $d_{fr} = T^{MMD}\big(l_E^{rear}, l_E^{front}\big)$
4. If $D_r = True$ & $D_f = False$: #E1
   - Return $D = E \rightarrow S$
5. Elif $D_r = False$ & $D_f = True$: #E4
   - Return $D = S \rightarrow E$
6. Elif $D_r = True$ & $D_f = True$: #E2 and E3
   - Return $D = S;E$
7. If $D_{fr} = True$:
   - Return $D = S;E$
8. Elif $d_{fr} = True$:
   - Return $D = S?E$
9. Else $d_{fr} = False$:
   - Return $D = S!E$

Table 6. Examples of temporal correlation analysis based on MMD testing for the events E_1, E_2, E_3, and E_4 shown in Figure 19. E_5 is similar to E_4 with 7 seconds temporal shift. E_A, E_B, E_C are specified at location 900, 500, 200, seconds respectively.

	l_E^{front} $vs.\Theta$		l_E^{rear} $vs.\Theta$		l_E^{front} $vs.\, l_E^{rear}$		$l_E^{front} \cup l_E^{rear}$ $vs.\Theta$		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.

An example of a correlation analysis of events and time series is shown in Figure 21 where a monitored variable is correlated to a set of events processed in Section 8. The identified events that have a temporal correlation with the time series are indicated in red, black, and yellow.

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 the 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. This situation is currently the subject of study.

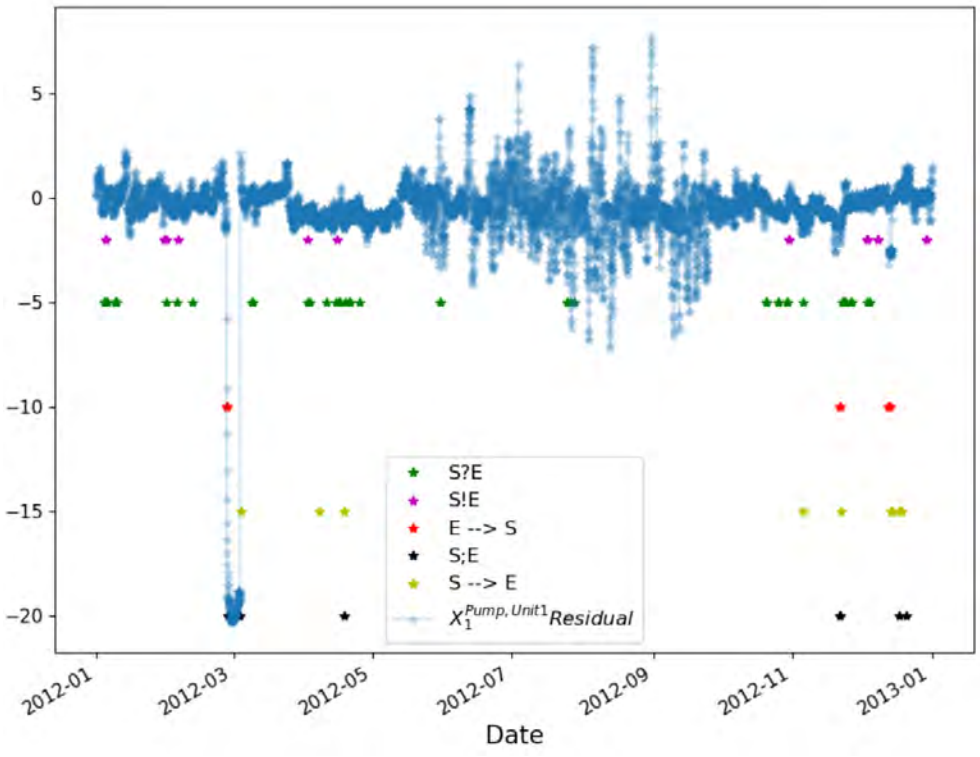

Figure 21. Example of a correlation analysis of events and time series using the notation shown in Algorithm 1.

10 ER KNOWLEDGE GRAPH

This section provides details about the topology (see Section 10.1) and the process of construction (see Section 10.2) of the reliability knowledge graph.

10.1 ER KNOWLEDGE GRAPH DEFINITION

As indicated in Section 1, we aim to capture both system and plant architecture and available ER data in a single relational database. System architecture is modeled through MBSE diagrams (see Section 6), which are then translated into a single graph where each node in this graph indicates a specific asset,

component, or function.

In this paper, we refer to knowledge graph taxonomy as the hierarchical schema that defines the key elements (i.e., nodes, and edges) of the ER knowledge graph itself. In this respect, Figure 22 provides the basic taxonomy of the ER knowledge graph and it explicitly shows the three types of nodes here considered: MBSE entities, numeric ER data, or textual ER data. The edges between MBSE and data nodes are indicated as "*data association*." The methods presented in Section 9 provide information about the temporal correlations between events. From the knowledge graph point of view, if a temporal correlation is identified, then a "*data correlation*" edge between the two nodes is added. Table 7 and Table 8 provide more quantitative details about the sets of nodes and edges, respectively, that define the considered ER knowledge graph taxonomy.

Numeric ER data is processed using the anomaly detection methods shown in Section 6. From a knowledge graph point of view, two nodes are created for each monitored variable: the full temporal profile of the considered variable and the list of anomalies identified from such a time series using the methods described in Section 7. Provided system design knowledge, these two nodes that contain numeric data are then linked to the node pointing to the MBSE entity being monitored.

Similar reasoning applies to each textual data element. From the TLP knowledge extraction methods, a graph is constructed as described in Section 8. If the textual element is indicating a specific asset or component that is represented in the system MBSE diagram, the textual element is associated with the node pointing to the MBSE entity mentioned in the text.

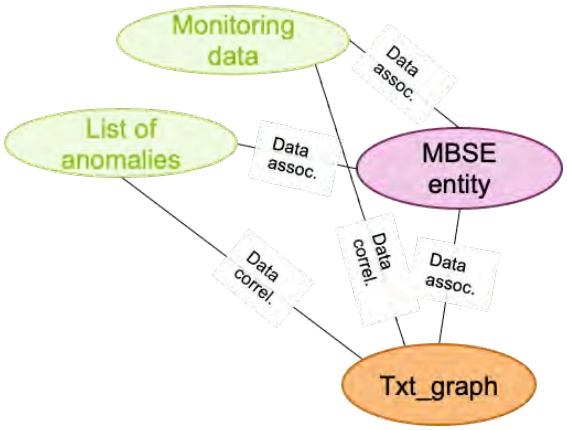

Figure 22. Topology of the ER knowledge graph; numeric (shown in green) and textual data elements (shown in orange) are logically connected to nodes pointing to MBSE entities (shown in purple).

10.2 ER KNOWLEDGE GRAPH CONSTRUCTION

Provided the set of processed ER data elements—either numeric (see Section 7) or textual (see Section 8)—the goal becomes to organize each element into a graph structure that captures the cause-effect relations (logical and temporal) identified in Section 9. Our approach began with the graph structure derived from the MBSE models of the system and assets under consideration (see Section 6), then progressed through the following steps:

1. Associate an ER textual data element with one (or more) MBSE entity.
2. Identify ER numeric data elements that have a logical path to the ER textual data element identified in Step 1.
3. Determine whether there is a temporal relation between the ER textual data element identified in Step 1 and the ER numeric data elements identified in Step 2 (see Section 9).

4. If both the temporal and logical relation have been identified in Step 3:
    a. Link the portion of the ER numeric data element to its corresponding MBSE element.
    b. Link the data element identified in Step 4a to the ER textual data element identified in Step 1.
5. Repeat Steps 1–4 for each ER textual data element.

Table 7. Taxonomy of the nodes of the knowledge graph.

Node type	Data format	Attributes
MBSE	Textual	Entity ID (optional)
Numeric	Time series associated with a set of monitoring data variables (Pandas data frame)	-
	List of identified anomalies (see data format shown in Section 7)	-
Textual	Graph constructed from textual element	Reported date Text summarization (optional) Health status assessment (optional) Conjecture flag (optional)
	Nuclear entity	Entity subclass ID

Table 8. Taxonomy of the edges of the knowledge graph.

Edge type	Properties	Linked nodes
MBSE edges	Directional or nondirectional	Set of edges that connect MBSE entities (form and functional elements)
Data association	Nondirectional	MBSE to data (either numeric or textual) nodes
Data correlation	Directional or nondirectional	Data node to data node

The resulting relational database will take the form of a graph structure reflecting the links between the data elements associated with a particular MBSE entity. Again, the actual skeleton of the graph structure is directly derived from the MBSE diagram of the system and assets under consideration. In this respect, Figure 23 shows the CWS graph structure directly generated from the provided MBSE diagram. Note that the graph nodes can reflect different data types (form or function), and the same applies to edges.

For the present article, we focused on the textual portion of the available ER dataset for the considered CWS over a 10-year lifespan. The knowledge extraction methods presented in the past sections were employed to analyze all shift logs, WOs, and IRs, enabling us to identify the nature of textual elements and the MBSE elements associated with them. As an example, Figure 23 shows how the knowledge graph is populated by first obtaining the graph from the system MBSE model; then, anomalies identified using the methods indicated in Section 7 and the events processed using the TLP methods shown in Section 8 are associated with one or more MBSE entity.

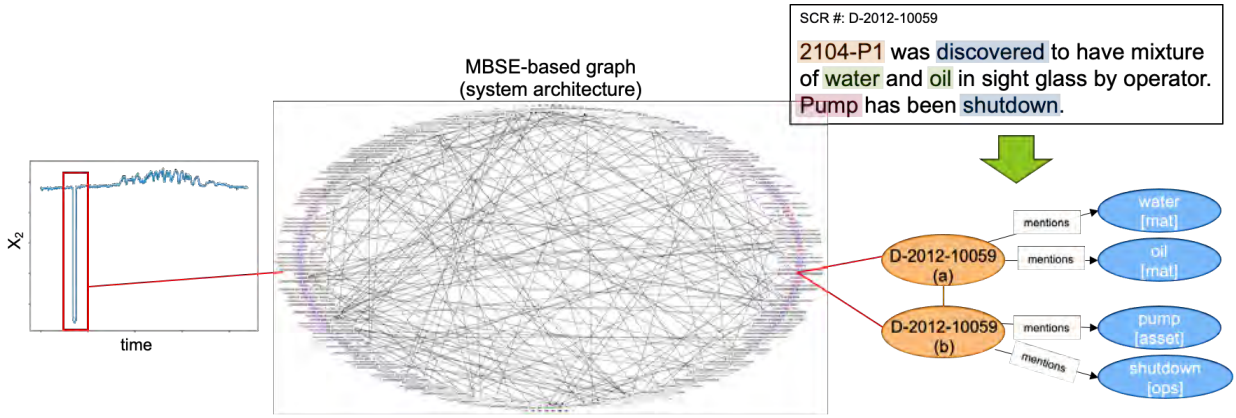

Figure 23. Graphical representation of the obtained knowledge graph data structure; graph obtained from the system MBSE models (shown in the center) captures system architecture, while numeric (i.e., anomalies) and textual data elements are associated to specific nodes of the MBSE graph.

11 CONCLUSIONS

This paper has presented an approach designed to holistically integrate ER data (both numeric and textual) to track and record historic system and asset reliability performance in the form of a knowledge graph. The first distinction of our approach to construct a knowledge graph is that it embraces observed and recorded data and also system architecture.

We in fact rely on MBSE models to capture system architecture through diagrams (form and functional representation based on diagrams), which are then translated into a single digital graph structure. This graph becomes the actual skeleton of the knowledge graph. The rationale behind this modeling choice is that the ability to identify cause-effect relations between events requires architectural models of the considered system to "put data into context."

Once the skeleton of the knowledge graph is constructed, it is then populated by associating processed ER data to specific nodes (i.e., MBSE entities) of the graph. Numeric ER data is processed through statistical and ML-based anomaly detection methods with the goal of identifying anomalous behaviors. On the other hand, textual ER data is parsed by TLP methods with the goal of extracting knowledge from text and generating a data graph out of textual elements.

The fourth step in the construction of the knowledge graph is identifying the temporal correlation between events (from textual data) and anomalies (from time series). This discovery process is performed through statistical testing methods that capture causal relations (temporal and logical).

The obtained knowledge graph merges system architecture and ER data into a single digital structure. This data structure can be then employed to perform several tasks, including identifying patterns of anomalies, diagnoses of causes from a set of anomalies across systems, assessments of historic asset health performances, and updates of plant probabilistic risk analysis models.

Note that the proposed approach is not bound to a specific anomaly detection or knowledge extraction method; we in fact provide well defined application programing interfaces (APIs) such that currently employed methods in plant monitoring and diagnostic (M&D) centers can be easily interfaced. The methods described here can be considered as state-of-the-art since they rely on recent data analytics advancements designed to overcome some limitations of current state-of -practice methods.

Note also that the already built knowledge graph (e.g., for the CWS system) can be easily expanded by adding or merging the knowledge graphs developed for supporting systems (e.g., the 4160V AC system for

the CWS system). The developed knowledge graph construction process is in fact modular in the sense that a knowledge graph can be constructed for each system, but then these graphs can be merged once the cross-system dependencies are capture in the system MBSE models.

Modularity can be also achieved from a data point of view; additional data sources can be found in each utility such as: outage data (i.e., maintenance and surveillance operations performed periodically during plant outages), asset usage data (e.g., historic number of hours an asset has been running), regulatory related data (e.g., the basic event ID of an asset as part of the plant risk model, or the set of risk-informed plans associated with that asset), and economical data (e.g., procurement and maintenance costs). These data sources can be added to knowledge graph provided a well-defined label to the node in the graph that contains such data. This feature allows multiple stakeholders (e.g., system engineers, plant risk analysts, financial teams) to provide their own perspective of an asset (i.e., operational, regulatory, economical) into a unique and coherent structure designed to overcome current data limitations of nuclear utilities: missing, redundant, or contradictory information.

REFERENCES

Booch, G., J. Rumbaugh, and I. Jacobson. 2017. *Unified Modeling Language User Guide (2nd Edition)*. Addison-Wesley.

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

Yeh, C.-C., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. Furtado Silva, A. Mueen, E. Keogh. 2016. *All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets*. IEEE ICDM 2016.

Coble, J., P. Ramuhalli, L. Bond, J.W. Hines, B. Upadhyaya. 2015. A review of prognostics and health management applications in nuclear power plants. Int. J. Progn. Heal. Manag. 6, 2271. https://doi.org/10.36001/ijphm.2015.v6i3.2271.

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.

LML steering committee. "Lifecycle Modeling Language (LML) Specification." https://22132398.fs1.hubspotusercontent-na1.net/hubfs/22132398/LML%20specification%201.4.pdf.

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.

Law, S. M. 2019. *STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining*. Journal of Open Source Software, 4, 39, 1504.

Wang, C., D. Mandelli, and J. Cogliati. 2024. “Technical Language Processing of Nuclear Power Plants Equipment Reliability Data.” *Energies,* vol. 17, no. 7.

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.

Zhao, X., 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.

Moerchen, F. 2010. Temporal pattern mining in symbolic time point and time interval data. In Proceedings of the KDD’10: The 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Nashville, TN, USA, 30 March–2 April 2010; pp. 2–1.

Wang, C., D. Mandelli, and J. Cogliati. 2024. “Technical Language Processing of Nuclear Power Plants Equipment Reliability Data”. Energies, vol. 17, no. 7: 1785. https://doi.org/10.3390/en17071785.

Çelik, M., F. Dadaşer-Çelik, and A. Dokuz. 2011. Anomaly detection in temperature data using DBSCAN algorithm. International Symposium on Innovations in Intelligent Systems and Applications. Istanbul, Turkey.

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databased with noise. kdd, 96(34), 226-231.

Godbole, C., G. Delipei, X. Wu, M. Avramova, and U. Rohatgi. 2022. Machine Learning-based Prediction of Departure from Nucleate Boiling Power for the PSBT Benchmark. Advances in Thermal Hydraulics (ATH 2022).

Huber, P. J. (1981). Robust Statistics. Net York: John Wiley and Sons, Inc.

Work, B. C.-O. (n.d.). CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17045963.

Zhao, X., K. Shivran, R. Salko, and F. Guo. 2020. On the Prediction of Critical Heat Flux using a Physics-Informed Machine Learning-Aided Framework. Applied Thermal Engineering, 164 (114540).

Appendix B

Technical Language Processing of Nuclear Power Plants Equipment Reliability Data

energies

MDPI

Article

Technical Language Processing of Nuclear Power Plants Equipment Reliability Data

Congjian Wang, Diego Mandelli * and Joshua Cogliati

Idaho National Laboratory, 1955 Fremont Ave., Idaho Falls, ID 83415, USA; congjian.wang@inl.gov (C.W.); joshua.cogliati@inl.gov (J.C.)
* Correspondence: diego.mandelli@inl.gov

Abstract: Operating nuclear power plants (NPPs) generate and collect large amounts of equipment reliability (ER) element data that contain information about the status of components, assets, and systems. Some of this information is in textual form where the occurrence of abnormal events or maintenance activities are described. Analyses of NPP textual data via natural language processing (NLP) methods have expanded in the last decade, and only recently the true potential of such analyses has emerged. So far, applications of NLP methods have been mostly limited to classification and prediction in order to identify the nature of the given textual element (e.g., safety or non-safety relevant). In this paper, we target a more complex problem: the automatic generation of knowledge based on a textual element in order to assist system engineers in assessing an asset's historical health performance. The goal is to assist system engineers in the identification of anomalous behaviors, cause–effect relations between events, and their potential consequences, and to support decision-making such as the planning and scheduling of maintenance activities. "Knowledge extraction" is a very broad concept whose definition may vary depending on the application context. In our particular context, it refers to the process of examining an ER textual element to identify the systems or assets it mentions and the type of event it describes (e.g., component failure or maintenance activity). In addition, we wish 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, then machine learning (ML) and rule-based algorithms are employed to identify physical entities (e.g., systems, assets, and components) and specific phenomena (e.g., failure or degradation). A few applications relevant from an NPP ER point of view are presented as well.

Keywords: natural language processing; knowledge extraction; machine learning

Citation: Wang, C.; Mandelli, D.; Cogliati, J. Technical Language Processing of Nuclear Power Plants Equipment Reliability Data. *Energies* **2024**, *17*, 1785. https://doi.org/10.3390/en17071785

Academic Editor: Dan Gabriel Cacuci

Received: 1 February 2024
Revised: 19 March 2024
Accepted: 1 April 2024
Published: 8 April 2024

1. Introduction

To reduce operation and maintenance costs [1,2], existing nuclear power plants (NPPs) are moving from corrective and periodic maintenance to predictive maintenance strategies [3]. This transition is designed so that maintenance occurs only when a 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 [4–7] are essential for supporting predictive strategies, as they can provide precise information about the health of a system, structure, or component (SSC), track its degradation trends, and estimate its expected time of failure. With such information, maintenance operations can be performed on a component right before its expected failure time [8].

This dynamic context of operations and maintenance activities (i.e., predictive) requires new methods of processing and analyzing equipment reliability (ER) data [7,8]. One relevant issue is that ER data can be contained in heterogenous data formats: textual,

numeric, image, etc. An analysis of numeric ER data has been addressed in many previous works [5–9] and applied to many operational directions including anomaly detection, diagnosis, and prognosis. Here we are targeting the analysis of textual ER data. The information contained in NPP textual ER data can either describe the occurrence of abnormal events (e.g., system, structure and components [SSC] failure or observed degradation)—with such documents being referred to here as issue reports (IRs)—or the conduct of maintenance or surveillance activities (referred to here as work orders [WOs]). Only recently has the analysis of textual data been investigated via machine learning (ML) methods [10–13] designed to assess the nature of the data (e.g., safety or non-safety related) by employing supervised or semi-supervised ML models [14,15].

This paper primarily focuses on applying natural language processing (NLP) methods [16–19] for ER data analysis in order to support robust decision-making in a plant operations context. In more detail, our methods are designed to assist system engineers in the identification of anomalous behaviors that might occur in a system (e.g., the periodic failure of a pump control board), the possible cause–effect relations between events (e.g., a lack of adequate flow rate generated by the pump prior to the failure of its control board), and their potential consequences (e.g., pump taken off line which causes power plant derate, and a consequent loss of production). The same methods are also designed to support decision-making such as the scheduling of the appropriate maintenance activities (e.g., a replacement of the pump control board which requires a specific procurement order) and planning based on past operational experience (e.g., identify average time to replace pump control board). In addition, note that trending at the plant level of events of a similar nature (which requires methods to parse a large amount of data automatically rather than relying on manual search) provides insights on key performance indicators of the plant itself, which are under regulatory oversight. All of these tasks are currently performed manually with all limitations that such processes entail (in terms of resources required and efficiency).

Here, the objective in analyzing textual ER data is to move away from supervised/semi-supervised ML model analysis tools [10–13] and to instead automate the extraction of quantitative knowledge from textual data in order to assist system engineers in assessing SSC health trends and identify SSC anomalous behaviors. Knowledge extraction [20–24] is a very broad concept whose definition may vary depending on the application context. When applied to NPP ER textual data (i.e., IRs or WOs), the knowledge extraction approach described herein is designed to extract its syntactic and semantic elements. In more detail, it is designed to identify elements of interest (e.g., types of phenomena described and types of SSCs affected), extract temporal and location attributes, understand the nature of the reported event, and extract causal or temporal relationships between events. This type of NLP analysis has especially been applied in the medical field as shown in [25,26]. However, recent interest has also emerged in other fields including energetic [27], chemical [28,29], bioinformatics [30,31], material science [32], arts and humanities [33], and patent [34] analysis.

Our approach relies on both ML- and rule-based NLP methods designed to identify specific keywords, sentence architecture relations, and structures within each sentence and paragraph. The choice of a rule-based system rather than relying on language models (as, for example, shown in [35]) was dictated by the limitations of the fine-tuning of such models (e.g., the availability of training data) for a very specific field of application (which can also be NPP dependent) and also by security reasons (e.g., sharing data on third-party servers). Applying such analyses to NPP ER textual datasets makes it possible to track the historical health performance of NPP assets and then use the observed health trends to adjust the schedule of future surveillance and maintenance operations [7]. Such a process can have a major impact on the reduction of NPP operational costs. The interest in NLP knowledge extraction methods applied to NPP ER textual data has started only recently. In particular, references [36,37] provide an overview of the advantages that can be reached using technical language processing (TLP) as an iterative human-in-the-loop approach

to analyze NPP textual data to optimize plant operation and asset management. As a result of these considerations, reference [38] provides, to our knowledge, the first attempt to analyze WO textual data using an ontology-based approach. This paper can be seen as an extension of [38] where it also targets the analysis of IRs and other plant textual data (e.g., plant outage data elements). Such an extension does not rely on an ontology as indicated in [38] because of the challenges in constructing a general-purpose ontology that would encompass all possible use cases in an NPP context. Our approach follows some of the elements shown in [39–41], especially in terms or relation extraction and it adapts them into an NPP context.

A relevant observation here is that most of the time, NPP ER textual elements are composed by short (typically about 6–10 words long) sentences that are not properly structured from a grammatical point of view. This poses a challenge when applying the methods described in [21,23,24]. This paper is divided into two parts: Section 2 gives details on each NLP element that constitutes our knowledge extraction workflow, and Section 3 provides examples of applying the developed methods in order to support decision-making in an NPP operational context.

2. Knowledge Extraction Methods

Figure 2 provides an overview of the NLP methods that together constitute the knowledge extraction workflow. These methods are grouped into the following three main categories:

- *Text preprocessing*: The provided raw text is cleaned and processed in order to identify specific nuclear entities and acronyms (e.g., HPI in reference to a high-pressure injection system), and to identify and correct typos (i.e., through a spell check method) and abbreviations (e.g., "pmp" meaning "pump").
- *Syntactic analysis*: The goal of this analysis is to identify the relationship between words contained within a sentence, the focus being on understanding the logical meaning of sentences or parts of sentences (e.g., subjects, predicates, and complements).
- *Semantic analysis*: We rely on the results of this analysis to identify the nature of the event(s) described in the text, along with their possible relationships (temporal or causal).

In the following sections, we provide details on each different NLP method. The methods presented here have been coded in a Python-based coding environment and they leverage a few openly available NLP libraries: `SpaCy` [42], `PySBD` [43], and `nltk` [44]. The choice of the coding environment was also suggested based on current configurations of operating U.S. nuclear plant equipment reliability software suites which store IRs and WOs and allow externally developed data analytics methods to be easily interfaced.

2.1. Spellcheck, Acronym, and Abbreviation Handling

NPP IRs and WOs are often comprised of short sentences that often contain abbreviations. The presence of abbreviations negatively impacts our ability to extract knowledge from such texts. Thus, abbreviations must be identified and then replaced with the complete form of the words. The starting point is a library of word abbreviations collected from documents available online. 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 may have multiple words associated with it. Similarly, a word may be abbreviated in multiple different ways.

In each sentence, abbreviations are handled by first identifying any misspelled words. Each misspelled word is then searched for in the developed library. If an abbreviation in the library matches the misspelled word, the abbreviation is replaced by the complete form of the word. If no abbreviation is found, we proceed by searching for the closest one by employing the Levenshtein distance as a metric. If multiple words match the obtained abbreviation, the one that best fits the context of the sentence is selected.

Acronyms represent another class of textual elements often seen in ER textual data, and typically refer to specific NPP SSCs. They are handled similarly to abbreviations,

with a library of acronyms having been compiled based on publicly available U.S. Nuclear Regulatory Commission (NRC) and Electric Power Research Institute (EPRI) documents.

Once the abbreviations and acronyms have been handled, the remaining misspelled words are run through our spell-checking methods for a final round of corrections. Figure 1 shows an example of spell checking and acronym/abbreviation handling being used to clean up specific words in the raw text.

Figure 1. Example of spell checking ("pmp") and acronym (HPI) and abbreviation ("refurb") handling.

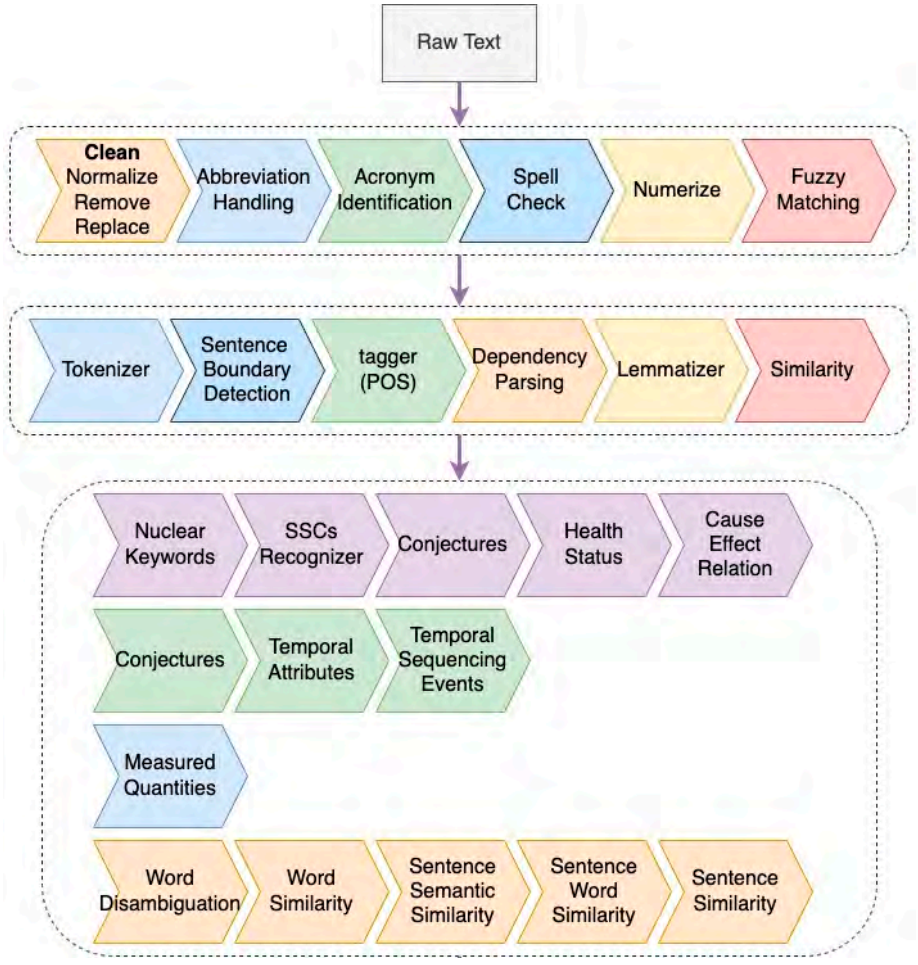

Figure 2. Graphical illustration of the NLP elements that comprise the knowledge extraction workflow.

2.2. Sentence Segmentation

The next important step is to determine the sentence boundaries; that is, segment the text into a list of sentences. This is a key underlying task for NLP processes. For the present work, we employed `PySBD`—a rule-based sentence boundary disambiguation `Python` package—to detect the sentence boundaries. We developed a custom method that uses `PySBD` and `SpaCy` to split raw text into a list of sentences. In general, there are three different approaches to segmenting sentences [16,17]: (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 chose the rule-based approach since the errors are interpretable and the rules can be adjusted incrementally. Moreover, the resulting performance can exceed that of the ML models. For example, `PySBD` passes 97.93% of the Golden Rule Set exemplars (i.e., a language-specific set of sentence boundary exemplars) for English—a 25% improvement over the next-best open-source `Python 3.9` tool (43).

2.3. Tokenization

The next step in textual processing is to tokenize the text [16,17], a process basically designed to segment the text into a list of words or punctuations (see Figure 3). First, the raw text is split based on the whitespace characters. The tokenizer then 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 (e.g., punctuation such as commas, periods, hyphens, or quotation marks)?

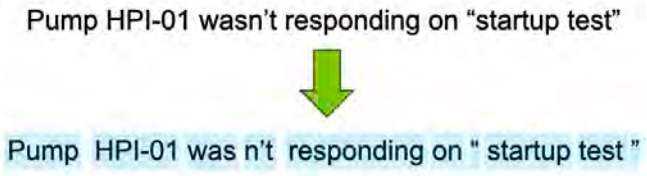

Figure 3. Tokenization process: The tokens obtained from the provided text are highlighted in blue.

If a match is found, the rule is applied and the tokenizer continues its loop, starting with the newly split substrings. In this manner, the tokenizer can split complex, nested tokens such as combinations of abbreviations and multiple punctuation marks.

2.4. Part of Speech

After the correct segmentation of sentences, we rely on the `SpaCy` tagger to parse each sentence and tag each token therein. The "TAG" and "POS" (part of speech) attributes are generated for each token (see Section 2.3). "POS" is the simple universal POS tag (https://universaldependencies.org/u/pos/ [accessed on 4 February 2024]) that does not include information on any morphological features and only covers the word type (e.g., adjectives, adverbs, verbs, and nouns). 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.

The "TAG" attribute expresses both 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 heavily relied on these POS and TAG tags to determine the nature of a given IR or WO (see Section 2.14).

2.5. Dependency Parsing

POS [18] tagging provides information on word types and morphological features but not dependency information between words. Some examples of dependencies are nominal subject (nsubj), direct object (dobj), and indirect object (iobj). The parser uses a variant of the non-monotonic arc-eager transition system described in [42]. The parser uses the terms "head" and "child" to describe those words connected by a single arc in the dependency tree. The dependency labels are used for the arc label, which describes the type of syntactic relation that connects the child to the head. Figure 4 shows a graphic representation of a dependency tree created using `SpaCy`'s built-in `displaCy` visualizer, with the POS tag placed below each word. In the present work, we employed the dependency tree to

develop rules for identifying health information and causal relationships between events (see Sections 2.14 and 2.15, respectively).

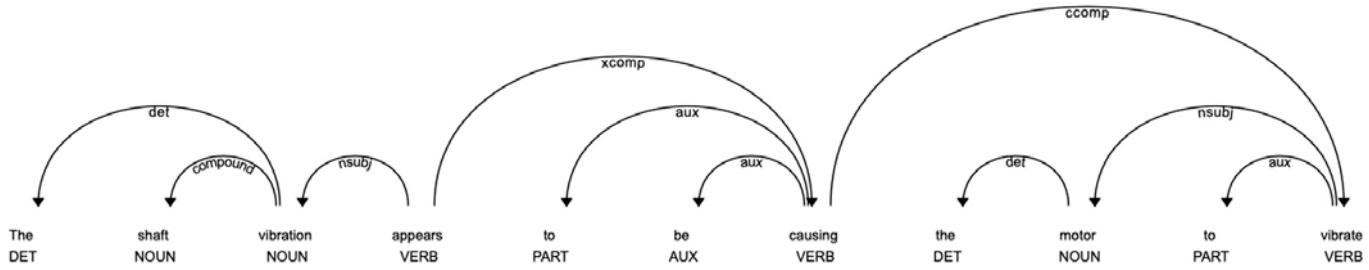

Figure 4. POS tagging and dependency parsing.

2.6. Lemmatization

A lemma is the base form of a token. For 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). For the present study, we employed the `SpaCy` lemmatizer to reduce inflectional or derivationally related forms of words to a common base form. In this case, we only needed to provide the keyword base forms that would significantly reduce the total number of keywords.

2.7. Coreference Resolution

Coreferences often occur in texts in which pronouns (e.g., it, they) are used to reference elements previously mentioned in the text. Coreference resolution is aimed at identifying the textual element linked to the given pronoun. For an example, see Figure 5, in which the pronoun "they" refers to the previously defined textual element "cracks". From our analysis tools, we employed `Coreferee` to resolve coreferences within English texts. `Coreferee` uses a mixture of neural network and programmed rules to identify potential coreference mentions.

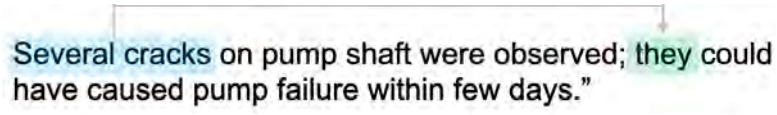

Figure 5. Example of coreference resolution (indicated as an arrow): the pronoun "they" (highlighted in green) refers to the previously defined textual element "cracks" (highlighted in blue).

2.8. Identification of Temporal Quantities

Temporal quantities, which indicate time instances when specific events have occurred, can come in different forms. For the scope of this article, we partitioned these forms into four classes (see Table 1) that specify the occurrence of an event in absolute terms (i.e., date or time) or in relative terms (i.e., duration or frequency). A relevant observation is that the provided temporal information may contain some uncertainty (e.g., an approximated estimate of the temporal occurrence of an event). Such situations were handled by defining a specific list of keywords that indicate approximation, as well as their corresponding set of relations based on observed datasets (see Table 2). The set of temporal relations shown in Table 3 was developed based on [45] and by relying on the large `TimeBank` corpus [46]. Figure 6 shows an example outcome of our identification methods.

Table 1. Examples of date, time, duration, and frequency temporal expression.

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

Table 2. Portion of the list of approximations that might be associated with a temporal attribute.

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

Table 3. 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"

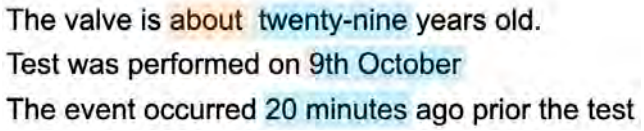

Figure 6. Example identification of temporal (blue) and approximation (orange) attributes.

2.9. Identification of Temporal Sequencing of Events

Another class of textual data elements that can often be retrieved from NPPs is found in IRs covering multiple events linked by temporal relations. Temporal relations can be either quantitative (e.g., an event that occurred two hours after another event) or qualitative (e.g., an event that occurred prior to another event). Note that a temporal relation does not necessarily imply a causal relation. In this paper, we build on the work in [47], which lists the major temporal relations between events:

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

Note that event duration is considered a temporal attribute (see Section 2.8). An analysis of sentences containing temporal relations involves identifying specific keywords, relations, and grammatical structures in each sentence—similarly to what was presented in Section 2.8. In this respect, Tables 4 and 5 provide the set of keywords (i.e., verbs, adjectives, and adverbs) that were identified for order, concurrence, and coincidence of events. A set of grammatical structures that indicate the order and coincidence of events was also developed (see Tables 6 and 7, respectively). The example provided in Figure 7 shows two identified temporal attributes that indicate a temporal sequence and concurrency of events.

Table 4. Example of keywords and structures that indicate the order of events.

Keywords			Structures
Verbs	**Adjectives**	**Adverbs**	
Antedate Follow Postdate Precede Predate Succeed	After Before Consecutive Earlier Following Former Later Next Past Precedent Previous	Afterward Consecutively Consequently Directly Hereafter Later Next Previously Subsequently Successively Then	Soon after After that After a while

Table 5. 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 6. 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 7. List of relations that indicate the concurrence and coincidence of events.

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

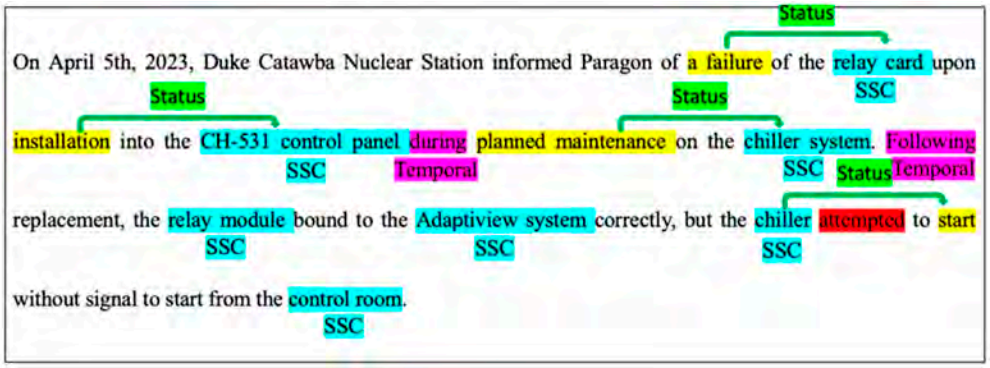

Figure 7. Example analysis of sentences containing temporal entities (highlighted in purple) identified from https://www.nrc.gov/docs/ML2320/ML23207A076.pdf (accessed on 4 February 2024).

2.10. Identification of Measured Quantities

Next, we aimed to identify a precise observation (i.e., a measured point value or delta estimate) of a measured variable. This observation required a numeric value followed by its unit; however, it is not unusual for the unit to be missing. Note that, based on the observed NPP ER textual data, measured quantities can be specified in a large variety of ways (see Table 8 for examples), and not solely in the classic form "number + unit of measure".

Table 8. Examples of quantitative observations.

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

This list was based on [48] and it was tested using openly available scientific literature. We leverage `quantulum3` and text syntactic relations listed in Table 9 to extract measured quantities. The tool `quantulum3` can identify all possible numerical, values either with or without units, whereas syntactic information helps disambiguate the units from the natural language.

Table 9. List of sentence relations for quantitative observation.

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

Figure 8 gives an example of identifying measured quantities. The textual elements were taken from a few different NRC licensee event reports. The correctly identified quantities are highlighted in blue, the rest are highlighted in red. As seen, the developed method leads to issues regarding certain specific situations: namely, unknown units of measures (e.g., Gy) and unit prefixes (e.g., milliRem instead of mRem). We are currently working to address such limitations by making new improvements to `quantulum3` and implementing ad-hoc methods whenever these limiting situations are encountered.

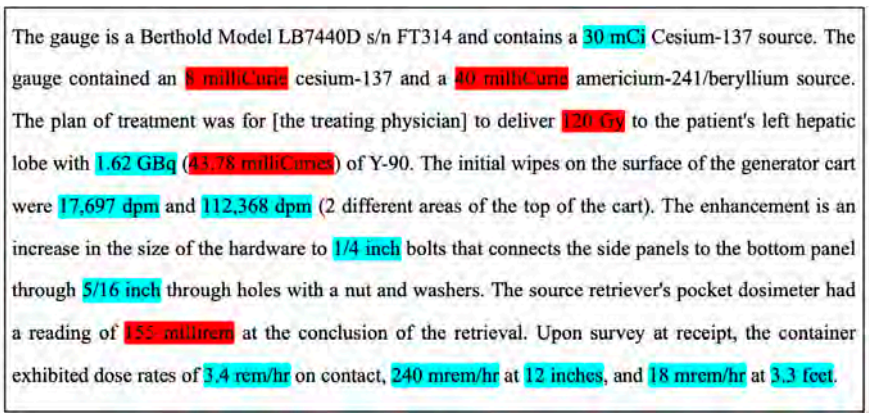
The gauge is a Berthold Model LB7440D s/n FT314 and contains a 30 mCi Cesium-137 source. The gauge contained an 8 milliCurie cesium-137 and a 40 milliCurie americium-241/beryllium source. The plan of treatment was for [the treating physician] to deliver 120 Gy to the patient's left hepatic lobe with 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 8. Example of identifying measured quantities from text taken from https://www.nrc.gov/reading-rm/doc-collections/event-status/event/2020/index.html (accessed on 4 February 2024).

2.11. Identification of Location Attributes

As with temporal attributes, location attributes provide qualitative information, in this case, information on where specific events have occurred. While location information does not equip system engineers with any additional health information, it might give clues about the health of a specific component whenever a reported event has occurred nearby it. For example, the textual report "An oil puddle was found nearby pump MFW-1A" identifies an element (i.e., oil) that may have a relation to a nearby pump (i.e., MFW-1A pump). In the literature, this type of attribute search is not of interest; however, from a safety/reliability standpoint, such information can be crucial for identifying the causes behind abnormal behaviors observed throughout an NPP.

Location attributes are identified by looking at the specific keywords and relations listed in Tables 10 and 11, respectively. Regarding the list of keywords listed in Table 10, we relied on an initial set of keywords that was then expanded using `WordNet` (WordNet is a lexical database originally created by Princeton University. It contains words, their meanings (e.g., synsets), and their semantic relationships, all of 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 degree of similarity between them.) [49] synonym search capabilities. Figure 9 shows an example of identifying location attributes. (The textual elements were taken from a few NRC licensee event reports.) In this case, the identification of these attributes was very robust.

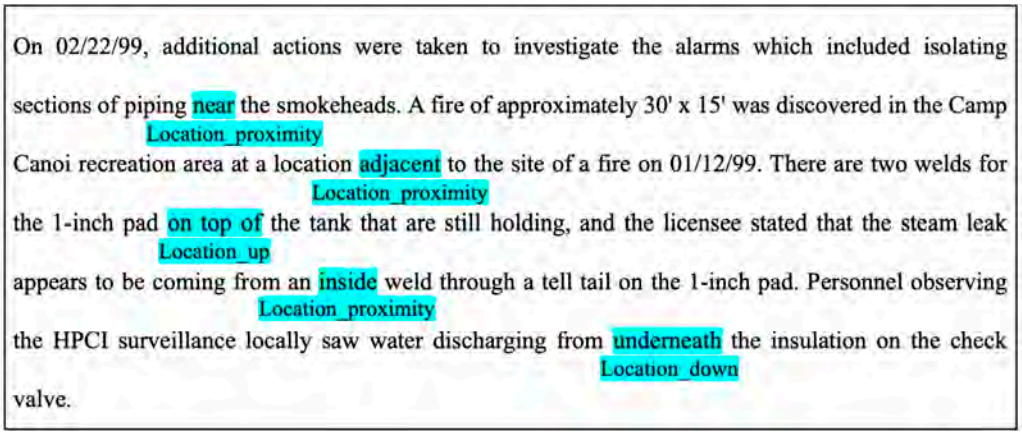

Figure 9. Example of identifying location attributes from text taken from https://www.nrc.gov/reading-rm/doc-collections/index.html#event (accessed on 4 February 2024).

Table 10. Example keywords that indicate a location attribute.

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

Table 11. List of relations that indicate a location attribute.

Relations
[verb] + "location keyword" + noun Subj + "location keyword" + obj

2.12. Identification of Nuclear Entities

NLP knowledge extraction methods require the ability to identify specific entities such as common SSCs that can be found in any NPP. A library for light water reactors has been developed in past years using available textual data form the NRC and EPRI. The entities contained in this library (numbering about 5000 and growing) are arranged into eight main classes and then subsequently divided into groups (mainly for data management purposes). Table 12 lists the various classes and groups created so far, along with examples of entities corresponding to each group.

Table 12. Class and groups of nuclear-related keywords.

Class	Group	Examples
Mechanical components	Fasteners	Anchor bolt, cap screw, latch, pin
	Rotary elements	Cam, shaft, gear, pulley
	Structural	Beam, column, sleeve, socket
	Purpose-specific	Filter, manifold, blade
Non-mechanical components	Electrical/electronic	Amplifier, relay, buzzer, capacitor
	Hydraulic/Pneumatic	Coupler, filter, pipe
Assets	Mechanical	Engine, vessel
	Electrical	AC bus, alternator, generator, transformer
	Hydraulic/Pneumatic	Pump, valve, condenser, fan
	Electronic	Computer, tablet, controller
	I&C	Digital meter, FPGA, transmitter, sensor
	Nuclear fuel	Fuel rod, control blade
NPP elements	Systems	Feedwater, switchyard, feedwater
	Architectural	Containment, control room, pump house
Tools and treatments	Tools	Jigsaw, solder gun, tape, crane
	Treatments	Bolting, riveting, grinding, infrared testing
Operands	Electrical	AC current, electromagnetic
	Hydraulic/Pneumatic	Compressed air, steam, gasoline, water
Compounds	Materials	Plastic, plywood, concrete, polyethylene
Reactions	Chemical reaction	Combustion, oxidation, evaporation
	Degradation mechanism	Corrosion, dissolution, fatigue
	Failure type	Leak, rupture, brittle fracture

Using this list, the goal is now to identify these types of entities within a textual data element. For the present work, we relied on `SpaCy` name entity recognition (NER) functions [50] to perform such searches. Identified entities were flagged with a specific tag ID and saved as part of the metadata associated with the textual data. Figure 7 provides an example of the outcome of the developed nuclear entity NER methods, with several elements, highlighted in blue, having been correctly identified.

2.13. Identification of Conjectures

In this step, we consider textual elements that contain information about future predictions (e.g., an event that may occur in the future) or hypotheses regarding past events (e.g., a failure that may have occurred). Even if the reported event has not occurred (or may not happen), this evaluation might be relevant for future diagnosis (identifying possible causes from observed events) or prognosis (identifying consequences from observed phenomena)

purposes. In this context, verb tense plays a role in identifying this kind of report. Future predictions are characterized by present- and future-tense verbs, whereas hypotheses about past events are typically characterized by past-tense verbs. Hence, we rely on the outcomes of the methods presented in Sections 2.4 and 2.5 in order to perform such syntactic analyses. Additionally, we developed an initial set of specific keywords (see Table 13) and relations (see Table 14) that can inform our methods whenever we are dealing with a conjecture observation. Once a conjecture is identified from a textual data element, a conjecture flag is set to "True" as part of the metadata associated with the textual data.

Table 13. Examples of keywords that indicate a conjecture observation.

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

Table 14. 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

2.14. Identification of Health Status

So far, we have demonstrated the capability to identify quantitative health information associated with an SSC when the textual report provides a precise observation (i.e., numeric value) of a measured variable (see Section 2.10), its proximity location (see Section 2.11), and its temporal attributes (see Section 2.8). Often, IRs reflect qualitative information on abnormal observed events (e.g., failures, or precursors to a degradation phenomenon). From a reliability standpoint, identifying the nature of the reported event plays a major role, with the goal being to track the health performance of a single SSC or multiple SSCs operating in similar operating conditions.

Based on the large number of IRs and WOs gathered from operating NPPs in the United States, and using the methods presented in Sections 2.4 and 2.5, we collected and extracted the underlying grammatical structures and converted them into relations (see Table 15). Similarly, a list of keywords (nouns, verbs, adverbs, and adjectives) for indicating the health status of a generic SSC is shown. These keywords have been partitioned into three main classes (see Tables 16–18) based on sentiment analysis [51], and then expanded using the `WordNet` [49] synonym search capabilities. Thus, identification of the health status of the textual clause can be assessed by searching in the text for the developed lists of relations and keywords.

Table 15. List of sentence relations for making qualitative observations.

Relation	Example
Subj + "status verb"	Pump was not functioning
Subj + "status verb" + "status adjective"	Pump performance was 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 16. 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

Table 17. 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 18. Partial list of keywords that indicate neutral information.

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

2.15. Identification of Cause–Effect Relations

An occasional pattern in textual ER data is the reporting of multiple events as well as the causal relationship among them. In this regard, the simplest type of paragraph found in textual ER data will refer to an event (i.e., the cause) that triggered a second event (i.e., the effect). However, variations in such paragraphs do exist (see Figure 10): multiple causes can trigger a single effect, or a single cause can trigger multiple effects.

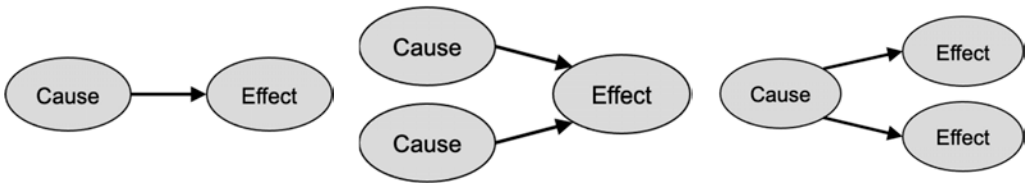

Figure 10. Graphical representation of elemental cause–effect structures: direct cause–effect association (**left**), multiple causes and single effect association (**center**), multiple effects and single cause association (**right**).

Here, we did not employ ML algorithms (e.g., through the utilization of classification methods [52]), but instead once again relied on rule-based [53] methods, since our goal was to extract quantitative information from textual data rather than "classify" the nature of the raw text. In other terms, rather than just classifying the textual data element as to

whether it does or does not contain a causal statement, we aim to identify which element is the cause and which is the effect. Similarly to what was described in Section 2.14, these rules are based on the identification of the following:

- Keywords (e.g., nouns, verbs, and adverbs) that reflect that the sentence may contain a causal relation between its subject(s) and object(s) (see Table 19). We successfully expanded out the initial set of keywords by using the `WordNet` [49] synonym search capabilities.
- Relations between subjects and verbs contained in a sentence that are designed to reconstruct the causal relations (see Table 20). The list of these relations was developed by applying the methods described in Sections 2.4 and 2.5 to a portion of the `CausalBank` [54] dataset, which contains about 314 million pairs of cause–effect statements.
- NLP relations composed of multiple words that indicate a casual transition between clauses contained in a sentence or between sentences (see Table 21).

Table 19. Partial list of keywords that indicate a cause–effect paragraph.

Nouns	Verbs	Adverbs
Augment	Augment	
Backfire	Backfire	
Begin	Begin	Afterwards
Bring about	Bring about	Consequently
Build-up	Build-up	Eventually
Cause	Cause	Finally
Change	Change	Hence
Combat	Combat	So
Compensate	Compensate	Subsequently
Counter	Counter	Then
Create	Create	Therefore
Deactivate	Deactivate	Thus
Decelerate	Decelerate	Ultimately
Decrease	Decrease	

Table 20. List of relations that indicate a cause–effect paragraph.

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 21. 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

We applied the developed cause–effect identification methods to the publicly available NRC LER 2021-001-00, "Atmospheric Steam Dump Valves Inoperable Due to Relay Failure". In this context, Figure 11 presents a subset of three cause–effect relations that were identified. In particular, for each of the three identified relations, the figure shows the original text and details about the relation, per the following format: "*(cause, status), cause-effect keyword, (effect, status)*".

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 11. Example of identifying cause–effect relations (source: NRC LER 2021-001-00, "Atmospheric Steam Dump Valves Inoperable Due to Relay Failure").

An initial testing of the capabilities of the developed methods was performed on an openly available dataset generated within SemEval. In particular, we considered the SemVal2010_task8 dataset [55] built to test the performance of NLP methods regarding the discovery of causal relations. The performances were measured in terms of precision (as the ration between true positives over the sum of true positives and false positives) and recall (as the ration between true positives over the sum of true positives and false negatives). The obtained values for precision and recall were estimated as 68% and 88%, respectively. The performances were measured by looking at the subset of sentences in the dataset that were originally labeled as "cause-effect". Through a careful investigation, our methods were labeling as "cause-effect" some sentences originally labeled as "Product-Producer". In some of these cases those sentences were actually containing a cause–effect relation that we wanted to identify. Thus, the actual performances could be better.

2.16. Identification of Text Similarity

Word, sentence, and document similarity analyses are part of NLP, and play a crucial role in text analytics (e.g., text summarization and representation, text categorization, and knowledge discovery). A wide variety of methodologies have been proposed during the last two decades [56,57], and can mostly be classified into five groups: (1) lexical knowledge base approaches, (2) statistical corpus approaches (word co-occurrence), (3) ML and deep learning approaches, (4) sentence-structure-based approaches, and (5) hybrid approaches. However, a few common major drawbacks stem from these approaches: computational inefficiency, a lack of automation, and a lack of adaptability and flexibility.

In the present work, we attempted to address these drawbacks by developing a tool that is generally usable in applications requiring similarity analysis. As shown in Figure 12, we 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 a syntactic dependency (DEP) tag. Such data will provide syntactic structure information (i.e., negation, conjecture, and syntactic dependency) about the sentence, and this information can be used to guide the similarity measuring process.

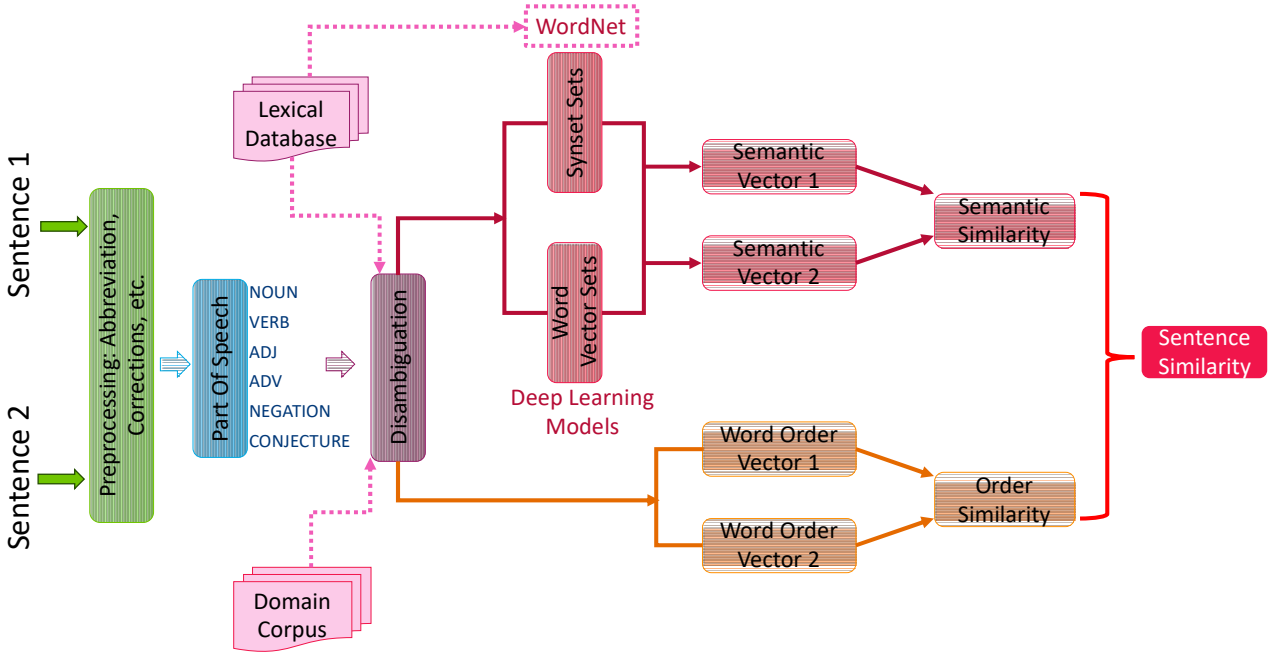

Figure 12. Illustration of the sentence similarity calculation.

Disambiguation is employed to determine the best sense of the word, especially when coupled with specific domain corpus. It ensures the right meaning of the words (e.g., the right synsets of the words in a lexical database) within the sentence is captured. A predefined word hierarchy from a lexical database (i.e., `WordNet`) is then used to calculate the degree of word similarity. However, some words are not contained in the lexical database, as it only connects four POS types: nouns, verbs, adjectives, and adverbs. Moreover, these words are grouped separately and do not feature any 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, ML-based word embedding is introduced to enhance the similarity calculation. Regarding the previous example, the similarity score then becomes 0.715. 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, whereas syntactic similarity is measured based on word order similarity. The following sections further describe each of the steps in more detail.

As mentioned in Sections 2.4 and 2.5, POS data provide information on word types and morphological features, and dependency parsing provides information on the syntactic dependency between words. Both POS and dependency parsing can help identify important information such as NOUN, VERB, ADJ, ADV, negation, conjecture, subject, and object, and this information is then used to compute the sentence syntactic similarity.

Lexical databases such as `WordNet` consider semantic connections between words, and this can be utilized to determine their semantic similarity. As summarized by [58], many different methods can be employed 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 [59,60] to compute the similarity score between two words/synsets, here indicated as w_1 and w_2, as presented in Equation (1):

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

$$with\ f_{length}(l) = e^{-\alpha l}\ \ g_{depth}(d) = \frac{e^{\beta d} - e^{-\beta d}}{e^{\beta d} + e^{-\beta d}}$$

where the following apply:

- l indicates the path length between w_1 and w_2.
- d indicates the path depth between w_1 and w_2.
- $f_{length}(l)$ and $g_{depth}(d)$ are functions which decompose the contribution to S_w respectively for path length and depth between w_1 and w_2.
- $\alpha \in [0, 1]$, $\beta \in (0, 1]$ are scaling parameters for the contribution of the 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 in [60].

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 when their subsumer is located at the lower layer of the hierarchy. This is because the lower layer contains more specific features and semantic information than does the upper layer.

The "sense" of a given word represents its precise meaning under a specific context. Disambiguation is the process used to identify which sense of the word is best in the context of a particular statement. Without proper disambiguation, errors may be introduced at the early stage of the similarity calculation when using lexical databases. For example, in `WordNet`, synsets denote the senses of the word, and are linked to each other via their explicit semantic relationships. When different synsets are used to calculate word pair similarity, their semantic relationship can be drastically different, potentially having a significant effect on the similarity score. In the present work, we tried to disambiguate the word sense by considering the context of the word. One way to do this is to account for the surrounding words, since they can provide contextual information. However, this may not work for simple or short sentences. In such cases, the domain-specific corpus can be leveraged to disambiguate the word. Once the best senses are identified for the words, the word similarity measure can be employed.

As proposed in [58], sentence similarity encompasses both semantic and syntactic similarity. Semantic similarity is captured via word semantic similarity, as discussed in previous sections, whereas syntactic similarity is measured by word order similarity. Word order similarity affords a way to assess sentence similarity in consideration of word order. As is well described in [58], the constructed semantic vectors and word order vectors can be used to compute sentence similarity. Here, we will briefly introduce the methods of constructing these vectors, and recommend that the reader refer to [58] for additional details.

Given two sentences, T_1 and T_2, a joint word set is formed (e.g., $T = T_1 \cup T_2$) that incorporates all of the distinct words from T_1 and T_2. The vectors derived from computing word similarities in $(T,\ T_1)$ and $(T,\ T_2)$ are called the semantic vectors, and are 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, such that 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 in [58], the word order similarity of two sentences is defined as follows:

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

where the word order vectors r_1 and r_2 are formed from (T, T_1) and $(T,\ T_2)$, 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)Both semantic and syntactic information (in terms of word order) play a role in measuring sentence similarity. Thus, the overall sentence similarity is defined in [58] as follows:

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

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

3. Applications of NLP Knowledge Extraction Methods

In current U.S. nuclear power plants, IRs and WOs are typically generated in digital form using pre-defined formats and they are stored in databases along with all of the information about plant operations (e.g., surveillance and maintenance). Such databases can be filtered depending on the type of analyses to be performed and locally downloaded in standard formats (typically in a comma separated value format). In our case, plant IRs and WOs are retrieved from plant databases as comma separated value format data files and then they are converted into a `Pandas` DataFrame. Each NLP function described in Section 2 has been coded as a stand-alone method that acts on a set of sentences which are stored as a `Pandas` DataFrame. Each method is designed to sequentially parse all sentences and either flag text elements (e.g., nuclear-related keyword) or populate a new column of the database (e.g., an assessment of conjecture or causal relation between events). Thus, depending on the desired application, the user can create workflows which consist of a set of methods described in Section 2 that operates sequentially on the same `Pandas` DataFrame. Note this modus operandi can be applied directly once a new IR or WO has been generated (i.e., online mode). Sections 3.1 and 3.2 provide details about the application of the methods described in Section 2 in two different operational scenarios. The first one focuses directly on NER and knowledge extraction from textual data to identify anomalous behaviors while the second one is designed to support the planning of NPP outage.

3.1. Analysis of NPP ER Data

The examples provided here are designed to demonstrate how the methods described in Section 2 can be used to process NPP IRs. In general, such text preprocessing is manual and potentially very time-consuming. In these examples, we have collected a list of typical IR descriptions (see Table 22) to test the effectiveness of such methods.

Table 22 shows the first example, with the extracted SSC entities and their health status highlighted in blue and yellow, respectively. For a better illustration of the extracted data, Table 23 presents the pair of extracted SSC entities and their health statuses. Note that there are two misidentifications highlighted in green. The first, (*pump, test*), is easily resolved if we also include the health status keyword "failed" (highlighted in red) in the health status, as marked in Table 22. Two health status options exist for the second misidentification: "found in proximity of rcp" and "oil puddle". To determine the correct health status for "pump", we employed word/phrase/sentence similarity (see Section 2.16) in order to compute the similarity scores between the SSCs and their potential health statuses. The one with the highest similarity score is selected as the identified health status. In this case, the similarity score between "puddle" and "pump" is 0.25, whereas that between "proximity" and "pump" is 0.027. Thus, "puddle"—with the additional information "oil"—is selected as the final health status for "pump".

Table 22. Example of information extraction. The following are identified in the text: nuclear entities (highlighted in blue), health status (highlighted in yellow), keywords indicating health status (highlighted in red).

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 23. Extracted SSC entities and their health status from the text provided in Table 22. Misidentifications are highlighted in green.

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 in regard to the text given in Table 22 are presented in Table 24. We employed a set of rule templates based on specific trigger words and relations (see Section 2.15). Once the SSCs entities and their health status were identified, we could apply these rules to identify the cause–effect relations. One cause–effect relation remained uncaptured, as "safety cage" was not originally listed as the identified SSC entity.

Table 24. Causal relations identified (nuclear keywords are highlighted in blue while health status are highlighted in yellow).

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 coreference identification. This process is intended to find expressions that refer to the same entity in the text—something that is of particular relevance in light of a lengthy piece of text that refers to an entity by using a pronoun rather than its proper name. Using our methods, the coreferences in the text presented in Table 22 can be identified, as shown in Table 25.

Table 25. 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 that the information provided by the sentence pertains to a future prediction (e.g., an event that may occur in the future) or a hypothesis about past events (e.g., a failure that may have occurred). In this context, verb tense plays a role in identifying these kinds of attributes. Future predictions are characterized by both present- and future-tense verbs; hypotheses about past events are typically characterized by past-tense verbs. Based on the text provided in Table 22, the sentences containing conjecture information were correctly identified and are listed in Table 26.

Table 26. 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.

3.2. Analysis of Plant Outage Data

Refueling outages are among the most challenging phases in an NPP's operating cycle. NPP outages require the scheduling of thousands of activities within an average of 30 days. During the outage planning phase, the outage schedule is determined via optimization tools, given the estimated time to perform each activity. Such temporal estimation is performed manually based on past operational experience.

The goal here is to perform the same task—but by applying the text similarity methods described in Section 2.16 to past outage data regarding activities performed during past

outages and the actual completion time for each activity. In other words, we aim to identify a subset of activities performed in previous outages that are similar to the activity being queried. The temporal distribution of the completion time associated with the queried activity can then be determined by collecting the historical completion time from the selected subset of (similar) past activities.

We now give an example of temporal distribution estimation—presented here for the queried activity "valve re-packing"—using a dataset provided by an existing U.S. NPP. The dataset contains activities performed over the course of five different outages. Data cleaning was performed for each of these activities. Once the historical plant outage data were cleaned via the methods presented in Sections 2.1–2.3, the similarity value between the queried activity and each historical activity was determined using the methods presented in Section 2.8. This resulted in an array of similarity values having dimensionality identical to the number of historical activities and the corresponding array (with identical dimensionality) containing the activity durations (see Figure 13). Note that the temporal values were intentionally perturbed to disguise proprietary data.

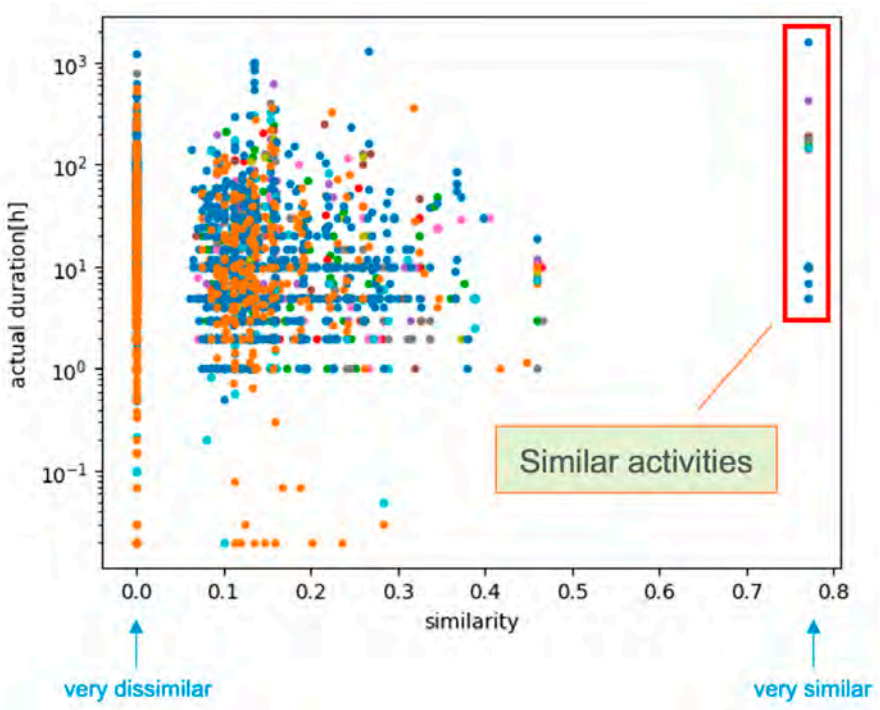

Figure 13. Scatter plot of all past outage activities in terms of actual duration and similarity values. Activities similar to the queried one (i.e., "valve re-packing") are highlighted in the red box.

The temporal distribution of the queried activity was determined by considering both the similarity and duration arrays. More precisely, we selected activities such that the similarity measure exceeded a specified threshold (typically in the 0.7–0.9 range). Of particular note here is that if a queried activity was never completed in past outages, no similar past activities will be found. This approach does not in fact perform any type of regression. The output consists of a histogram representing the duration variance to complete the queried activity upon being provided past outage data (see Figure 14). Given these results, the analysis now carries the potential to statistically analyze the actual duration of similar activities in order to identify possible outliers obtained from the similarity search, track the historical trend in activity completion time, and evaluate the impact of employed human resources on completion time.

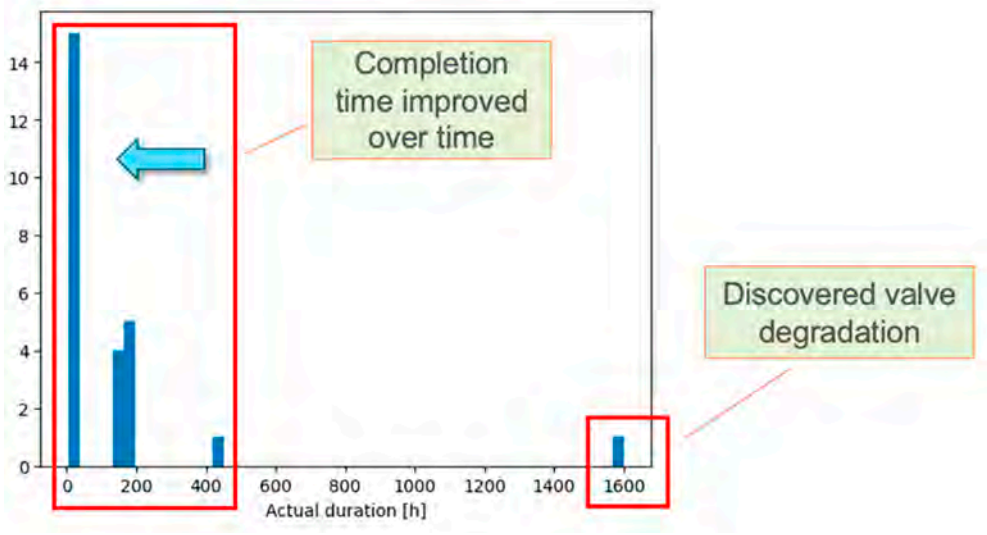

Figure 14. Example similarity search results: a histogram representing the duration variance to complete the queried activity by selecting the activities highlighted in red in Figure 13.

4. Conclusions

This paper presented an overview of a computational tool designed to extract information from ER textual data generated by NPPs. This tool consists of several methods aimed at parsing sentences in search-specific text entities (e.g., measured quantities, temporal dates, and SSC). The semantic analysis tools are designed to then capture the semantic meaning of the event(s) described in the provided texts, including health information, cause–effect relations, or temporal sequences of events. Of importance here is the set of preprocessing tools devised to clear textual elements from acronyms, abbreviations, and grammatical errors. Such cleaning methods are essential for improving the performance of the knowledge extraction methods.

We presented a few applications of the methodology that extended beyond the analysis of NPP IRs and WOs. In these applications, despite the ER textual elements being short by nature, our tools successfully extracted the semantic meaning and identified the vast majority of the specified entities. We also indicated how our sentence similarity measures can be used to parse past outage databases in order to inform plant outage managers of the historical durations required to complete specific activities. Analyses of NRC reports provided a few good examples of how our methods can capture the cause–effect or temporal relations among different events.

The capabilities of the developed tools are unique in the nuclear arena, and are based on the parallel development that is taking place in the medical field. As a matter of fact, we relied on a few libraries initially developed to conduct knowledge extraction from medical textual data elements (e.g., patients' medical reports and doctor diagnoses). Extending such methods to a different field, namely, nuclear energy, required the development of additional methods and libraries to fit the new use cases.

Author Contributions: Methodology, C.W. and D.M.; Software, C.W., J.C. and D.M.; Formal analysis, D.M.; Writing—original draft, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was made possible through funding by the United States Department of Energy's Light Water Reactor Sustainability Program under contract No. DE-AC05-00OR22725.

Data Availability Statement: Data employed in this paper was either proprietary or taken from openly available documents as indicated throughout the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Banks, J.; Merenich, J. Cost Benefit Analysis for Asset Health Management Technology. In Proceedings of the Proceedings Annual Reliability and Maintainability Symposium, Orlando, FL, USA, 22–25 January 2007; pp. 95–100.
2. Zio, E.; Compare, M. Evaluating maintenance policies by quantitative modeling and analysis. *Reliab. Eng. Syst. Saf.* **2013**, *109*, 53–65. [CrossRef]
3. Compare, M.; Baraldi, P.; Zio, E. Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0. *IEEE Internet Things J.* **2020**, *7*, 4585–4597. [CrossRef]
4. Pipe, K. Practical prognostics for Condition Based Maintenance. In Proceedings of the 2008 International Conference on Prognostics and Health Management (PHM), Denver, CO, USA, 6–9 October 2008; pp. 1–10.
5. Vichare, N.; Pecht, M. Prognostics and health management of electronics. In *Encyclopedia of Structural Health Monitoring*; Wiley: Hoboken, NJ, USA, 2009. [CrossRef]
6. Zhang, W.; Yang, D.; Wang, H. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. *IEEE Syst. J.* **2019**, *13*, 2213–2227. [CrossRef]
7. Zio, E. Data-driven prognostics and health management (PHM) for predictive maintenance of industrial components and systems. *Risk-Inf. Methods Appl. Nucl. Energy Eng.* **2024**, *2024*, 113–137.
8. Coble, J.; Ramuhalli, P.; Bond, L.; Hines, J.W.; Upadhyaya, B. A review of prognostics and health management applications in nuclear power plants. *Int. J. Progn. Heal. Manag.* **2015**, *6*, 2271. [CrossRef]
9. Zhao, X.; Kim, J.; Warns, K.; Wang, X.; Ramuhalli, P.; Cetiner, S.; Kang, H.G.; Golay, M. Prognostics and Health Management in Nuclear Power Plants: An Updated Method-Centric Review With Special Focus on Data-Driven Methods. *Front. Energy Res.* **2021**, *9*, 696785. [CrossRef]
10. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
11. Park, J.; Kim, Y.; Jung, W. Use of a Big Data Mining Technique to Extract Relative Importance of Performance Shaping Factors from Event Investigation Reports. In *International Conference on Applied Human Factors and Ergonomics*; Springer: Cham, Switzerland, 2017; pp. 230–238.
12. Zhao, Y.; Diao, X.; Huang, J.; Smidts, C. Automated identification of causal relationships in nuclear power plant event reports. *Nucl. Technol.* **2019**, *205*, 1021–1034. [CrossRef]
13. Al Rashdan, A.; Germain, S.S. Methods of data collection in nuclear power plants. *Nucl. Technol.* **2019**, *205*, 1062–1074. [CrossRef]
14. Zhu, X.; Goldberg, A.B.; Brachman, R.; Dietterich, T. *Introduction to Semi-Supervised Learning*; Morgan and Claypool Publishers: San Rafael, CA, USA, 2009.
15. Chapelle, O.; Schlkopf, B.; Zien, A. *Semi-Supervised Learning*, 1st ed.; The MIT Press: Cambridge, MA, USA, 2010.
16. Jurafsky, D.; Martin, J. *Speech and Language Processing*; Pearson International Edition: London, UK, 2008.
17. Indurkhya, N.; Damerau, F.J. *Handbook of Natural Language Processing*, 2nd ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2010.
18. Clark, A.; Fox, C.; Lappin, S. *The Handbook of Computational Linguistics and Natural Language Processing*, 1st ed.; John Wiley & Sons: New York, NY, USA, 2012.
19. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimedia Tools Appl.* **2023**, *82*, 3713–3744. [CrossRef]
20. Baud, R.H.; Rassinoux, A.-M.; Scherrer, J.-R. Natural Language Processing and Semantical Representation of Medical Texts. *Methods Inf. Med.* **1992**, *31*, 117–125. [CrossRef] [PubMed]
21. Mooney, R.J.; Bunescu, R. Mining knowledge from text using information extraction. *ACM SIGKDD Explor. Newsl.* **2005**, *7*, 3–10. [CrossRef]
22. Sbattella, L.; Tedesco, R. Knowledge Extraction from Natural Language. In *Methodologies and Technologies for Networked Enterprises*; Lecture Notes in Computer Science; Anastasi, G., Bellini, E., Di Nitto, E., Ghezzi, C., Tanca, L., Zimeo, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7200. [CrossRef]
23. Krallinger, M.; Rabal, O.; Lourenco, A.; Oyarzabal, J.; Valencia, A. Information retrieval and text mining technologies for chemistry. *Chem. Rev.* **2017**, *117*, 7673–7761. [CrossRef]
24. Yan, R.; Jiang, X.; Wang, W.; Dang, D.; Su, Y. Materials information extraction via automatically generated corpus. *Sci. Data* **2022**, *9*, 401. [CrossRef] [PubMed]
25. Chasseray, Y.; Barthe-Delanoë, A.-M.; Négny, S.; Le Lann, J.-M. Knowledge extraction from textual data and performance evaluation in an unsupervised context. *Inf. Sci.* **2023**, *629*, 324–343. [CrossRef]
26. Björne, J.; Salakoski, T. Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing. In Proceedings of the BioNLP 2018 Workshop, Melbourne, Australia, July 2018; pp. 98–108. Available online: https://aclanthology.org/W18-2311/ (accessed on 1 February 2024).
27. VanGessel, F.G.; Perry, E.; Mohan, S.; Barham, O.M.; Cavolowsky, M. Natural language processing for knowledge discovery and information extraction from energetics corpora. *Propellants Explos. Pyrotech.* **2023**, *48*, 109. [CrossRef]
28. Shetty, P.; Ramprasad, R. Machine-Guided Polymer Knowledge Extraction Using Natural Language Processing: The Example of Named Entity Normalization. *J. Chem. Inf. Model.* **2021**, *61*, 5377–5385. [CrossRef]

29. Yang, X.; Zhuo, Y.; Zuo, J.; Zhang, X.; Wilson, S.; Petzold, L. PcMSP: A dataset for scientific action graphs extraction from polycrystalline materials synthesis procedure text. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 6033–6046.
30. Bravo, Á.; Piñero, J.; Queralt-Rosinach, N.; Rautschka, M.; Furlong, L.I. Extraction of relations between genes and diseases from text and large-scale data analysis: Implications for translational research. *BMC Bioinform.* **2015**, *16*, 55. [CrossRef]
31. Giorgi, J.; Bader, G.; Wang, B. A sequence-to-sequence approach for document-level relation extraction. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 10–25.
32. Weston, L.; Tshitoyan, V.; Dagdelen, J.; Kononova, O.; Trewartha, A.; Persson, K.A.; Ceder, G.; Jain, A. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *J. Chem. Inf. Model.* **2019**, *59*, 3692–3702. [CrossRef]
33. Alani, H.; Kim, S.; Millard, D.E.; Hall, M.J.; Weal, W.W.; Hall, M.J.; Lewis, W.; Shadbolt, N.R.; Paul, H. Automatic ontology-based knowledge extraction and tailored biography generation from the web. *IEEE Intell. Syst.* **2002**, *18*, 14–21. [CrossRef]
34. Souili, A.; Cavallucci, D.; Rousselot, F. Natural Language Processing (NLP)—A Solution for Knowledge Extraction from Patent Unstructured Data. *Procedia Eng.* **2015**, *131*, 635–643. [CrossRef]
35. Dagdelen, J.; Dunn, A.; Lee, S.; Walker, N.; Rosen, A.S.; Ceder, G.; Persson, K.A.; Jain, A. Structured information extraction from scientific text with large language models. *Nat. Commun.* **2024**, *15*, 1418. [CrossRef] [PubMed]
36. Brundage, M.P.; Sexton, T.; Hodkiewicz, M.; Dima, A.; Lukens, S. Technical language processing: Unlocking maintenance knowledge. *Manuf. Lett.* **2021**, *27*, 42–46. [CrossRef]
37. Dima, A.; Lukens, S.; Hodkiewicz, M.; Sexton, T.; Brundage, M.P. Adapting natural language processing for technical text. *Appl. AI Lett.* **2021**, *2*, 33. [CrossRef] [PubMed]
38. Woods, C.; Selway, M.; Bikauna, T.; Stumptnerb, M.; Hodkiewiczc, M. An Ontology for Maintenance Activities and Its Application to Data Quality. *Semant. Web.* **2023**, *2023*, 3067–4281. [CrossRef]
39. Han, X.; Gao, T.; Lin, Y.; Peng, H.; Yang, Y.; Xiao, C.; Liu, Z.; Li, P.; Zhou, J.; Sun, M. More data, more relations, more context and more openness: A review and outlook for relation extraction. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, 4–7 December 2020; pp. 745–758. Available online: https://aclanthology.org/2020.aacl-main.75 (accessed on 1 February 2024).
40. Zhuang, W. Architecture of Knowledge Extraction System based on NLP. In Proceedings of the ICASIT 2021: 2021 International Conference on Aviation Safety and Information Technology, Changsha, China, 18–20 December 2021; pp. 294–297.
41. Shimorina, A.; Heinecke, J.; Herledan, F. Knowledge Extraction From Texts Based on Wikidata. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, Seattle, WA, USA, 10–15 July 2022; pp. 297–304.
42. Honnibal, M.; Montani, I. SpaCy 2: Natural language understanding with Bloom embeddings. *Convolutional Neural Netw. Increm. Parsing* **2017**, *7*, 411–420.
43. Sadvilkar, N.; Neumann, M. PySBD: Pragmatic Sentence Boundary Disambiguation. In Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), Association for Computational Linguistics, Online, November 2020; pp. 110–114. Available online: https://aclanthology.org/2020.nlposs-1.15/ (accessed on 15 January 2024).
44. Bird, S.; Loper, E.; Klein, E. *Natural Language Processing with Python*; O'Reilly Media Inc.: Sebastopol, CA, USA, 2009.
45. Sanampudi, S.K.; Kumari, G. Temporal Reasoning in Natural Language Processing: A Survey. *Int. J. Comput. Appl.* **2010**, *1*, 68–72. [CrossRef]
46. Pustejovsky, J.; Verhagen, M.; Sauri, R.; Littman, J.; Gaizauskas, R.; Katz, G.; Mani, I.; Knippen, R.; Setzer, A. *TimeBank 1.2 LDC2006T08. Web Download*; Linguistic Data Consortium: Philadelphia, PA, USA, 2006.
47. Moerchen, F. Temporal pattern mining in symbolic time point and time interval data. In Proceedings of the KDD'10: The 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Nashville, TN, USA, 30 March–2 April 2010; pp. 1–2.
48. Gopfert, J.; Kuckertz, P.; Weinand, J.M.; Kotzur, L.; Stolten, D. Measurement Extraction with Natural Language Processing: A Review. *Find. Assoc. Comput. Linguist. EMNLP* **2022**, *2022*, 2191–2215.
49. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *11*, 39–41. [CrossRef]
50. Altinok, D. *Mastering spaCy: An End-to-End Practical Guide to Implementing NLP Applications Using the Python Ecosystem*; Packt Publishing: Birmingham, UK, 2021.
51. Fang, X.; Zhan, J. Sentiment analysis using product review data. *J. Big Data* **2015**, *2*, 5. [CrossRef]
52. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; The MIT Press: Cambridge, MA, USA, 2012.
53. Doan, S.; Yang, E.W.; Tilak, S.S.; Li, P.W.; Zisook, D.S.; Torii, M. Extracting health-related causality from twitter messages using natural language processing. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 71–77. [CrossRef] [PubMed]
54. Li, Z.; Ding, X.; Liu, T.; Hu, J.E.; Van Durme, B. Guided Generation of Cause and Effect. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence IJCAI-20, Yokohama, Japan, 7–15 January 2020.
55. Hendrickx, I.; Kim, S.; Kozareva, Z.; Nakov, P.; Séaghdha, D.Ó.; Padó, S.; Pennacchiotti, M.; Romano, L.; Szpakowicz, S. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden, 15–16 July 2010; pp. 33–38.

56. Wang, J.; Dong, Y. Measurement of Text Similarity: A Survey. *Information* **2020**, *11*, 421. [CrossRef]
57. Gomaa, W.H.; Fahmy, A. A survey of text similarity approaches. *Int. J. Comput. Appl.* **2013**, *68*, 13–18.
58. Navigli, R.; Martelli, F. An Overview of Word and Sense Similarity. *Nat. Lang. Eng.* **2019**, *25*, 693–714. [CrossRef]
59. Li, Y.; McLean, D.; Bandar, Z.; O'Shea, J.; Crockett, K. Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 1138–1150. [CrossRef]
60. Li, Y.; Bandar, Z.; McLean, D. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.* **2003**, *15*, 871–882. [CrossRef]