



HAL
open science

Hybrid Data-Driven and Knowledge-Based Predictive Maintenance Framework in the Context of Industry 4.0

Fidma Mohamed Abdelillah, Hamour Nora, Ouchani Samir, Benslimane Sidi-Mohammed

► **To cite this version:**

Fidma Mohamed Abdelillah, Hamour Nora, Ouchani Samir, Benslimane Sidi-Mohammed. Hybrid Data-Driven and Knowledge-Based Predictive Maintenance Framework in the Context of Industry 4.0. MEDI 2023, Model and Data Engineering: 12th International Conference, Nov 2023, Sousse, Tunisia. pp.319-337, 10.1007/978-3-031-49333-1_23 . hal-04370795

HAL Id: hal-04370795

<https://hal.science/hal-04370795>

Submitted on 24 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hybrid Data-driven and Knowledge-based Predictive Maintenance Framework in the Context of Industry 4.0

Fidma Mohamed Abdelillah¹[0009-0006-7187-1534], Hamour Nora²[0009-0001-1786-103X], Ouchani Samir³[0000-0002-7997-8225], and Benslimane Sidi-Mohammed¹[0000-0002-7008-7434]

¹ CESI Lineact, Lyon, France , mafidma@cesi.fr

² CESI Lineact, Lyon, France , nhamour@cesi.fr

³ CESI Lineact, Aix-en-Provence, France , souchani@cesi.fr

⁴ LabRI-SBA Lab, Ecole Supérieure en Informatique, Sidi Bel Abbes, Algeria, s.benslimane@esi-sba.dz

Abstract. The emergence of Industry 4.0 has heralded notable progress in manufacturing processes, utilizing sophisticated sensing and data analytics technologies to maximize efficiency. A vital component within this model is predictive maintenance, which is instrumental in ensuring the dependability and readiness of production systems. Nonetheless, the heterogeneous characteristics of industrial data present obstacles in realizing effective maintenance decision-making and achieving interoperability among diverse manufacturing systems. This paper addresses these obstacles by introducing a hybrid approach that harnesses the power of ontologies, machine learning techniques, and data mining to identify and predict potential anomalies in manufacturing processes. Our work concentrates on designing an intelligent system with standardized knowledge representation and predictive capacities. By bridging the semantic divide and enhancing interoperability, ontologies enable the amalgamation of various manufacturing systems, thereby optimizing maintenance decision-making in real-time. As demonstrated in the experimental results, this approach not only ensures system reliability but also fosters a seamless, integrated, and efficient production landscape.

Keywords: Industry 4.0 · Industrial Cyber-Physical System · Predictive Maintenance · Chronicle Mining · Ontology · SWRL Rules.

1 Introduction

Smart factories in the Industry 4.0 era, have emerged as transformative manufacturing environments that harness cutting-edge technologies to revolutionize production processes. These smart factories leverage sophisticated sensing technologies and data analytics to gain real-time insights into their operations. They can optimize manufacturing processes by analyzing vast amounts of data, leading to higher production efficiency and reliability [5]. Artificial Intelligence (AI) techniques, such as machine learning and data mining, play a pivotal role in this context. These advanced AI methods enable the factories to identify patterns, trends, and anomalies within the data, helping them detect potential issues and predict maintenance needs in advance [7].

The complexity of industrial data introduces significant challenges in achieving seamless interoperability across manufacturing systems. The diverse nature of this data leads to the emergence of complex knowledge structures, creating what is known as a semantic gap issue. This gap impedes effective communication and sharing of information between different components and systems within the manufacturing environment. As a result, the full potential of data-driven decision-making and automation remains untapped [14]. Moreover, industrial systems, particularly Cyber-Physical Systems (CPS), operate in knowledge-intensive domains that demand uniform and standardized knowledge representation [18]. A cohesive and consistent approach to knowledge representation becomes indispensable to enable real-time reasoning and automated decision-making. Overcoming these hurdles is crucial to harnessing the power of Industry 4.0 technologies and fully embracing the potential of predictive maintenance in intelligent manufacturing systems.

Amidst the diverse data sources and industrial requirements, effective predictive maintenance in Industry 4.0 faces numerous challenges. The semantic gap issue and lack of uniform knowledge representation present obstacles to smooth interoperability and automated decision-making, ultimately limiting

the overall effectiveness of predictive maintenance systems [18]. Additionally, maintenance processes may encounter difficulties in detecting and addressing issues proactively. Hence, By focusing on bridging the semantic gap, ensuring uniform knowledge representation, and enabling real-time reasoning, we aim to develop a robust and efficient predictive maintenance framework that empowers industries to proactively tackle maintenance issues and optimize their manufacturing operations.

In this paper, we introduce a Hybrid data-driven and knowledge-based Predictive Maintenance for industry Systems (HPMS). HPMS employs a combination of statistical AI technologies such as machine learning and data mining, along with symbolic AI technologies. It utilizes logic rules generated from chronicle patterns and domain ontologies for ontology reasoning, as well as SQWRL queries for retrieving temporal information on failure points. This integrated approach aims to enable the automatic detection of machinery anomalies and accurate prediction of future events, leading to enhanced efficiency and effectiveness of predictive maintenance in various industrial settings. By harnessing HPMS, industries can achieve automated anomaly detection and precise event prediction, optimizing their maintenance practices and overall operational efficiency. The primary contributions of this paper are as follows.

1. Developing and implementing a Hybrid Predictive Maintenance System (HPMS) that merges both statistical and symbolic AI approaches, specifically leveraging machine learning and chronicle mining methodologies.
2. Employing Semantic Web Rule Language (SWRL) rules, which are derived from chronicle patterns and domain ontologies, to enhance its ontology reasoning capabilities.
3. Integrating Semantic Query-Enhanced Web Rule Language (SQWRL) queries to extract temporal information regarding failure points, thereby augmenting its real-time reasoning capabilities.
4. Upon detection and prediction of failures, clustering the failure points by taking into account the minimum time to failure. This facilitates effective and efficient prioritization of the said points.

The paper begins with a comprehensive literature review 2, exploring existing approaches in predictive maintenance and Industry 4.0, with a particular focus on data-driven and hybrid approaches. The proposed 'HPMS' framework, which integrates ontologies and data-driven techniques, is thoroughly explained in Section 3. The experimental setup is then outlined in Section 4, detailing the dataset and chosen machine learning algorithms for evaluation. Section 5 concludes with a summary of contributions and potential future research directions to improve 'HPMS' implementation.

2 Related Work

In the Industry 4.0 era, significant research efforts were dedicated to automating and improving smart manufacturing processes. AI-based methods have shown promising results, especially in the realm of predictive maintenance for Industry 4.0 tasks. By surveying the existing AI-based smart manufacturing and predictive maintenance approaches, this section categorized them into four distinct categories: (i) data-driven; (ii) physical model-based; (iii) knowledge-based; and (iv) hybrid model-based.

2.1 Data-driven approaches

In recent times, data-driven approaches have emerged as a significant solution for smart manufacturing and predictive maintenance. IWSNs, CPS, and Internet of Things (IoT) are employed together to collect and intelligently process big industrial data, aiding in decision-making. The exponential growth of data volume, coupled with the rapid advancement in data acquisition technologies, has heightened attention towards data-driven methods for the predictive maintenance of industrial equipment [23]. Sezer et al. [19] proposed a cost-effective CPS architecture for monitoring machining conditions, utilizing cloud-stored data and a Recursive Partitioning and Regression Tree model to predict part rejection while integrating time series analysis and machine learning. Zhang [24] proposed an online data-driven framework for bearing RUL ⁵ prediction using deep CNN. The method employed the Hilbert-Huang transform for

⁵ Remaining Useful Life: The length of time a machine is likely to operate before it requires repair or replacement.

preprocessing, a nonlinear degradation indicator for learning, and a support vector regression model for RUL prediction. Dual-task deep LSTM networks were introduced by Miao et al. [13] for aeroengine degradation assessment and RUL prediction simultaneously, yielding more reliable results tailored to each aeroengine’s health state. Also, De Luca et al. [6] presented a DL-based approach for PdM tasks. They utilized a highly efficient architecture with a multi-head attention (MHA) mechanism, achieving superior results in terms of RUL estimation while maintaining a compact model size. Experimental results on the NASA dataset demonstrated the approach’s effectiveness and efficiency.

2.2 Physical model-based approaches

Physical models are models that utilize physical laws, often from first principles, to quantitatively characterize the behavior of a failure mode. Such models typically employ mathematical representations of the physical behavior of a machine’s degradation process to calculate the RUL of the machinery. The mathematical representation captures how the monitored system responds to stress on both the macroscopic and microscopic levels [20]. In Heyns et al. [8], Gaussian mixture models (GMMs) are used to detect faults in vibration signals, especially indicating potential gear damage. The method computes the negative log-likelihood (NLL) of signal segments, measuring their deviation from a healthy gearbox’s reference density distribution. By synchronously averaging the NLL discrepancy signal, a clear representation emerges, offering insights into the gear damage’s nature and severity. Also, Tiwari et al. [21] introduced Gaussian Process Regression (GPR) for tracking bearing features that incorporate uncertainty in predictions and is used to evaluate and predict. Three GPR models with different covariance functions are explored for feature tracking and RUL assessment. In another study by Wu et al. [22], the primary focus centers on predicting RUL of lithium-ion batteries, enhancing the dependability and safety of battery-powered systems. This research employs an empirical degradation model alongside the particle filter (PF) algorithm to facilitate real-time parameter updates.

2.3 Knowledge-based approaches

A system based on knowledge uses a knowledge base to store a computational model’s symbols in the form of domain statements and performs reasoning by manipulating these symbols. These systems determine appropriate decisions by measuring the similarity between a new observation and a databank of previously described situations. Knowledge-based approaches are categorized into **knowledge graphs and ontologies**, **rule-based systems**, and **fuzzy systems**.

In a rule-based system, knowledge is represented using "IF-THEN" rules, comprising a knowledge base housing rules, a facts base storing inputs, and an inference engine applying rules to the facts base to derive new insights [1]. The term "knowledge graph" is often synonymous with "**ontology**", which is explicitly defined as "a precise description or identification of conceptualization for a domain of interest" [1]. Ontologies necessitate formal logic to distinctly define concepts and relationships [1]. Fuzzy-knowledge-based models employ fuzzy logic, akin to rule-based systems, using IF-THEN rules. Fuzzy logic manages partial truth values between true and false, quantifying the degree of truth or falsity and resonating with human perceptions.

Konys [11] offered structured guidance for knowledge management-based sustainability assessment, advocating ontology as a means of conceptualizing knowledge and enhancing its accessibility and reusability. Mehdi et al. [12] introduced "sigRL", a semantic rule language, facilitating efficient diagnostic programming with a high-level, data-independent vocabulary, diagnostic ontologies, and queries. They presented the diagnostic system "SemDia", showcasing sigRL’s role in effective and efficient diagnostics.

2.4 Hybrid model-based approaches

Hybrid model-based approaches involve combining the strengths of different methodologies, such as physical models, data-driven techniques, and knowledge-based systems. These approaches seek to leverage the benefits of each individual method to enhance predictive accuracy and flexibility. A hybrid model-based

predictive maintenance task can be classified into two main approaches: The series hybrid model and the parallel hybrid model [26].

Zhou et al. [25] devised a pioneering feature-engineering-based machine learning approach that operates on real production data. This technique delves into sequences of welding instances obtained from manufacturing lines, amalgamating engineering insights and data science methodologies. By employing sophisticated feature engineering tactics bolstered by domain knowledge, this approach captures intricate dependencies across welding sequences, filling a gap in the existing literature. Qiusi [17] introduced an innovative fusion of evidential theory tools and semantic technologies to bolster predictive maintenance. They harnessed the Evidential C-means (ECM) algorithm, among other tools, to assess failure criticality by considering time constraints and maintenance costs. This method simultaneously leveraged domain ontologies and rule-based extensions to formalize domain knowledge, enabling precise forecasts of future failures' timing and criticality.

Klein [10] incorporated expert knowledge of class or failure mode-dependent attributes into a specialized neural network framework. Accompanied by an attribute-wise encoding approach based on 2D convolutions, this strategy facilitated the sharing of knowledge through the utilization of filters among analogous data streams. Furthermore, Cao et al. [4] presented an ontology-driven strategy to enhance predictive maintenance, combining fuzzy clustering techniques with semantic technologies. This approach analyzed historical machine data to understand the significance of failures. Utilizing semantic technologies for analyzing fuzzy clustering outputs, it achieved precise predictions of failure timing and criticality. The outcome was a domain-specific ontology enriched with predictive maintenance knowledge, supported by a set of SWRL predictive rules for informed assessments of machinery failure timing and impact.

2.5 Learned lessons

This literature review provides a comprehensive analysis of existing approaches for predictive maintenance in Industry 4.0, identifying the strengths and weaknesses of different methods and offering insights into their suitability and effectiveness in various industrial contexts. The study emphasizes that there is no one-size-fits-all solution for predictive maintenance in Industry 4.0. The primary objective of this state-of-the-art review was to address the gap in the existing literature on predictive maintenance approaches.

With respect to the studied state-of-the-art, we present a novel data-driven Knowledge-based system for Predictive Maintenance in Industry 4.0 (HPMS). HPMS moves beyond the limitations of the aforementioned approaches by combining the power of data-driven methods with a uniform representation of knowledge. This integration enhances predictive maintenance capabilities by leveraging statistical AI technologies like machine learning and chronicle mining alongside symbolic AI techniques.

3 HPMS Framework

Our developed HPMS framework strategically amalgamates statistical and symbolic AI techniques to devise a hybrid predictive maintenance mechanism. It incorporates machine learning, chronicle mining, domain ontologies, and logic rules, effectively capitalizing on both *data-driven* and *knowledge-driven* approaches. HPMS employs Semantic Web Rule Language (SWRL) rules, which are engendered from chronicle patterns and ontology reasoning, to autonomously detect anomalies in machinery and prognosticate future events. Figure 1 delineates the HPMS framework, showcasing its five key processing steps devised to augment the predictive maintenance capabilities of an industrial system.

3.1 Failure Chronicle mining

In industrial environments, maintenance data is typically represented as timestamped sequences. To identify frequently occurring patterns in such data, Sequential Pattern Mining (SPM) is used as an instrumental technique. Initially explored for analyzing customer purchase behavior, SPM targets the discovery of sequential patterns with support exceeding a predetermined threshold.

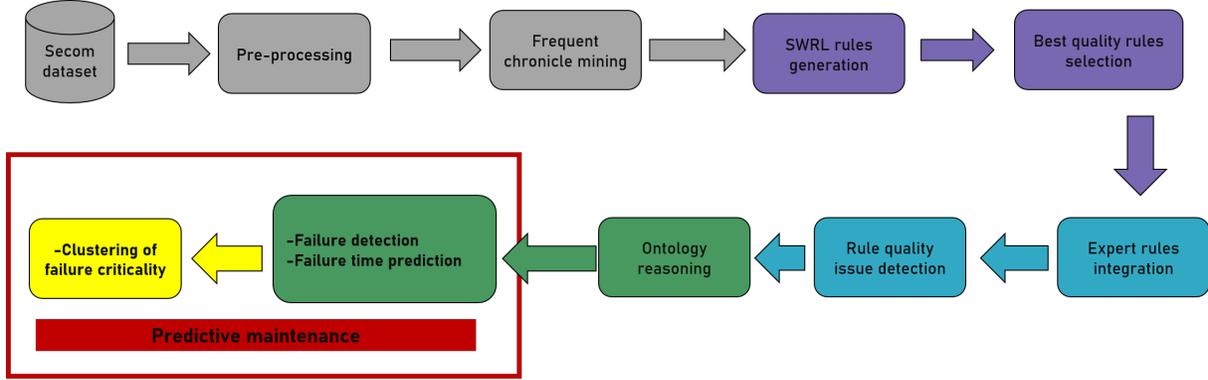


Fig. 1: An overview of the proposed HPMS framework.

However, traditional SPM algorithms fall short in capturing time intervals between elements and items. In the realm of predictive maintenance, the emphasis is on temporal patterns, specifically chronicles. Chronicles are unique sequential patterns that represent event sequences with associated time intervals (temporal constraints). In the predictive maintenance domain, failure chronicles are created to capture the temporal patterns preceding machinery failures. These play a pivotal role in pre-empting machine anomalies and amalgamating frequent chronicle mining with semantic approaches to enhance predictive maintenance tasks. Definition 1 describes formally a chronicle.

Definition 1 (Chronicle). A chronicle is a pair $C = (E, T)$ such that:

1. $E = \{e_1, \dots, e_n\}$ is a set of events, where $\forall e_i \in E$ we have $e_i \prec_E e_{i+1}$ such that \prec_E is pre-order relation that defines the sequence orders of the events in E .
2. $T = \{t_{ij} : 1 \leq i < j \leq |E|\}$ is a set of temporal constraints on E such that for all pairs (i, j) satisfying $i < j$, t_{ij} is denoted by $e_i[t_{-ij}, t_{+ij}]e_j$. A pair of events (e_1, t_1) and (e_2, t_2) are said to satisfy the temporal constraint $e_1[t_-, t_+]e_2$ if $t_2 - t_1 \in [t_-, t_+]$.

3.2 Predictive rule generation

In this part, we focus on SWRL rule generation, a crucial step in HPMS for rule-based reasoning in predictive maintenance. Following rule generation, we rigorously prune to remove low-quality rules, retaining only relevant and accurate ones, thus improving the system effectiveness.

3.2.1 SWRL rules generation During this stage, a set of predictive SWRL rules are created based on the extracted frequent failure chronicles. These rules constitute logical IF-THEN statements facilitating rule-based reasoning. Algorithm 1 automatically transforms the extracted chronicles into a set of SWRL predictive rules by extracting distinct event types from a failure chronicle, determining their order and temporal constraints, and subsequently formulating rules as implications between the antecedent and consequent sections. Algorithm 1 extracts the last non-failure events before the failure in the chronicle (lines: 1-2) and processes each time interval to obtain atoms representing the interval, preceding event (lines: 3-4), and subsequent event (line: 5). These atoms form the antecedent part of the SWRL rule (lines: 6-7). It then extracts the time constraint between the last event and the failure event, creating another atom representing this constraint (lines: 9-10). These atoms form the consequent part of the SWRL rule (lines: 11-12). By creating an implication between the antecedent and consequent (lines: 14-15), the algorithm generates a predictive SWRL rule (R) for reasoning and prediction in the context of the failure chronicle.

Algorithm 1 Transformation of a chronicle into a predictive SWRL rule.

Input: $\mathcal{F} = (\Sigma, \tau)$: A failure chronicle model wherein the last event type signifies a failure event, Σ designates its episode, and τ denotes its temporal constraints.

Output: R : A predictive SWRL rule

```

1:  $EL \leftarrow \text{LastNonfailureEvent}(\mathcal{F})$   $\triangleright$  Extraction of the last non-failure events preceding the failure within a chronicle.
2:  $R \leftarrow \emptyset, A \leftarrow \emptyset, C \leftarrow \emptyset$ 
3: for each  $e_i[t_{ij}, t_{ij}]e_j \in \tau$  do
4:    $pe \leftarrow \text{PrecedingEvent}(e_i[t_{ij}, t_{ij}]e_j)$   $\triangleright$  Extraction of the preceding event of this time interval.
5:    $se \leftarrow \text{SubsequentEvent}(e_i[t_{ij}, t_{ij}]e_j)$   $\triangleright$  Extraction of the subsequent event of this interval.
6:    $AtomA \leftarrow [t_{ij}, t_{ij}] \wedge pe \wedge se$ 
7:    $A \leftarrow AtomA \wedge A$ 
8: end for
9: for each  $el \in EL$  do
10:   $ftc \leftarrow \text{FailureTimeConstraint}(el, \mathcal{F})$   $\triangleright$  Extraction of the time constraint between the last event prior to the failure and the failure event.
11:   $AtomC \leftarrow el \wedge ftc$ 
12:   $C \leftarrow AtomC \wedge C$ 
13: end for
14:  $R \leftarrow (A \rightarrow C)$   $\triangleright$  Generation of rule  $R$  as an implication between the antecedent and consequent.
15: Return  $R$ 

```

3.2.2 Best quality rules selection Upon completing the rule generation process, real-world data often contains imprecision, and rule-based classification and prediction can suffer from overfitting issues, leading to rules of low quality [3]. we conduct a subsequent rule pruning step to eliminate these rules. To achieve this, we adopt a multi-objective optimization strategy that considers two crucial measures: rule accuracy and rule coverage.

By analyzing the correlation between each decision rule (R) and the target class (F) through a contingency table, which tabulates data categories and their frequencies defined in Table 1.

The accuracy is defined as $\text{Accuracy}(R) = \frac{n_{af}}{n_a}$ and

the rule coverage is defined by $\text{Coverage}(R) = \frac{n_{af}}{n_f}$ where

n_{af} denotes examples where both the antecedent and consequent of the rule are true, while $n_{a\bar{f}}$ represents cases where the antecedent is true, but the consequent is not. Similarly, $n_{\bar{a}f}$ signifies scenarios where the antecedent is false, but the consequent is true, and $n_{\bar{a}\bar{f}}$ denotes instances where neither the antecedent nor the consequent holds. These derived accuracy and coverage metrics serve as evaluative indicators of the overall rule quality.

	F is true	F is not true	
A is true	n_{af}	$n_{a\bar{f}}$	n_a
A is not true	$n_{\bar{a}f}$	$n_{\bar{a}\bar{f}}$	$n_{\bar{a}}$
	n_f	$n_{\bar{f}}$	N

Table 1: Contingency table for evaluation.

In our multi-objective optimization process, we employ the fast non-dominated sorting algorithm, also known as the Pareto sorting algorithm [15]. This algorithm efficiently selects high-quality rules by prioritizing accuracy and coverage as key objectives, and effectively capturing the subset with the best balance of these measures.

3.3 Expert rule integration

The process of expert rule integration involves incorporating domain experts' knowledge by integrating expert rules to supplement the existing chronicle rule base. This integration aims to address the inherent incompleteness of the chronicle rule base. During this process, rule quality issues such as *redundancy*, *conflict*, and *subsumption* are examined to ensure effective reasoning performance when combining diverse rules. To automatically detect rule quality measures, Algorithm 2 was proposed that includes functions for extracting atom sets from expert rules and subsequent steps for identifying issues with the chronicle rule base. The algorithm starts by initializing an updated rule base (C') with the existing chronicle rule base

(C). It then iterates through each rule (R) in C to identify potential issues with an expert rule (Re) (1-2). If ChroRedundancy(Re, R) is true, indicating redundancy, the algorithm prints a ChroRedundancy message and retains the original rule base (C') (3-4). If ChroSubsumes(Re, R) is true, indicating subsumption, it removes R from C' and integrates Re (6-8). Similarly, if ChroConflict(Re, R) is true, indicating a conflict, it removes R from C' and integrates Re. After evaluating all rules, the algorithm returns the updated rule base (C') with the integrated expert rule. Thus, HPMS integrates an expert rule into a chronicle rule base while ensuring that the resulting rule base remains consistent and effective in capturing knowledge for predictive maintenance tasks.

Algorithm 2 Enhancing Detecting Issues in Expert Rule Integration within the Chronicle Rule Base.

Input: A chronicle rule base \mathcal{C} which contains a set of failure chronicles, and an expert rule R_e which is in the form of a failure chronicle.
Output: \mathcal{C}' : The integrated rule base.

- 1: $\mathcal{C}' \leftarrow \mathcal{C}$.
- 2: **for** each $R \in \mathcal{C}$ **do**
- 3: **if** ChroRedundancy(R_e, R) **then** ▷ Rule redundancy issue detected.
- 4: ▷ A ChroRedundancy message is printed, no change in the rule base.
- 5: **end if**
- 6: **if** ChroSubsumes(R_e, R) **then** ▷ Rule subsumption issue detected.
- 7: Remove(R, \mathcal{C}') ▷ Remove the subsumed chronicle rule from the rule base.
- 8: Integrate(R_e, \mathcal{C}') ▷ Integrate the expert rule into the rule base.
- 9: **end if**
- 10: **if** ChroConflict(R_e, R) **then** ▷ Rule conflict issue detected.
- 11: Remove(R, \mathcal{C}') ▷ Remove the conflict chronicle rule from the rule base.
- 12: Integrate(R_e, \mathcal{C}') ▷ Integrate the expert rule into the rule base.
- 13: **end if**
- 14: **end for**
- 15: **return** \mathcal{C}'

3.4 Failure detection and prediction

HPMS incorporates the failure detection and prediction process which leverages ontology reasoning for detecting anomalies and potential failures in the manufacturing environment. To enable this, HPMS utilizes the Manufacturing Predictive Maintenance Ontology (MPMO) as the domain ontology which encapsulates domain-specific concepts and relationships, providing a formal representation of key elements in the designed manufacturing environment, combined with generated SWRL rules to perform reasoning tasks. The SWRL rules act as a set of logical implications that define relationships and dependencies between different elements in the manufacturing process. These rules are created based on domain-specific knowledge and expertise. They describe how events, conditions, and patterns observed in the data can lead to potential failures or anomalies.

In the domain of intelligent systems, ontologies play a vital role in encapsulating domain knowledge. Within HPMS, the Manufacturing Predictive Maintenance Ontology (MPMO) [2] has been developed to define concepts and relationships within chronicles. The definitions of key concepts and relationships in the MPMO ontology have been formalized based on fundamental notions discussed in Section 3.1. Figure 2 illustrates the main modules in the MPMO ontology. To enhance reusability, we have employed the ontology modularization method during development, resulting in three small, reusable modules: the *Condition Monitoring* Module, the *Manufacturing* Module, and the *Context* Module.

When executing the reasoning process, HPMS employs a reasoner capable of interpreting and applying the SWRL rules to the MPMO ontology. The reasoner performs semantic-based inference, matching the facts and rules in the ontology to derive new knowledge to semantic-based inference [9]. By applying ontology reasoning, HPMS is capable of effectively detecting machinery failures. Furthermore, HPMS

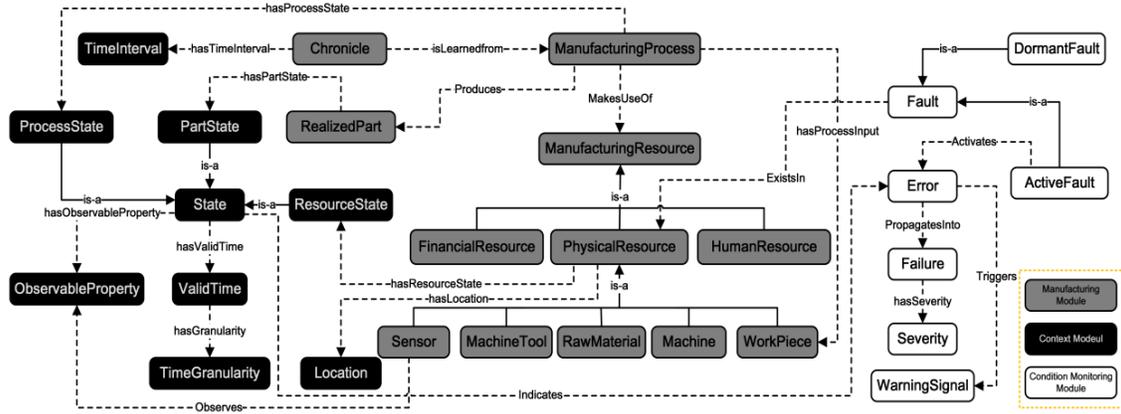


Fig. 2: Main modules in MPMO ontology [3].

extends beyond simple detection by utilizing SWRLAPI⁶ to retrieve temporal information about failure points thanks to the chronicle format, SQWRL queries are generated and executed, enabling HPMS to predict the timing of these failures. This predictive capability facilitates predictive maintenance planning and intervention to mitigate the impact of potential failures. By leveraging the knowledge and inference capabilities provided by MPMO and the SWRL rules, HPMS achieves accurate and timely identification of machinery failures. The predictive aspect of HPMS adds an extra layer of foresight, enabling maintenance teams to take proactive measures to prevent or minimize the impact of future failures.

3.5 Failure criticality prediction

The final step in our predictive maintenance approach is the *failure criticality prediction*, a crucial process for prioritizing maintenance actions. After detecting the failure points and predicting their temporal information, our goal is to cluster the predicted failure points based on their minimum time to failure. The latter serves as a critical indicator of the urgency and severity of each potential failure. Through this clustering process, we can efficiently allocate resources and schedule maintenance activities. To achieve this, we apply the K-means clustering algorithm, which groups the failure points into three levels based on their minimum time to failure. By analyzing similarities and dissimilarities between the failure points, the algorithm assigns them to different clusters, allowing the identification of patterns and relationships among the failures. Once the failure points are clustered, we assign criticality levels to each cluster based on the urgency and potential impact of the failures. Clusters with shorter minimum time to failure and higher potential consequences are deemed more critical, necessitating immediate attention and intervention. Following clustering, the criticality of each failure group is determined based on the assigned clusters and their characteristics, enabling effective prioritization of maintenance efforts.

The failure criticality prediction process provides valuable insights into the maintenance prioritization strategy. Furthermore, the clustering results can be visualized through graphical representations to facilitate decision-making and enhance situational awareness. Thus, maintenance teams can easily identify high-risk clusters and allocate resources accordingly, ensuring optimal utilization of resources and minimizing the impact of failures on production and safety. Consequently, HPMS leverages a powerful combination of ontology reasoning, SWRL rules, and data-driven approaches to effectively detect potential failures in the manufacturing environment. The process involves the extraction of frequent failure chronicles, the transformation of these chronicles into SWRL predictive rules, and the application of rule pruning using a multi-objective optimization approach. This results in a refined set of high-quality

⁶ SWRLAPI is a standalone SWRL API-based application.

rules for failure prediction. Furthermore, the failure criticality prediction step ensures the prioritization of maintenance actions, enabling efficient resource allocation and mitigation of potential failures' impact.

4 Implementation & Experimental Results

In this section, we showcase experimental results from HPMS. We outline the settings, detail the dataset characteristics, and provide illustrative figures depicting HPMS steps and outcomes.

4.1 Data Preparation

In our settings, we have used UCI SECOM ⁷ dataset comprises 1567 data records with 590 attributes, representing test points, each consisting of 589 numerical attributes and a timestamp indicating the recording time. The dataset contains a binary label for pass/fail status, with 104 instances of production failures out of 1567 records. The data is stored in a raw text file, with timestamps allowing temporal analysis of events. The SECOM dataset was chosen due to its temporal aspect, facilitating the study of predictive maintenance systems and anomaly detection over time. Being derived from real-world industrial scenarios, the dataset presents complex and diverse patterns that reflect challenges encountered in actual industrial environments, enhancing the value of the experiments. To ensure the accuracy and efficiency of our failure prediction task, we perform essential preprocessing steps on the UCI SECOM dataset.

- **Feature selection:** Not all the data in the dataset is relevant for this specific task, as some may contain noise or be redundant. To address this challenge in high-dimensional data, we employ a feature selection technique to identify and retain only the 10 relevant subsets of features.
- **Discretization:** After the feature selection step, the dataset undergoes data preprocessing to transform continuous variables into a discrete representation. This is achieved through data discretization. Discretization involves dividing continuous variables into a finite set of intervals, where each interval is associated with a specific range of data values. In our case, we discretize the dataset by creating 20 bins for each variable, representing discrete numeric intervals. Integer values from 1 to 20 are used to represent these intervals (Listing 1.1). This discretization step enables us to handle the data effectively and prepare it for subsequent analysis.

```
1 2 4 5 7 9 11 13 15 17 19,1 3 5 7 9 11 13 15 17 20,2 4 5 7 9 11 14 15 18 19,
```

Listing 1.1: the nominal attributes obtained from the discretization step

- **Sequentialization:** Following the data discretization, the sequentialization process organizes the data in the form of (event-timestamp) pairs. Within each data sequence, the last event represents a failure, while the events preceding it are considered 'normal'. The first given sequence (Listing 1.2) starts with events 2, 4, 5, 7, 10, 11, 13, 15, 17, 19, each followed by a timestamp denoted by "<timestamp>". These events are separated by "-1" from the next event in the sequence. After the last event "19" and its corresponding timestamp, we encounter the pattern "-1 -1 -1 -1", which indicates the end of this failure sequence. By extracting frequent failure sequences, we can identify patterns leading up to failures and gain insights into failure prediction.

```
1 2 4 5 7 10 11 13 15 17 19 <1199911860> -1 2 4 5 7 10 11 13 15 17 19
   <1199916300> -1 2 4 5 7 10 11 13 15 17 19 <1199919900> -1 2 4 5 7 10 11
   13 15 17 19 <1199920740> -1 2 4 5 7 10 11 13 15 17 19 <1199921040> -1 2 4
   5 7 10 11 13 15 17 19 <1199921280> -1 2 4 5 7 9 11 13 15 17 19
   <1199944560> -1 -1 -1 -1
```

Listing 1.2: an extracted failure sequence.

With the completion of these preprocessing steps, we have prepared the UCI SECOM dataset for chronicle mining and the development of HPMS.

⁷ The UCI SECOM (Semiconductor Manufacturing) [dataset](#) consists of manufacturing operation data and the semiconductor quality data.

4.2 Chronicle Mining

After preprocessing the UCI SECOM dataset, we employ the **Clasp-CPM** algorithm⁸ to obtain a set of **frequent failure chronicles** to describe failure events and their temporal constraints. This involves analyzing historical data sequences from manufacturing processes to identify sequential patterns (**SP**) of failures and their temporal relationships. By mining these chronicles, we extract common event sequences that precede failures. In a failure chronicle, the last node represents a failure event. These frequent failure chronicles will be transformed into SWRL rules for failure prediction. Figure 3 illustrates a graph-based representation of a chronicle. The integers associated with each node are nominal attributes from the discretization step. Those integers (e.g., 4 5 7 11 13 19 in the upper episode) collectively form the description for an event. An event with an integer value of 0 indicates a failure, which is typically the last event within a chronicle, while edges represent time intervals between events.

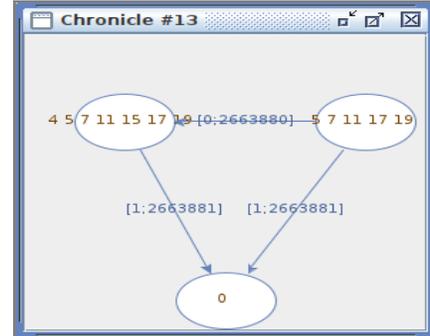


Fig. 3: The graphical representation of a failure chronicle.

4.3 SWRL rules generation

To generate SWRL predictive rules for ontology reasoning, we utilize the set of frequent failure chronicles obtained from the previous step that will be transformed into rules using Algorithm 1. By analyzing failure chronicle data, the algorithm identifies unique events contributing to failure occurrences and represents them as distinct event types. This enables us to understand the sequence and temporal constraints among these events. As a result of this process, a set of SWRL rules is formulated, which act as logical implications between the antecedent (body), describing normal events and their temporal constraints, and the consequent (head), containing information about the temporal constraints between normal events and the failure event parts. In this rule (Listing 1.3), the antecedent (body) of the SWRL rule includes various conditions represented by the statements **ManufacturingProcess(?s)**, **hasEvent(?s, ?e1)**, and **hasItem(?e1,2)**, etc. These conditions specify the normal events and their corresponding items, which are associated with the ManufacturingProcess represented by ?s. The consequent (head) of the rule, represented by **hasMinF(?e1, 1)** and **hasMaxF(?e1, 2656921)**, provides information about the temporal constraints between normal events and the failure event, indicated by ?e1. The **hasMinF** and **hasMaxF** relations are used to define the minimum and maximum failure values associated with the specific event ?e1.

```

1 ManufacturingProcess(?s) ^ hasEvent(?s, ?e1) ^ hasItem(?e1,2) ^ hasItem(?e1,4)
  ^ hasItem(?e1,5) ^ hasItem(?e1,7) ^ hasItem(?e1,10) ^ hasItem(?e1,19) ->
  hasMinF(?e1, 1) ^ hasMaxF(?e1, 2656921)

```

Listing 1.3: SWRL rule generated from a failure chronicle

4.4 Best quality rules selection

In HPMS, we have implemented a rule pruning module to carefully select a subset of high-quality rules from the chronicle rule base. First, we calculate the average values of these two quality measures across various levels of chronicle support⁹ which allows us to examine the correlation between them. Then, we employ the fast non-dominated sorting algorithm for multi-objective optimization, which ranks rules

⁸ ClaSP is used for discovering closed sequential patterns in sequence databases. It can be run using SPMF – an open-source software and data mining library – <https://www.philippe-fournier-viger.com/spmf/>

⁹ An occurrence of a chronicle \mathcal{C} in a sequence \mathcal{S} is a subsequence of \mathcal{S} that satisfies all temporal constraints in \mathcal{C} . The support of a chronicle \mathcal{C} in a set \mathcal{D} of sequences is the number of sequences in \mathcal{D} in which \mathcal{C} occurs.

based on Pareto dominance. By considering the rules within the first Pareto Front, we have selected a subset of three high-quality rules that strike a balance between Accuracy and Coverage measures. These selected rules are retained for rule-based reasoning in our chronicle rule base, as shown in Fig. 4a. To evaluate the performance of each rule, we define a fitness function as the product of two crucial measures: **Accuracy** and **Coverage** (Fig. 4b). These measures play a vital role in assessing the quality of SWRL rules. By pruning out less robust rules, we ensure that HPMS makes more accurate and efficient predictions for better decision-making in industrial environments.

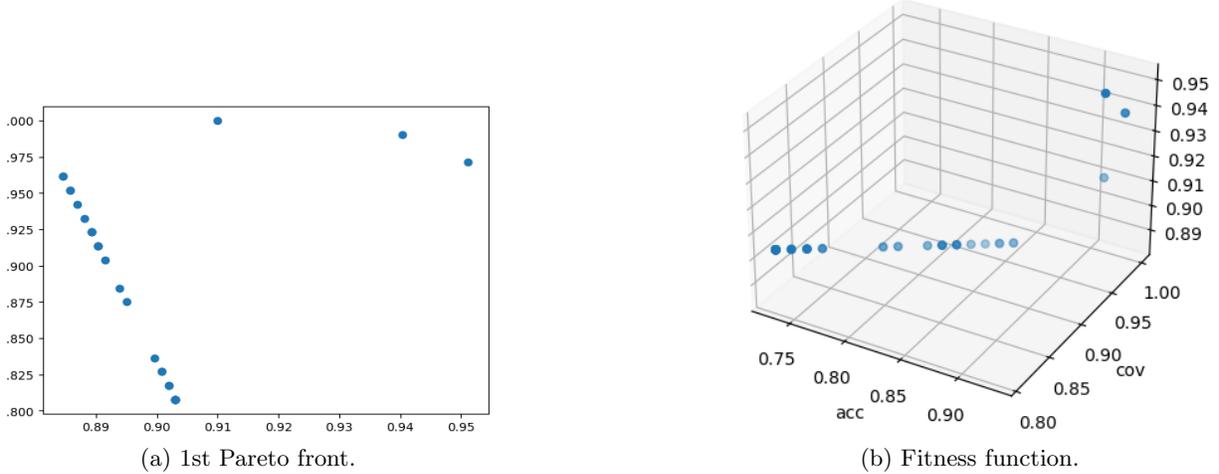


Fig. 4: Fast non-dominated algorithm output visualization.

4.5 Expert rules insertion

After obtaining a set of high-quality rules, the next phase involves integrating expert rules to complement the chronicle rule base. The purpose of this step is to enhance the overall fitness of the rule base and improve the performance of failure prediction.

The expert rules, stored separately from the chronicle rule base, can be imported into HPMS by assigning it to the expert profile. HPMS also detects potential quality issues with the rules, which are then displayed at the lower part of the interface. Expert rules have a similar format to the chronicle rules but may differ in rule atoms within the antecedent or consequent parts (Listing 1.4).

```

1 Expert rule 1:
2 ManufacturingProcess(?s) ^ hasEvent(?s, ?e1) ^ hasItem(?e1,2) ^ hasItem(?e1,4)
  ^ hasItem(?e1,5) ^ hasItem(?e1,7) ^ hasItem(?e1,11) ^ hasItem(?e1,17) ^
  hasItem(?e1,14569)-> hasMinF(?e1, 1) ^ hasMaxF(?e1, 2648721)
    
```

Listing 1.4: an expert rule with different atoms

After obtaining a set of best-quality rules, users can proceed to the expert rule integration phase by either opening the expert rule file and pushing the stored expert rules into the system or by directly entering the rules. However, if any quality issues arise, HPMS automatically performs appropriate actions based on the decision-making process outlined in Algorithm 2. For instance, if a conflict is detected between the integrated expert rule and the chronicle rule base, HPMS automatically removes the conflicting chronicle rule and adds the expert rule to the rule base as shown in Figure 5. Since the fitness values of expert rules are assigned as 1, they are considered to have higher quality than chronicle rules. Therefore, in case of any issues, HPMS always **prioritizes the expert rule over the chronicle rule**.

This expert rule integration phase allows users to enhance the rule base's overall fitness by incorporating domain-specific knowledge and complementing the extracted chronicle rules. By integrating expert rules alongside the chronicle rule base, HPMS achieves better performance in terms of failure prediction.

```
==>Oops,Input expert rule has conflict with the chronicle rule with index 4,
chronicle rule 4 has been removed from the rule base,
the expert rule has been added to the rule base.
```

Fig. 5: the output of the HPMS in case of conflict.

4.6 Failure detection and Time prediction

After integrating expert rules, the next phase involves failure detection. For this purpose, we utilize the Drools rule engine [16] to perform ontology reasoning on the individuals present in the MPMO ontology. MPMO serves as the domain ontology, and the Drools rule engine efficiently processes the rules and populated individuals, allowing for effective failure detection. Once the ontology reasoning is completed, SQWRL queries are generated to retrieve the prediction results by processing the antecedent of SWRL rules as pattern specifications for queries (Listing 1.5). The SQWRL queries aim to retrieve the minimum time and maximum time related to failures, resulting in a set of prediction results.

We obtained 27 rows of prediction results as shown in Fig. 6a. SWRLTAB table contains 4 columns indicating the time span between a specific event and a future failure. After the initial round of prediction, we further enhance the rule base by integrating additional expert rules. The expanded rule base, comprising 13 rules, is used for the second round of prediction. Utilizing this updated rule base, we conduct ontology reasoning and obtain 41 rows, each indicating the temporal information of a specific failure from the SQWRL query. This iterative process of rule base enhancement and ontology reasoning leads to progressive updates, enabling HPMS to detect and predict more potential failures and their respective time of occurrence. Thus, empowering HPMS to provide more comprehensive failure detection and prediction capabilities in manufacturing processes.

```
1 hasEvent(?mp, ?e) ^ hasMaxF(?e, ?xf) ^ sf:ManufacturingProcess(?mp) ^ hasMinF(?e
, ?nf) -> sqwrl:select(?mp, ?e, ?nf, ?xf) ^ sqwrl:columnNames("
ManufacturingProcess", "Event", "MinTimeToFail", "MaxTimeToFail")
```

Listing 1.5: the generated SQWRL query

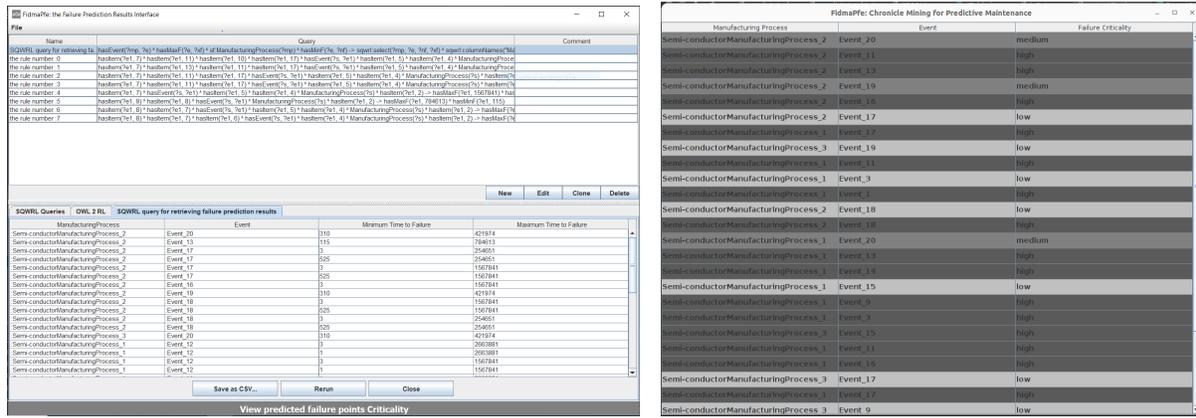
4.7 Failure criticality

In HPMS, we incorporated the K-means clustering algorithm using the Weka library to perform failure criticality analysis based on the minimum time to failures. By setting the value of k to 3, representing three clusters (high, medium, and low), we aimed to categorize the failure points into different levels of criticality. After applying the K-means algorithm to the obtained 27 rows of prediction results, we successfully clustered the data into three distinct groups representing different levels of failure criticality. To present these predictions in a clear and accessible manner, we have organized them in a table format with three columns: "Manufacturing Process," "Events," and "Criticality." Each row in the table corresponds to a specific predicted failure point, and the level of criticality for each point is indicated by the color assigned to the row. To represent the varying levels of criticality, we adopted a color-coding scheme, using light gray to represent low criticality, medium gray to represent medium criticality, and dark gray to represent high criticality as shown in Fig. 6b. By using the K-means clustering technique and visualizing the results in a table with color-coded criticality levels, we were able to effectively assess the level of danger associated with each failure point. This analysis provides valuable insights into the criticality of failures and helps prioritize necessary actions or interventions based on their potential impact.

As seen, we believe that the experimental results of our failure criticality process have demonstrated the effectiveness and usefulness of our predictive maintenance framework, HPMS.

5 Conclusion

In this work, we introduced a **hybrid predictive maintenance system** –HPMS– employing ontologies and machine learning within Industry 4.0, to navigate issues arising from industrial data heterogeneity



(a) SWRLTAB table contains prediction results

(b) Failures criticality frame

Fig. 6: Failure detection and Time prediction

and semantic gaps. HPMS utilizes ontologies for standardized knowledge representation and machine learning for failure prediction, significantly enhancing production efficiency. The efficacy of this approach was affirmed through the implementation of a tool and rigorous experimental scenarios. Encouraging outcomes from HPMS underscore the potential for predictive maintenance advancements and future research avenues.

The future work will center on several key areas. First, the investigation of stream reasoning techniques will address real-time data handling, enhancing the system’s ability to process and reason over dynamic data streams. The development of adaptive ontologies and rule bases is also crucial to accommodate evolving knowledge and context, ensuring the system’s agility and accuracy. Furthermore, efforts will be directed toward enhancing predictive models, specifically focusing on refining failure detection and prediction mechanisms for improved performance. Additionally, the exploration of decentralized-based deployment will allow for efficient scaling and resource utilization, leveraging the advantages of the blockchain platforms. Finally, the creation of predictive maintenance visualization tools coupled with the digital twins paradigm will provide intuitive insights, aiding in comprehending and acting upon the generated predictions effectively.

Bibliography

- [1] Ayadi, A.: Semantic approaches for the meta-optimization of complex biomolecular networks. Ph.D. thesis, Université de Strasbourg; Institut supérieur de gestion (Tunis) (2018)
- [2] Cao, et al.: Smart condition monitoring for industry 4.0 manufacturing processes: An ontology-based approach. *Cybernetics and Systems* **50**, 1–15 (02 2019)
- [3] Cao, et al.: Combining evidential clustering and ontology reasoning for failure prediction in predictive maintenance. In: *ICAART* (2), pp. 618–625 (2020)
- [4] Cao, Q., Samet, A., Zanni-Merk, C., de Bertrand de Beuvron, F., Reich, C.: An ontology-based approach for failure classification in predictive maintenance using fuzzy c-means and swrl rules. *Procedia Computer Science* **159**, 630–639 (2019)
- [5] Dalenogare, L.S., Benitez, G.B., Ayala, N.F., Frank, A.G.: The expected contribution of industry 4.0 technologies for industrial performance. *International Journal of Production Economics* **204**, 383–394 (2018), ISSN 0925-5273
- [6] De Luca, R., Ferraro, A., Galli, A., Gallo, M., Moscato, V., Sperli, G.: A deep attention based approach for predictive maintenance applications in iot scenarios. *Journal of Manufacturing Technology Management* **34**(4), 535–556 (2023)

- [7] Frank: Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics* **210**, 15–26 (2019)
- [8] Heyns, et al.: Combining synchronous averaging with a gaussian mixture model novelty detection scheme for vibration-based condition monitoring of a gearbox. *Mechanical Systems and Signal Processing* **32**, 200–215 (2012)
- [9] Khamparia, A., Pandey, B.: Comprehensive analysis of semantic web reasoners and tools: a survey. *Education and Information Technologies* **22** (11 2017)
- [10] Klein, a.o.: Enhancing siamese neural networks through expert knowledge for predictive maintenance. In: *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning*, pp. 77–92, Springer (2020)
- [11] Konys, A.: An ontology-based knowledge modelling for a sustainability assessment domain. *Sustainability* **10**(2), 300 (2018)
- [12] Mehdi, et al.: Semdia: semantic rule-based equipment diagnostics tool. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 2507–2510 (2017)
- [13] Miao, et al.: Joint learning of degradation assessment and rul prediction for aeroengines via dual-task deep lstm networks. *IEEE Transactions on Industrial Informatics* **15**(9), 5023–5032 (2019)
- [14] Nilsson, J., Sandin, F.: Semantic interoperability in industry 4.0: Survey of recent developments and outlook. In: *2018 IEEE 16th international conference on industrial informatics (INDIN)*, pp. 127–132, IEEE (2018)
- [15] Ortega Lopez, G., Filatovas, E., Garzon, E.M., G. Casado, L.: Non-dominated sorting procedure for pareto dominance ranking on multicore cpu and/or gpu. *Jrnl. of Global Optim.* **69**, 1–21 (11 2017)
- [16] Proctor, M.: Drools: a rule engine for complex event processing. In: *Applications of Graph Transformations with Industrial Relevance: 4th International Symposium, AGTIVE 2011, Budapest, Hungary, October 4-7, 2011, Revised Selected and Invited Papers 4*, pp. 2–2, Springer (2012)
- [17] Qiushi: Combining chronicle mining and semantics for predictive maintenance in manufacturing processes. *Semantic Web* **11**(6), 927–948 (2020)
- [18] Romeo, L., Loncarski, J., Paolanti, M., Bocchini, G., Mancini, A., Frontoni, E.: Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0. *Expert Systems with Applications* **140**, 112869 (08 2019)
- [19] Sezer, E., Romero, D., Guedea, F., Macchi, M., Emmanouilidis, C.: An industry 4.0-enabled low cost predictive maintenance approach for smes. In: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pp. 1–8, IEEE (2018)
- [20] Sikorska, J., et al.: Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing* **25**(5), 1803–1836 (2011), ISSN 0888-3270
- [21] Tiwari, R., Sharma, N., Kaushik, I., Tiwari, A., Bhushan, B.: Evolution of iot & data analytics using deep learning. In: *2019 international conference on computing, communication, and intelligent systems (ICCCIS)*, pp. 418–423, IEEE (2019)
- [22] Wu, Y., Li, W., Wang, Y., Zhang, K.: Remaining useful life prediction of lithium-ion batteries using neural network and bat-based particle filter. *IEEE Access* **PP**, 1–1 (04 2019)
- [23] YuGi, et al.: Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal* **13**(3), 2213–2227 (2019), <https://doi.org/10.1109/JSYST.2019.2905565>
- [24] Zhang: Bearing performance degradation assessment using long short-term memory recurrent network. *Computers in Industry* **106**, 14–29 (2019)
- [25] Zhou, et al.: Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding. *Journal of Intelligent Manufacturing* **33**(4), 1139–1163 (2022)
- [26] Zonta, T., da Costa, C.A., Zeiser, F.A., de Oliveira Ramos, G., Kunst, R., da Rosa Righi, R.: A predictive maintenance model for optimizing production schedule using deep neural networks. *Journal of Manufacturing Systems* **62**, 450–462 (2022)