23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

An Ontology-based Approach for Failure Classification in Predictive Maintenance Using Fuzzy C-means and SWRL Rules

Qiushi Cao[a,*], Ahmed Samet[b], Cecilia Zanni-Merk[a], François de Bertrand de Beuvron[b], Christoph Reich[c]

[a]*Normandie Université, INSA Rouen, LITIS, 76000 Saint-Étienne-du-Rouvray, France*
[b]*ICUBE/SDC Team (UMR CNRS 7357)-Pole API BP 10413, 67412 Illkirch, France*
[c]*Hochschule Furtwangen University, 78120 Furtwangen, Germany*

Abstract

Within manufacturing processes, anomalies such as machinery faults and failures may lead to the outage situation of production lines. The outage of production lines is detrimental for the availability of production systems and may cause severe economic loss. To avoid the economic loss that may be caused by the outage situation, the prediction of anomalies on production lines is a crucial concern for manufacturers. Recently, data mining techniques have been applied to the manufacturing domain for predicting occurrence time of anomalies, such as the moment of machinery failure. However, existing predictive maintenance approaches have been limited to the prediction of the time of occurrence of machinery failures, while lacking the capability for identifying the criticality of the failures. This may lead to inappropriate maintenance plans and strategies. In this context, in this paper, we introduce a novel ontology-based approach to facilitate predictive maintenance in industry. The proposed approach is a combination use of fuzzy clustering and semantic technologies, where fuzzy clustering techniques are used to learn the criticality of failures based on machine historical data, and semantic technologies use the results of fuzzy clustering to predict the time of failures and the criticality of them. As results, a domain ontology for modeling predictive maintenance knowledge is developed, and a set of Semantic Web Rule Language (SWRL) predictive rules are proposed to reason about the time and criticality of machinery failures. A case study on a real-world industrial data set is followed to evaluate the usefulness and effectiveness of the proposed approach.

Keywords: Industry 4.0; Predictive maintenance; Ontologies; SWRL rules; Fuzzy clustering

---

* Corresponding author. Tel.: +33 769660272.
  E-mail address: qiushi.cao@insa-rouen.fr

1. Introduction

With the current trend of automation and data exchange in manufacturing technologies, traditional manufacturing factories are transforming into so-called "smart factories", which apply high-tech sensing and computation technologies on different manufacturing processes and production systems. As today's manufacturing market is becoming more competitive, how to improve the availability, sustainability, and quality of manufacturing services in smart factories is a crucial concern for manufactures. This situation has triggered the demand for implementing predictive maintenance on production lines, which refers to the maintenance activities that are performed to avoid failure occurrences and to improve the availability and safety of the maintained system.

In the domain of predictive maintenance, prediction of the moments of future anomalies (outliers) is an important aspect to prevent and resolve persistent dilemmas in maintenance management [2]. Normally, such a prediction approach starts with collecting and analyzing the historical data that are collected from the system being monitored. During the data collection and analysis phase, data mining techniques are widely used for automatically discovering knowledge from data sets. However, existing predictive maintenance approaches in the manufacturing domain are limited to the deployment of condition monitoring systems for detecting anomalies and predicting the time of machinery failures in the future, while lacking the solutions for identifying the criticality of machinery failures, based on their temporal information [2]. Also, to have a comprehensive understanding of the data mining results, it is required for users to have a deep understanding of the domain. This may bring obstacles to novice users for planning and performing adequate maintenance strategies. To deal with this issue, using ontology-based approaches can serve as a good representation for data mining results, since they can represent the extracted patterns in a formal and structured format, and give rich semantics to the data mining results. By definition, an ontology is a specification of representational vocabulary for a shared domain, and it is developed to support the sharing and interpretation of formally represented knowledge in artificial intelligence systems [11]. In this context, to ease the interpretation and reuse of data mining results, and to facilitate the classification of machinery failures according to their time of occurrences, semantic technologies, especially ontologies with their rule-based extensions are of paramount importance.

In this paper, we propose a novel hybrid ontology-based approach for predictive maintenance in manufacturing processes. The proposed approach is a combination use of fuzzy clustering and semantics. Among them, fuzzy clustering approaches, such as fuzzy c-means (FCM) [5], is used to learn the criticality of different failures in machine historical data. The different time duration from normal conditions of machinery to the failures are used as training examples to FCM with three fixed classes: High Criticality, Medium Criticality, and Low Criticality, which indicate the three levels of failure criticality. On the other hand, semantic technologies, such as ontologies and logic rules are used to predict the time and criticality of future failures. As results, the Manufacturing Failure Prediction Ontology (MFPO) is developed, and a set of Semantic Web Rule Language (SWRL) predictive rules [14] are proposed to reason about the criticality of machinery failures. To elaborate the proposed approach, a real-world data set about an industrial semi-conductor manufacturing process is used to demonstrate the whole process.

The remainder of the paper is structured as follows. In Section 2, the related works about using ontology-based approaches for predictive maintenance are introduced. In Section 3, the proposed ontology-based approach is demonstrated, including the description of the MFPO ontology, and the fuzzy SWRL rules for classifying failures. Section 4 elaborates the proposed approach according to a real-world industrial data set. Section 5 concludes the paper and outlines future perspectives.

2. Related work

As the manufacturing domain is becoming more dynamic and knowledge-intensive, the use of semantic technologies, especially ontology-based approaches for predictive maintenance turned to be a notable research topic. Recently, several ontologies and their rule-based extensions were proposed to facilitate knowledge representation and reuse in the predictive maintenance domain. In this section we review the most relevant research works.

The knowledge model for fleet predictive maintenance, introduced in [16], was developed to handle contextual knowledge within a fleet scale. In their work, the authors developed a domain ontology to categorize fleet elements into three levels. This categorization of fleet elements enables the analysis of contextual data from different levels, thus enabling the analysis of abnormal health conditions from different components within a fleet system. For the pre-

dictive maintenance in the wind energy domain, an ontology was developed and used as a basis of a fault detection and diagnosis system for wind turbines' condition monitoring [19]. The ontology models vital characteristics of a Wind Energy Converter's (WEC's) gearbox, and it has been used to detect possible failures and their exact positions in the WEC's gearbox, by executing SWRL rules and ontology queries. Among the existing research efforts in the manufacturing domain, the ontology introduced in [9] is one of the earliest contributions. In their work, a domain-specific ontology [10] was developed to streamline the implementation of industrial condition monitoring and to standardize the exchange of condition monitoring data. During the implementation of the condition monitoring system, the developed ontology served as a commonly accepted data and knowledge representation schema for diagnosis-oriented maintenance. Another notable contribution is OntoProg [17], which is an ontological model developed for the Prognostics Health Management (PHM) of mechanical items of manufacturing systems. OntoProg uses PHM terms in International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) as knowledge source. The significance of OntoProg is the standardization of the implementation of PHM in manufacturing systems, including the conceptualization of key concepts such as *FailureMode*, *PotencialCause* and *Symptom*. A set of SWRL rules were proposed to enhance the expressiveness of the ontology. At last, we mention PriMa, which is a prescriptive maintenance model for production systems in smart factories [2]. Within the framework of PriMa, ontologies and case-based reasoning are used to build semantic learning and reasoning models. The implementation of PriMa in real production systems has shown significant reduction of downtime.

The review of the related research work reveals two issues. Firstly, there is a missing link between the temporal information of an anomaly (e.g., the occurrence time of a future machinery failure) and the criticality of the anomaly. To assess the availability of computing systems, the outage duration is a key consideration which indicates the criticality of a mechanical failure [4]. Therefore, predicting the moments of machinery failures are crucial for computing the outage duration and the criticality of the failures. However, existing ontology-based approaches fail to propose decision makings about the criticality of machinery failures based on the occurrence times of future machinery failures. This brings obstacles to users for performing appropriate maintenance actions with considering time limitations. Secondly, since most of the existing ontologies and rule-based approaches are based on crisp logic, they are not competent in dealing with uncertain situations in predictive maintenance. For the classification of failures, using a fuzzy approach allows better expression of imprecise relations. For example, the classification of a failure can be associated with a fuzzy index, indicating the degree of its membership to a "low" or "high" criticality level. By this, the crisp logic-based rules can be transformed into fuzzy rules, which enhances the representation of imprecise criticality level of machinery failures. In this way, compared with the crisp logic-based approaches, the fuzzy approach provides a better solution for solving the symbol anchoring problem [7].

3. The hybrid ontology-based approach for predictive maintenance

This section introduces the proposed hybrid ontology-based approach for failure classification in predictive maintenance. The proposed approach starts with the implementation of sequential pattern mining (SPM) [1] on raw industrial data sets, after which frequent sequential patterns are obtained. The obtained frequent sequential patterns contain failure events as well as the temporal information of these failures (e.g., the time stamp indicating when the failure will happen). After that, fuzzy clustering technique such as FCM clustering is applied to the extracted temporal information of failures, in order to classify different failures according to their time. This classification enables the identification of failure criticality. Then, domain ontologies such as the MFPO ontology with its rule-based extension are used to formalize the domain knowledge and enable the prediction of the time and criticality of future failures. In this work, we consider the frequent sequential patterns that are in a rule format[1] This allows us to use these sequential patterns for ontology reasoning, to facilitate failure prediction in industry. Fig. 1 shows the steps described above.

---

[1] In this work we consider rules in the format $A \xrightarrow[[t_1,t_2]]{} Failure$ where A is the antecedent part of the rule that predicts a failure in $[t_1, t_2]$ time units.
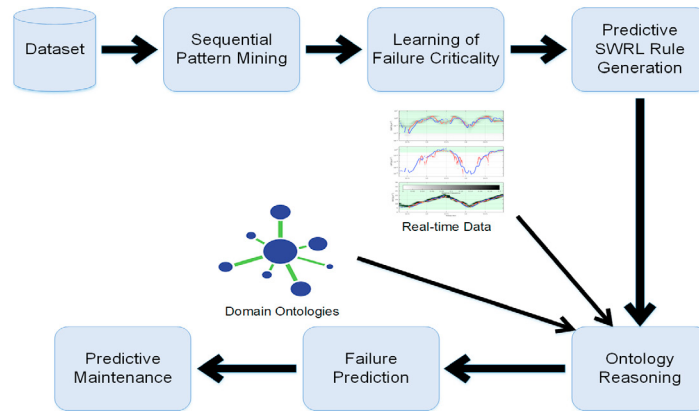
Fig. 1. Different steps of the hybrid ontology-based approach for predictive maintenance.

3.1. Predictive maintenance data collection and analysis

In smart factories, predictive maintenance tasks are performed by learning from historical maintenance data and predicting future events [2]. Within manufacturing systems, system operation data are collected from sensors, actuators, and mobile devices that are located at machine components and also the manufacturing environment. Normally, data collected for predictive maintenance are represented as sets of sequences with time stamps [20]. To deal with this type of data format, SPM techniques have been used to mine frequently occurring sequential patterns. In our approach, the aim of applying SPM to industrial data sets is to derive the temporal information of failurs. This enables the implementation of fuzzy clustering techniques on SPM results, for the identification of failure criticality based on their time of occurrences.

3.2. Fuzzy classification of failure criticality

In our previous work [6], one issue encountered is the incorrect classification of failure criticality. Since the failure classification method introduced in [6] is based on crisp logic, it failed to classify the criticality of a failure into the correct category when there are uncertain situations. For example, if a failure is predicted to happen within a considerably short amount of time after a "normal" condition, then the criticality level of this failure is identified as "high", meaning that maintenance actions need to be proposed immediately. However, if the predicted time duration between a "normal" condition and a failure falls into the "medium" category and the time duration is considerably close to the numerical threshold between "medium" and "high", then the system failed to classify the criticality of this failure into the correct category. To cope with this issue, a fuzzy approach which is able to handle such type of uncertainty situations is required. It should be noted that the failure criticality descriptor we consider in this paper is the time duration among normal conditions and machinery failures. Other descriptors that may affect the failure criticality, such as the level of mechanical component fracture and wear are out of the scope of this paper.

To cope with the aforementioned uncertain situations, in this work, the FCM algorithm is used for fuzzy clustering of failure criticality. After implementing the SPM step introduced in section 3.1, the frequent sequential patterns which contain temporal information (e.g., the time duration among "normal" events and the failure events) of failure events are extracted. The temporal information is represented as data points during the implementation of the FCM algorithm. Then the FCM algorithm is used to classify the failures by grouping similar data points into clusters. This clustering is achieved by iteratively minimizing a objective function which is dependent on the distance of the data points to the cluster centers. The objective function is computed by the following equation:

$$J = \sum_{j=1}^{N} \sum_{i=1}^{C} u_{ij}^{m} \| x_j - c_i \|^2, \tag{1}$$

where u_{ij} represents the degree of membership of the data point x_j in the ith cluster, c_j stands for the d-dimension center of the cluster, m is any real number greater than 1, and $\| \cdot \|$ denotes any norm expressing the similarity between any measured data and the center.

During the implementation of FCM algorithm, the objective function is minimized with the update of membership u_{ij} and the cluster centers c_j. The update of these two parameters is described by the following equations:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}(\frac{\|x_i-c_j\|}{\|x_i-c_k\|})^{\frac{2}{(m-1)}}}, c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}. \tag{2}$$

By this, the FCM algorithm starts with an initial guess for cluster centers, and then iteratively updates the cluster centers and the degrees of membership for each data point. The iteration stops when $max_{ij}|u_{ij}^{(k+1)} - u_{ij}^k| < \varepsilon$, where k is the number of iteration steps, and ε is a termination criterion [5]. The whole process converges when a local minimum value of J is obtained.

In this work, the time duration from normal conditions to failures are used as training examples to FCM with three fixed classes, and the three classes indicate three levels of failure criticality:

- High Criticality: time from a normal condition to the failure is relatively short and the production line should be stopped for immediate maintenance.
- Medium Criticality: the failure may happen after a moderate amount of time and machine operators need to plan maintenance in a limited time.
- Low Criticality: the failure may happen in the long future and machine operators will have sufficient time to plan maintenance actions.

The results of the FCM algorithm are formalized by the MFPO ontology and SWRL rules, which aim to predict the time of future failures and their criticality, thus facilitating the knowledge representation and interpretation in predictive maintenance of industrial manufacturing process.

3.3. The MFPO ontology

Within an intelligent system, ontologies contain the domain knowledge to operate. They have long been used for formally describing entities within an area of reality and all relationships among those entities [3]. Normally, ontologies comprise well-defined terms and well-defined relationships, and they allow ontology reasoning by which the individuals are classified into appropriate classes. In smart factories, to facilitate the knowledge representation and reuse of the SPM results, also to enable the classification of machinery failures according to their temporal information of occurrences, an ontology which formalizes knowledge of both manufacturing and predictive maintenance domains is required. In this work, we present the MFPO ontology, which is developed to formalize the domain knowledge in predictive maintenance.

The global architecture of the MFPO ontology is shown in Fig. 2. We use UML class diagram to represent concepts and properties, where boxes are classes in the MFPO ontology, and solid arrows are object properties. Data properties are indicated by class attributes. For the purpose of clarity, only a subset of the whole classes and properties are presented. The Web Ontology Language (OWL) [15] is chosen as the development language of the ontology. The MFPO ontology is an extended work of the ontology introduced in [6], and the descriptions of the common classes of the two ontologies (e.g., *RealizedPart*, *ManufacturingProcess*, *ManufacturingFacility*...) can be found in [6].

Compared to the ontology introduced in [6], the key extension proposed in the MFPO ontology is the capability of handling uncertainty. In the MFPO ontology, the nominal categories of classes are described by data properties. In this way, the classes in the MFPO ontology are associated with membership degrees which range from 0 to 1. For example, in the ontology introduced in [6], *hasFailureCriticality* is an object property whose domain is the class *Failure*, and range is the predefined individuals *Low*, *Medium* and *High*. After applying the aforementioned method, this object property is replaced by three data properties: *hasFailureCriticalityLow*, *hasFailureCriticalityMedium*, and
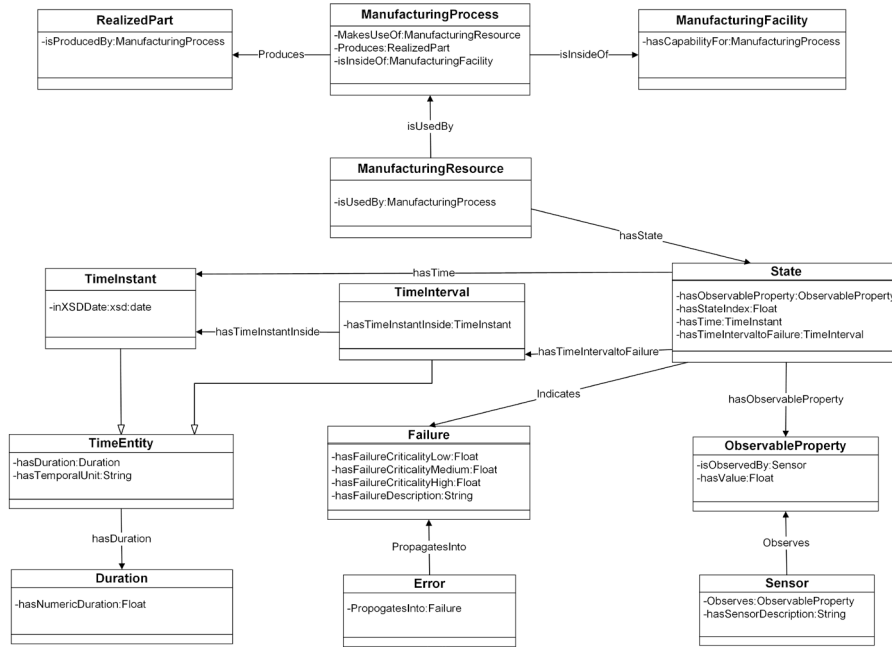
Fig. 2. The global architecture of the MFPO ontology.

hasFailureCriticalityHigh, and the sum of the numeric values of these three data properties is 1. By this, the *Failure* class in the MFPO ontology is associated with membership degrees to three nominal categories, thus enabling ontology reasoning of vague knowledge about this class.

3.4. Ontology reasoning for failure prediction

To predict the time of failures and also classify failures into different categories based on their temporal information, we propose SWRL rules [14] for ontology reasoning. The proposed SWRL rules reason on the individuals in the MFPO ontology, and infer new knowledge about failure prediction.

As stated in Fig. 2, the *State* of a *ManufacturingResource* is determined by a set of *ObservableProperties* (with their associated values). Based on this definition, we construct the antecedent of a SWRL rule by describing quantitative values of *ObservableProperties* (attributes) and the temporal information of the failure. The consequent of such a rule comprises the time constraints of the failure and its criticality. In this paper, to enhance the prediction of failures, we focus on the use of chronicles, which are a special type of sequential patterns. Inside a chronicle, events are ordered and temporal orders of events are quantified with numerical bounds [20]. Fig. 3 shows an example chornicle, where time constraints that describe the pattern {A, B, C} are noted by A[2,5]B, B[1,5]C and A[6,7]C. Here [2,5], [1,5] and [6,7] are lower and upper bounds of the time intervals among events. The reason for choosing chronicles in our approach is that they can be used to predict not only the failure event but also the time interval of its occurrence, as time dimension is crucial in predictive maintenance.

Fig. 3. A chronicle representing three events, within which the last event is a failure.

After obtaining the chronicles, to enable the automatic generation of SWRL rules, in this work we propose a novel algorithm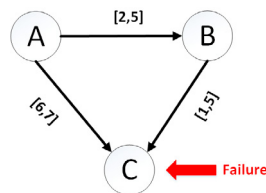 to transform chronicles (such as the one in Fig. 3) into predictive SWRL rules. Algorithm 1 demonstrates the general idea of our rule transformation method. It runs in five major steps:

1. the function *LastNonfailureState* extracts the last non-failure state (event) within a chronicle, and the function *LastNonfailureState* extracts the failure within a chronicle.
2. For each temporal constraint in a chronicle, the two functions *ProceedingEvent* and *SubsequentEvent* extract the proceeding and subsequent events of the time interval that is defined in this temporal constraint. Then the two events and this time interval forms different atoms in the antecedent of the rule, and they are treated as conjunctions.
3. For the last non-failure state before the failure, extract the time interval and its duration between this state and the failure. Then the descriptions of all normal states and this time duration form different atoms in the consequent of the rule, and they are treated as conjunctions.
4. FCM algorithm is applied to classify the failures according to their criticality. The failures are classified into three categories, and three object properties in the MFPO ontology are used to represent the degrees of membership to different clusters. The degrees of membership are treated as a conjunction, to form the consequent of the rule.
5. At last, a rule is constructed as an implication between the antecedent and the consequent.

Algorithm 1 Algorithm to transform a chronicle into a predictive SWRL rule.

Require: $\mathcal{S}_F$: a chronicle within which the last state (event) is a failure, $\mathcal{E}$: a set of the states that are described within a chronicle.

Ensure: R ▷ R: the SWRL rule to be constructed.

1: $ls \leftarrow LastNonfailureState(\mathcal{S}_F, \mathcal{E})$ ▷ Extract the last non-failure state before the failure within a chronicle.

2: $f \leftarrow theFailure(\mathcal{E})$ ▷ Extract the failure within a chronicle.

3: $R \leftarrow \emptyset$, $A \leftarrow \emptyset$, $C \leftarrow \emptyset$, $Atom_a \leftarrow \emptyset$, $Atom_c \leftarrow \emptyset$, $F_{FailureCriticalityLow} = 0$, $F_{FailureCriticalityMedium} = 0$, $F_{FailureCriticalityHigh} = 0$.   ▷ A: the antecedent of the SWRL rule. C: the consequent of the SWRL rule. $Atom_a$: a subset of all atoms within the antecedent. $Atom_c$: a subset of all atoms within the consequent. $F_{FailureCriticalityLow}$, $F_{FailureCriticalityMedium}$, $F_{FailureCriticalityHigh}$: the three degrees of membership to the three categories of failure criticality.

4: **for each** $e_i \in \mathcal{E}$ **do**

5:      $pe \leftarrow ProceedingState(e_i, \mathcal{S}_F)$ ▷ Extract the proceeding state

6:      $se \leftarrow SubsequentState(e_i, \mathcal{S}_F)$ ▷ Extract the subsequent state

7:      $Atom_a \leftarrow pe \wedge se$

8:      $A \leftarrow Atom_a \wedge pe \wedge se$

9: **end for each**

10: $ftd \leftarrow FailureTimeDuration(ls, f)$ ▷ Extract the time duration between the last non-failure state and the failure.

11: $F_{FailureCriticalityLow} \leftarrow DegreeofMembershipLow(ftd)$

12: $F_{FailureCriticalityMedium} \leftarrow DegreeofMembershipMedium(ftd)$

13: $F_{FailureCriticalityHigh} \leftarrow DegreeofMembershipHigh(ftd)$ ▷ Use the FCM algorithm to compute the degrees of membership of this time duration to the three clusters.

14: $C \leftarrow F_{FailureCriticalityLow} \wedge F_{FailureCriticalityMedium} \wedge F_{FailureCriticalityHigh} \wedge ftd$

15: $R \leftarrow (A \rightarrow C)$

16: **return** R

---

4. Experimentation

To evaluate the effectiveness of our approach, we conduct experimentation on a real world data set. In this work, we apply our approach on the SECOM data set [8], which contains measurements of features of semi-conductor productions within a semi-conductor manufacturing process. A software prototype is developed based on Java 10.0.2,

Protégé 5.5.0, OWL API [13] and SWRL API [18]. Among them, the OWL API is used to build and manipulate the MFPO ontology, and SWRL rules are proposed to reason about the criticality of failures. To enable ontology reasoning, the SWRL API, which includes a SWRL Rule Engine API, is used to create the transformed rules and then execute them. At last, the inferred knowledge is returned to the OWL API, and stored in the new ontology. The running environment of the software prototype is Microsoft Windows 10.

4.1. Frequent chronicle mining for failure prediction

To predict the criticality of failures, we implement the frequent chronicle mining approach introduced in [20]. The extraction of chronicles starts with feature selection method introduced in [12], after which a feature subset of the SECOM data set is obtained while retaining a suitably high accuracy in representing the original data set. As a result, 10 most relevant attributes are selected as the optimal sub-set of all 590 attributes. After the feature selection, data discretization is employed to discretize continuous values for obtaining nominal ones. Thereafter, data sequentialization is used to transform the data into the form of pairs (event, time stamp), where each sequence finishes with a failure. With obtaining sequences that contain failures, The CloSpan [22] algorithm is applied to the pre-processed data set, to extract frequent sequential patterns. Also, the frequent chronicle mining algorithm introduced in [20] is used to extract the temporal constraints among these sequential patterns. Up to this step, we are able to obtain frequent failure chronicles which contain temporal information about the failures.

To improve the quality of failure prediction and criticality classification, we take *Chronicle Support* as a reference measure, to select the most relevant failure chronicles. According to the definition in [20], the support of a chronicle C in a set of sequences is the number of its occurrences. As a result, only a subset of all frequent chronicles are used for generating SWRL rules. Table 1 shows the failure chronicles that have the 10 highest chronicle support. In Table 1, each failure chronicle is described by the number of events that it contains, the number of time intervals among events, all the observed properties (attributes) that characterize the failure chronicle, and the chronicle support. For the ease of demonstration, we label the 590 attributes as $A1, A2, A3...A590$. After applying the the FCM algorithm to the numeric values of the minimum time duration (Min_{TD}) among the last normal events and the failures, the degrees of membership of failure criticality to different categories are obtained. Table 2 gives the classification results.

Table 1. Extracted failure chronicles that have the highest 10 chronicle support.

Failure Chronicle	Number of Events	Number of Time Intervals	Attributes	Chronicle Support
C_{F1}	3	3	$A_{63}, A_{64}, A_{102}, A_{204}, A_{209}, A_{476}$	83.65%
C_{F2}	3	3	$A_{63}, A_{64}, A_{102}, A_{204}, A_{209}, A_{347}, A_{476}$	82.69%
C_{F3}	3	3	$A_{58}, A_{64}, A_{102}, A_{204}, A_{209}, A_{476}$	82.69%
C_{F4}	3	3	$A_{58}, A_{63}, A_{102}, A_{204}, A_{209}, A_{347}$	81.73%
C_{F5}	3	3	$A_{58}, A_{63}, A_{64}, A_{102}, A_{204}, A_{209}, A_{347}, A_{476}$	81.73%
C_{F6}	3	3	$A_{58}, A_{102}, A_{204}, A_{209}, A_{347}, A_{476}$	80.77%
C_{F7}	3	3	$A_{58}, A_{204}, A_{209}, A_{347}, A_{476}$	80.77%
C_{F8}	4	4	$A_{63}, A_{64}, A_{102}, A_{204}, A_{209}, A_{347}, A_{476}$	78.84%
C_{F9}	4	4	$A_{58}, A_{63}, A_{102}, A_{204}, A_{209}, A_{347}$	78.84%
C_{F10}	4	4	$A_{58}, A_{204}, A_{209}, A_{347}, A_{476}$	78.84%

Table 2. The classification results of the 10 failure chronicles.

Failure Chronicle	Min_{TD} (Time unit: millisecond)	Failure Criticality High	Failure Criticality Medium	Failure Criticality Low
C_{F1}	13800	0.056136	0.832054	0.111810
C_{F2}	12840	0.034679	0.046937	0.918384
C_{F3}	7440	0.256704	0.035270	0.708026
C_{F4}	10500	0.000722	0.000384	0.998893
C_{F5}	14640	0.069900	0.197297	0.732802
C_{F6}	2280	0.998489	0.000231	0.001280
C_{F7}	1020	0.988364	0.002024	0.009611
C_{F8}	22380	0.000099	0.999598	0.000303
C_{F9}	1560	0.000099	0.999598	0.000303
C_{F10}	4320	0.997523	0.000409	0.002068

4.2. The transformation from chronicles to SWRL rules

Based on Algorithm 1 and the failure classification results in Table 2, the failure chronicles were transformed to SWRL rules. Fig. 4 presents an example SWRL rule we obtained after implementing the whole process. Firstly, one frequent chronicle that contains a failure is extracted by applying the frequent chronicle mining approach. Secondly, after the mining of frequent failure chronicles, the numeric value of the Min_{TD} between the last normal state (event) inside a failure chronicle and the failure is clustered by the FCM algorithm. The arrow at the right side of the figure refers to the data point in the FCM classification results that represents the Min_{TD} value contained in the failure chronicle. Thirdly, one SWRL rule is constructed based on the degrees of membership to the three categories. In this rule, *hasA63V*, *hasA102V*, *hasA209V*, *hasA347V*, *hasA476V* are data properties in the MFPO ontology that link individuals of the *State* class with XML Schema Datatype values. They correspond to the quantitative values of the attributes A_{63}, A_{102}, A_{209}, A_{347}, and A_{476} in the SECOM data set. *TimeInterval* corresponds to the root class of all individuals of time intervals. There are two object properties that link *TimeInterval* with *State*: *hasSubState* and *hasProState*, among which *hasSubState* corresponds to the subsequent state of a time interval, and *hasProState* indicates the proceeding state of a time interval. In this case, state S_1 is the proceeding state of the time interval between S_1 and S_2, and state S_2 is the subsequent state of this time interval. To describe the numerical intervals which are obtained by discretization, SWRL Built-Ins are used to specify the upper and lower numerical boundaries. The consequent of this rule comprises time constraints of the failure, different categories of failure criticatily, and their degrees of membership. The minimum and maximum time duration between a state with the failure is described by the data property *hasMinF* and *hasMaxF*. The degrees of membership to different criticality categories of the failure is given by the FCM algorithm, and they are computed based on the minimum time duration between a normal state to the failure. By this way, the classification of the failure is inferred by the launching of such a predictive SWRL rule.

Following this procedure, the classification results presented in Table. 2 are formalized by the MFPO ontology and SWRL rules, which aim to facilitate the predictive maintenance of the semi-conductor manufacturing process.

Fig. 4. The procedure of transforming a failure chronicle into a SWRL rule, based on the failure classification results.

5. Conclusion

In this paper, we tackle the predictive maintenance task by introducing a hybrid ontology-based approach. The proposed approach is based on the combined use of fuzzy clustering and semantics, where fuzzy clustering techniques are used to learn the criticality of failures based on machine historical data, and semantic technologies use the results of fuzzy clustering to predict the time of failures and the criticality of them. The contributions of this paper lie firstly in the formalization of the predictive maintenance knowledge based on ontologies, by which the SPM results are formalized and the criticality of failures are inferred. Secondly, the classification of failures is achieved by implement-

ing an unsupervised learning approach. The unsupervised learning approach uses FCM algorithm to cluster failures according to their time of occurrences, which reflects the criticality of them. Thirdly, the SWRL rules are proposed to formalize the classification results, in order to facilitate knowledge representation and interpretation for predictive maintenance.

For our future work, we will be working on two different issues. The first issue is the capitalization of experience. When the introduced rules fail to provide satisfactory results about failure prediction, expert rules need to be proposed to provide appropriate solutions. In this way, when the next time that the system fails to predict the machinery failures properly, these expert rules will also be launched to facilitate decision making for failure prediction. The second issue is the consideration of meta-knowledge, such as the context. Since the predictive maintenance systems are becoming more context sensitive [6], different sets of rules should be launched according to the different context of the monitored manufacturing process. To this end, context modeling and reasoning [21] will be used to facilitate context representation and context sharing.

Acknowledgements

References

[1] Agrawal, R., Srikant, R., et al., 1995. Mining sequential patterns, in: icde, pp. 3–14.

[2] Ansari, F., Glawar, R., Nemeth, T., 2019. Prima: a prescriptive maintenance model for cyber-physical production systems. International Journal of Computer Integrated Manufacturing , 1–22.

[3] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al., 2000. Gene ontology: tool for the unification of biology. Nature genetics 25, 25.

[4] Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C., 2004. Basic concepts and taxonomy of dependable and secure computing. IEEE transactions on dependable and secure computing 1, 11–33.

[5] Bezdek, J.C., Ehrlich, R., Full, W., 1984. Fcm: The fuzzy c-means clustering algorithm. Computers & Geosciences 10, 191–203.

[6] Cao, Q., Giustozzi, F., Zanni-Merk, C., de Bertrand de Beuvron, F., Reich, C., 2019. Smart condition monitoring for industry 4.0 manufacturing processes: An ontology-based approach. Cybernetics and Systems , 1–15.

[7] Coradeschi, S., Saffiotti, A., 2003. An introduction to the anchoring problem. Robotics and Autonomous Systems 43, 85–96.

[8] Dua, D., Graff, C., 2017. UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

[9] Emmanouilidis, C., Fumagalli, L., Jantunen, E., Pistofidis, P., Macchi, M., Garetti, M., et al., 2010. Condition monitoring based on incremental learning and domain ontology for condition-based maintenance, in: 11th International Conference on Advances in Production Management Systems, APMS.

[10] Fensel, D., 2001. Ontologies, in: Ontologies. Springer, pp. 11–18.

[11] Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge acquisition 5, 199–220.

[12] Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research 3, 1157–1182.

[13] Horridge, M., Bechhofer, S., 2011. The owl api: A java api for owl ontologies. Semantic Web 2, 11–21.

[14] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al., 2004. Swrl: A semantic web rule language combining owl and ruleml. W3C Member submission 21.

[15] McGuinness, D.L., Van Harmelen, F., et al., 2004. Owl web ontology language overview. W3C recommendation 10, 2004.

[16] Monnin, M., Abichou, B., Voisin, A., Mozzati, C., 2011. Fleet historical cases for predictive maintenance, in: The International Conference Surveillance, pp. 25–26.

[17] Nuñez, D.L., Borsato, M., 2018. Ontoprog: An ontology-based model for implementing prognostics health management in mechanical machines. Advanced Engineering Informatics 38, 746–759.

[18] O'Connor, M.J., Shankar, R.D., Musen, M.A., Das, A.K., Nyulas, C., 2008. The swrlapi: A development environment for working with swrl rules., in: OWLED.

[19] Papadopoulos, P., Cipcigan, L., 2009. Wind turbines' condition monitoring: an ontology model, in: 2009 International Conference on Sustainable Power Generation and Supply, IEEE. pp. 1–4.

[20] Sellami, C., Samet, A., Tobji, M.A.B., 2018. Frequent chronicle mining: Application on predictive maintenance, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE. pp. 1388–1393.

[21] Wang, X., Zhang, D., Gu, T., Pung, H.K., et al., 2004. Ontology based context modeling and reasoning using owl., in: Percom workshops, Citeseer. p. 22.

[22] Yan, X., Han, J., Afshar, R., 2003. Clospan: Mining: Closed sequential patterns in large datasets, in: Proceedings of the 2003 SIAM international conference on data mining, SIAM. pp. 166–177.