Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 200 (2022) 944–953

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Industry 4.0 and Smart Manufacturing

# Root Cause Analysis in the Industrial Domain using Knowledge Graphs: A Case Study on Power Transformers

Jorge Martinez-Gil[a,*], Georg Buchgeher[a], David Gabauer[a], Bernhard Freudenthaler[a], Dominik Filipiak[b], Anna Fensel[b,c]

[a] *Software Competence Center Hagenberg GmbH*
*Softwarepark 32a, 4232 Hagenberg, Austria*
[b] *Semantic Technology Institute Innsbruck, University of Innsbruck*
*Technikerstraße 21a, 6020 Innsbruck, Austria*
[c] *Wageningen University and Research*
*6708 PB Wageningen, the Netherlands*

**Abstract**

In the industrial domain, developing solutions that allow the identification, understanding, and correction of faults is essential due to the cost of handling such situations. However, to date, there are not many solutions capable of facilitating the human operator to discern the causes and possible solutions for a specific fault. In this work, we present knowledge graph-driven root cause analysis for working with faults in the industrial domain, based on three points of action: reasoning from the current state of machines or processes, failure classification using rules, and advanced querying using graph-query languages. We have conducted a power transformer case study that revealed that our proposed approach could be considered competitive as it has outperformed several alternative machine learning classifiers.



*Keywords:* Knowledge Graphs, Root Cause Analysis, Manufacturing and Production

## 1. Introduction

In the manufacturing and production field, failures are often a heavy burden to solve as they are associated with large resource consumption in time and cost due to machine downtime. The development of systematic solutions could mitigate this to guide the problem solver to assess and understand the causes of the failures [14]. Nowadays, there is a strong need for techniques and tools for fault detection in industrial environments. It is expected that this need will

* Corresponding author
*E-mail address:* jorge.martinez-gil@scch.at


10.1016/j.procs.2022.01.292

continue to grow as more relevant data is generated. However, to date, the use of techniques inspired in Knowledge Graphs (KGs) has not yet been intensely studied in this domain [2].

In the context of this work, we aim to showcase how KGs can facilitate the detection and prevention of faults in modern industrial environments such as Industry 4.0 [1]. Our hypothesis is that research on KGs is a natural path to explore in this scenario since it is widely assumed that domain-specific knowledge needs to be considered to define unwanted anomalies accurately [5]. Our research solution aims to investigate the root cause of the potential malfunctions in a wide range of industrial domains, although in the specific context of this work, we will look at a use case related to power transformers.

It is necessary to remark that a KG can naturally model a comprehensive catalog of symptoms, failures, and possible solutions for either processes or machinery of industrial nature. This means that all the information needed to identify the problem (through the symptoms), detect the faults (through the malfunctioning of some part of the system), and offer solutions (through the expert knowledge already built into the model) is self-contained. So it represents a significant advantage concerning most of the existing solutions that need to use external resources to operate.

Our research efforts are focused on three different operations that can be naturally performed through KGs:

- A reasoning mechanism for an automatic recommendation of possible solutions based on domain knowledge (reasoning over the T-Box)
- A powerful inference mechanism based on rules to determine the status of a given machine or process (reasoning over the A-Box)
- An advanced method for qualitative querying based on the SPARQL query language

To illustrate our research results, we will focus on performing Root Cause Analysis (RCA) of faults in power transformers. The reason is that properly determining the causes of a fault is a significant challenge to ensure the reliability and power quality in industrial facilities. That means that all possible failures need to be monitored to ensure the reliability of the power supply, and thus save costs due to shorter downtime. So we think that a solution of this kind can significantly assist human operators in performing their maintenance tasks.

Therefore, the major research contributions of this work can be summarized as follows:

- We show how KGs perform RCA to investigate failure-related issues in manufacturing and production processes through several mechanisms. We investigate how different methods (i.e., cause recommendation, failure classification, and advanced queries) can be integrated with production environments and how these methods can be provided to human operators for performing RCA.
- Besides, we show a real-world use case on power transformers where the benefits of a KG-driven solution are demonstrated.

The remainder of this paper is structured as follows. Section 2 focuses on the state-of-the-art RCA literature within the industrial domain. Section 3 introduces KGs and argues why their application is suitable for working with industrial sectors' failures. Section 4 presents our proposed methodology based on a real-world power transformer case study. Section 5 discusses our results. Finally, we conclude our paper and provide possible lines of future research in Section 6.

## 2. Root Cause Analysis

Due to the technical complexity involved in most of today's machine and process errors and failures, RCA techniques are of increasing importance to improve their analysis and diagnosis [10]. For these reasons, technologies based on Artificial Intelligence, such as KGs, represent a quantitative and qualitative framework in creating value for organizations in the industrial sector by increasing the accuracy and efficiency of such analysis and diagnosis processes [17].

In the frame of this work, we can formally define RCA as "a process designed for use in investigating and categorizing the root cause of events with safety, health, environmental, quality, reliability, and production impacts" [18].

This implies that RCA goes beyond simply dealing with machine or process malfunctions. It also involves the ability to prevent them, in what is known as prescriptive analytics.

Although after several decades of research, numerous techniques and methods have been proposed to perform RCA in different domains [7, 9, 12], there is still a significant lack of understanding about how such techniques can be adapted to environments with increasing complexity, such as in a modern industrial setting.

According to the literature, some of the most popular methods for identifying industrial systems' incorrect behavior are RCA and Anomaly Detection (AD) [3]. The main difference between them is that RCA tries to provide clues to the human operator to identify and understand the natural causes of the problem, whereas AD focuses only on identifying events that do not conform to specific patterns. Therefore, it is expected that the malfunction can be solved with the least possible consumption of resources.

Following Solé et al. [20], models for RCA can be roughly divided into two kinds depending on the presence of uncertainty within them: deterministic (based on, e.g., decision trees or first-order logic) and probabilistic (based on, e.g., Bayesian techniques or fuzzy logic). Models can be generated in a manual (experts provide domain knowledge and create the model[15]), automated (model is automatically made from data [11]), or assisted way (which combines the previous two [21]). Inference depends on a chosen model, which determines the concept of evaluation of such models (for example, that can be accuracy for rule-based inference or Maximum A Posteriori for Bayesian models).

Apart from models, due to the importance of the industrial sector in general and manufacturing and production industries in particular, many strategies have been proposed to proceed with RCA [25]. In the literature, it is possible to observe the following four major groups:

- *Event correlation analysis-based methods:* These methods attempt to find the correlation between given activities in order to identify the root cause of a failure. The working model consists of identifying the few events that are of essential importance in the information flow.
- *Log-based methods:* Another common approach is to analyze the log files to identify problems that have occurred on machines or in processes. In this way, mining log patterns can give important clues about failures in devices or processes.
- *Execution path mining-based methods:* Execution path mining-based methods are typically used to perform real-time RCA. Their mode of operation is based on tracing transaction flows between services or system modules. In that fashion, it is possible to have a global view of the system's current operations, and, therefore, it should be feasible to discover bottlenecks.
- *Dependency graph mining-based methods:* In recent years, methods based on dependency graphs have also been studied. The rationale behind this approach is to use a directed graph representing several dependencies among entities. Thus, the malfunctioning of an entity causes the subsequent malfunctioning of its associates.

However, to date, there is little work based on KGs. Several solutions can use declarative knowledge, including representation languages with a standard syntax and semantics for operators such as $\wedge, \vee, \neg, \Rightarrow, \exists$, but it has gone no further [21]. It seems reasonable to think that it could improve using vocabularies of representational terms in the form of human-readable text and machine-enforceable, declarative constraints on their well-formed use [11]. Term definitions could include restrictions on domains and ranges, placement in subsumption hierarchies, class-wide facts inherited to instances, and other associated rules. The adoption of that methodology would be the catalyst for the use of KG in this domain.

Moreover, RCA workflows based on KGs should contain two crucial steps: model construction and inference. The first one considers combining different types of available knowledge to yield a model based on vocabularies of representational terms. Moreover, a model of this kind would be used for the inference (paired with new observations from its domain of interests). In the remainder of this work, we articulate this notion.

## 3. Root Cause Analysis and Knowledge Graphs

To the best of our knowledge, the use of KGs remains largely unexplored in the RCA domain. However, this does not mean that there are no knowledge-intensive approaches to address the problem. In fact, there are already strategies that make intensive use of structured information to perform RCA. In these approaches knowledge is usually provided by domain experts who have extensive experience in risk analysis. Through this approach all possible failures and their observable effects on the system are defined. Among a number of strategies following this approach, the Failure Mode and Effects Analysis (FMEA) [6] and Fault Tree Analysis (FTA) [22] propose templates to collect this type of information. The main contribution of this study is to bring KGs to the table and illustrate their effectiveness when it comes to RCA.

A Knowledge Graph can be defined as pair whereby a set of concepts and a set of edges or ordered pairs of vertices so that an edge is associated with two distinct vertices. Even though KGs can represent concepts and relationships between them, an inference logic is required to determine that a failure has occurred. For this purpose, rule languages such as the Semantic Web Rule Language (SWRL) [8] might be used as these languages make it possible to define inference rules, which can be used in conjunction with a reasoner to obtain logical consequences. Moreover, KGs are well suited for describing knowledge in a way machines can exploit. In this work, we are focusing on KGs based on description logics. Therefore, we will use the Web Ontology Language[1] (OWL), considering that it is one of the most popular languages for formalizing KGs. Another benefit of OWL is that it is based on the well-known Resource Description Framework[2] (RDF), a widely-applied standard.

This way of working assumes that statements represent information. Each statement is composed of a subject, a predicate, and an object. Nodes either represent entities or attributes, while vertices illustrate relationships between entities or between entities and attributes. Based upon this definition, subjects and objects are usually nodes, while predicates are vertices. One of the technological advantages of RDF is the possibility that data can be stored through a serialization process (which includes a myriad of formats, such as RDF/XML, N-Triples, Turtle, RDFa, and JSON-LD). Subsequently, we are also considering the SWRL technology, as it allows processing rules that contain an antecedent and a consequent. This implies that if the conditions specified in the antecedent are reached, then the actions specified in the consequent must be executed. Based on that, the inference engine can be built.

The final part of the KG-driven RCA requires a semantic query language. In our case, SPARQL [16] is used to retrieve and manipulate stored data effectively and efficiently. We will present how prospective queries can be made thanks to this technology. In our particular case, RCA design is about building the KG and applying reasoning methods to suggest possible causes, reasoning by inference rules to detect failures, and an advanced querying mechanism to do an exploratory analysis of the situation. Therefore, the next section is dedicated to a case study on how KGs can perform RCA on power transformers.

## 4. Case Study

This section explains the use of KGs for RCA and it is structured as follows: We start with a problem statement dealing with failures in the power transformers domain. Then, we explain the methods we are going to work with within our case study. Furthermore, we explain how the methods for the recommendation of possible causes, the automatic classification using rules, and the advanced query using the SPARQL query language work in this domain. Finally, we offer a short discussion of the results obtained.

### *4.1. Problem Statement*

Power transformers are one of the most vital and critical equipment in power systems. The most important role of power transformers is to adjust voltage and current without affecting the total power. We can informally define a power transformer as an electrical device capable of transferring electrical energy from one circuit to another. An electric current in one coil of the power transformer produces a magnetic flux in the core of another power transformer, which causes the induction of an electromotive force in the other coils of the core.

When working with power transformers, if the power is fixed, when voltage AC is increased, then current will be decreased to ensure the total power is not affected. In this context, one of the most widespread ways to study failures is through the analysis of gases such as $CO$, $CO_2$, $CH_4$, $C_2H_6$, $H_2$, $O_2$, $N_2$ [23]. However, this traditional analysis has the disadvantages of a long fault diagnosis procedure, low fault location efficiency, reliance on individual ability, and

[1] `https://www.w3.org/OWL/`
[2] `https://www.w3.org/RDF/`

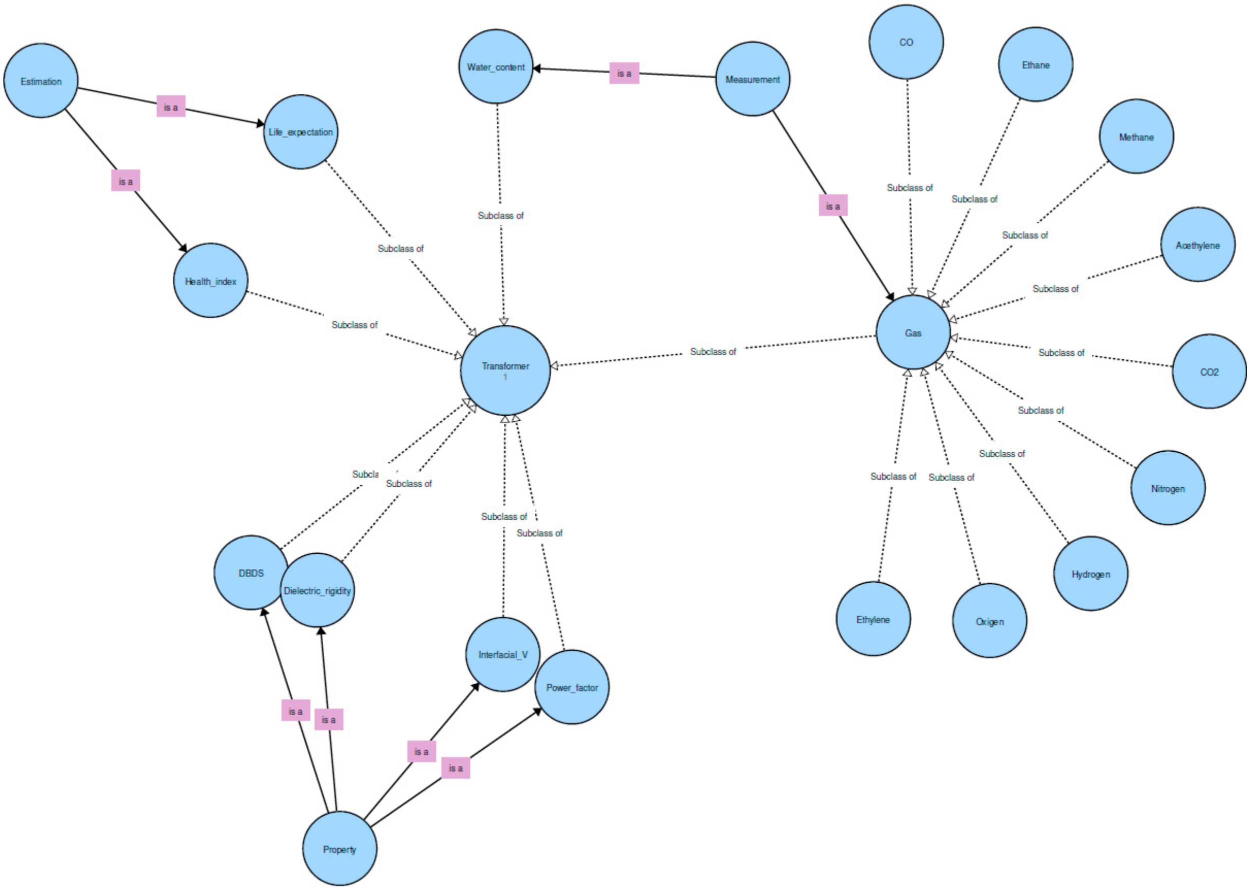


Fig. 1. KG representing a power transformer

large workforce investment [24]. Further ways of analyzing power transformers deal with the classification, clustering, and forecasting of multiple power transformer properties using artificial neural networks (ANNs) and support vector machines (SVMs) [26]. Even though those frameworks lead to highly accurate results, those models deal with a multitude of shortcomings, such as the fact that they require large amounts of data to be trained on which are often not available, the resulting model is complicated/infeasible to be interpreted by human operators, and the results are often not reproducible as the parameters vary depending on the number of epochs, batch size, learning rate, optimizer, etc. [13]. Therefore, the usage of KGs can be beneficial in overcoming those limitations.

### *4.2. Methods*

Within this study, we assume the model proposed by Velásquez et al. [23] which is outlined in Figure 1 and primarily based on measurements, estimates, and properties. Examples of measurements are the *Water level* or the *level of various gases* ($CO$, $CO_2$, $CH_4$, $C_2H_6$, etc.), estimations could assume the *Health index* or *Life expectation*, and properties are the *Power Factor* or *Dielectric rigidity*. This framework enables the derivation of specific actions given knowledge represented.

#### *4.2.1. Cause recommendation*

One substantial advantage of applying KGs is that they can emulate the human ability to perform inference. To do that, one need to assume a categorical target variable $y \in C = \{1, 2, \ldots, N\}$ where $C$ represents a set of failures classes. Subsequently, we can define a classification rule represented by a function $d(\mathbf{x})$ that defines all possible features $\mathcal{X}$ so that for every $\mathbf{x}$, $d(\mathbf{x})$ can be mapped to one class. As a result, all failures, including symptoms and solutions, need to be documented by experts.

#### *4.2.2. Failure Classification Using Rules*

Inference rules are not only prepared to work at the level of concepts and relationships. It is also possible to work at the numerical level to shape a rule-based classifier. The main limitation of such classifiers is that experts need to perform feature engineering and define rules in the form of $(Condition) \rightarrow y$. However, with such information, the inference engine can perform quite well. We will compare the quality of those rules with alternative classification approaches.

#### *4.2.3. Advanced queries*

Finally, a query language that is a specialized programming language for searching and changing graph's contents is implemented to retrieve and manipulate data. For example, all parts whose weight is less than 100 grams and whose estimated life span is less than one year can be automatically listed. This type of query is unthinkable in most technical manuals written in natural language. For this reason, documentation based on KGs facilitates the work of the operators, as we will also see in this section.

### *4.3. Cause Recommendation*

The first of the methods we will look at consists of the automatic recommendation of causes for a specific failure. It is necessary to have a KG in which the operator's observations regarding the state of the machinery or the process can be updated in real-time. Once these observations have been completed, reasoning can be conducted to determine the recommendations to be made.

The idea is that given a set of nodes $N$ and a set of relationships $R$; one needs to use a subset of the cross product $N \times R \times N$ to identify which $n \in N$ is the most similar. Moreover, in the context of this work, we might operate according to the close-world hypothesis [19] as we restrict ourselves to a particular and closed domain.

In our case, our $KG = (T, A)$ consists of a TBox $T$ and an ABox $A$ is assigned a semantics in terms of set-theoretic interpretations $I = (\Delta^I, \cdot^I)$, where $\Delta^I$ is a non-empty *domain* and $\cdot^I$ is the *interpretation function* that assigns an element in $\Delta^I$ to each $a \in I$.

### *4.4. Failure Classification*

Inference is a fundamental task in AI, which is, in turn, the area in which most of the solutions concerning RCA are framed. There are two prominent families of solutions: Deterministic solutions and probabilistic solutions. Deterministic solutions have no uncertainty associated with them, and all factors are known in advance. On the other hand, probabilistic solutions handle a certain degree of uncertainty. In this case study, we have come up with the following set of deterministic rules.

```
Life_expectation(?p) ^ hasValue(?p,?val) ^
    swrlb:lesserThan(?val,0.5^^xsd:float) → Failure(?p))
Life_expectation(?p) ^ hasValue(?p,?val) ^
    swrlb:greaterThan(?val,0.5^^xsd:float) → ComplementOf(Failure(?p))
Life_expectation(?p) ^ hasValue(?p,?val) ^
    swrlb:greaterThan(?val,0.5^^xsd:float) ^
    Nitrogen(?p) ^ hasLevel(?p,?level) ^
    swrlb:greaterThan(?level,62651.0^^xsd:float) → Failure(?p)
```

It is necessary to remark that items such as $Life\_expectation(?p)$ or $Nitrogen(?p)$ make direct reference to the dataset attributes. The symbol îs equivalent to an AND operator. Items of type $hasValue(?p, ?val)$ allow to obtain the value of the specific attribute $p$ in the variable $val$. Furthermore, last but not least, the constructors of type $swrlb : greaterThan$ are rule-language tools that implement comparison operators.

In this way, the code that we can see tells us that, as a general rule, a power transformer with a lifetime attribute less than 0.5 should be categorized as a failure. Similarly, if it is greater than 0.5 and with a nitrogen level greater than 62651, it should also be categorized as a failure. Although such rules seem simple, they can classify dataset instances with high accuracy. This is due a they use expert knowledge expressed in the form of rules. Furthermore, accuracy is chosen here since this measure is able to determine the percentage of times the classifier is able to correctly guess if the power transformer operating mode is correct.

Table 1. Classification results of various frameworks using the Velasquez dataset

| **Algorithm** | **Accuracy** |
|---|---|
| Multilayer Perceptron | 0.763 |
| Support Vector Machine | 0.780 |
| K-Nearest Neighbors | 0.797 |
| Random Forest | 0.814 |
| **SWRL (expert rules)** | 0.814 |

With the set mentioned above of inference rules, we have solved the issue of one benchmark dataset[3] that is prevalent in the domain of fault detection in power transformers. Table 1 highlights the results obtained from these rules and summarizes the comparison with state-of-the-art classifiers. It is shown that the results obtained from the KG are competitive and outperform a multitude of alternative techniques. Although it is true that to perform the experiments we have not proceeded to the optimization of the parameters, but have used the default values of the Scikit Learn free software machine learning library[4].

Moreover, one significant advantage of this approach is its interpretability [4]. By interpretability, we mean the possibility that a human operator can fully understand the process that has been followed until a decision is reached, or more formally, it is the ability to identify an intuitive mapping function between inputs and outputs. As inference rules are usually formulated very close to natural language, they are included among the most interpretable models. The majority of employed techniques in machine learning are non-interpretable out of the box (random forests, support vector machines, neural networks, etc.).

### *4.5. Advanced Querying*

Another interesting functionality that KGs make available to researchers and practitioners in the field of Root Cause Analysis is the capability of performing advanced queries over the KGs using graph query languages. In this case, we are using SPARQL to query the KG concerning power transformers that we have previously explained. Through these advanced graph query mechanisms, it is possible to check multiple facets of machine states or processes. For illustrative purposes, we can calculate the different pieces of a power transformer (i.e., PW101) given a volume property (associated with Water Level) greater than 100. The following query will print, in double column, each such piece $?a$ and its associated volume $?b$ if it is greater than 100.

```
SELECT DISTINCT ?a ?b WHERE {
  ?a RDF:type UC:Piece .
  ?a UC:part_of UC:PW101 .
  ?a UC:volume ?b.
  FILTER ( ?b > 100)
}
```

Furthermore, many prospective graph queries could support the human operator's task of RCA. For example, the following SPAQRL query will display the number of pieces *count*($?a$) about a given power transformer (i.e., PW101) weighing more than 30 grams.

---

[3] http://dx.doi.org/10.17632/rz75w3fkxy.1
[4] https://scikit-learn.org/

```
SELECT DISTINCT (count(?a) as ?count) WHERE {
  ?a RDF:type UC:Piece .
  ?a UC:part_of UC:PW101 .
  ?a UC:weight ?b.
  FILTER ( ?b > 30)
}
```

Queries of this kind would be extremely difficult to perform on a documentations expressed in natural language. For this reason, it is often a good idea to prepare the machine and process live documentation in the form of KGs. In this way, it is easier for problem solvers to inspect the devices or processes in production. Moreover, the working example demonstrates a simple query for pedagogical purposes.

However, much more complex queries can also be formulated that ease the power transformer analysis. The learning curve regarding graph query language needs also to be considered. But once that the human operators learn how to proceed with such queries, people could be provided with solutions to facilitate the formulation and understanding of such queries (wizards, predefined formulations, parameterized queries, and so on).

## 5. Discussion

We have seen the capabilities that KGs have to operate in this domain and validated the KG-driven RCA approach by conducting a power transformer case study. Some of the most interesting lessons derived from this work are related to the RCA system's performance based on domain knowledge and how credible historical data of anomalies and failures can improve it. Moreover, we have found that the accuracy of the proposed KG-driven RCA approach has outperformed several alternative machine learning classifiers, at least, in its default configuration. This makes this type of solution an alternative to take into account when implementing solutions for fault detection and correction in industrial environments.

Some of the essential advantages of using KGs in a RCA context are the capability to perform automatic reasoning since the set of concepts, relations, instances, and rules are described formally, allowing the automatic computation for logical inference. Moreover, the chance to involve the domain experts in the process could increase the operator's confidence in the retrieved solution. This might lead to continuous improvements in the system through relevance feedback. Last but not least, the KGs are interpretable what means that human operators can understand how the model has derived its final decision, thus ensuring a trustworthy system as opposed to the majority of black-box systems prevalent today in the industry.

However, there is still much pending work to improve KGs when working with numeric and tabular data since most of the data generated by machines or in factories are eminently numerical in nature (sensors, timestamps, measurements of other physical variables, etc.). In this way, further research is needed in order for KGs to become much more widely accepted in industrial settings. In this regard, there are several initiatives[5] that are stimulating research in this direction.

## 6. Concluding Remarks and Future Work

In this work, we have proposed a novel KG-driven RCA approach in an industrial setting. Although this model is not widespread in this area yet, we have described several situations in which its application might be beneficial. Furthermore, we have demonstrated how KGs, combined with inference rules, can automatically determine possible causes of failures and the possibility of advanced queries to help the human operator in the task of prospectively seeking the causes of malfunctions.

Additionally, this study has illustrated a case study with a focus on power transformers. For this purpose, we have shown three different actions that support human operators identifying and solving many transformer malfunctions. These actions comprise the automatic recommendation of possible causes for a given failure based on domain knowledge extracted from experts, the automatic classification of faults based on a set of rules defined using KG semantics,

[5] https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

and the advanced formulation of queries using a graph query language allowing operators to inspect the state of a machine or process.

Moreover, it should be noted that another significant advantage of our KG-based approach over the majority of state-of-the-art machine learning models is its interpretability that gives a chance to human operators to understand the model's decision-making process.

As part of future work, we are interested in an in-depth comparison between symbolic and statistical AI techniques. While the first techniques are based on a calculation using strings of characters representing real-world entities or concepts, the latter techniques process information of numeric nature. However, in an industrial context, which handles numerical information from sensors, logs, measurements, etc., we hypothesize that symbolic techniques play a crucial and complementary role in such environments. Possible solutions might consider developing domain-specific languages for rule specification, question answering, or recommendation systems. These concepts are more suited for human operators as specific interfaces could be provided. This means that appropriate tools for interacting with KGs need to be developed.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their comments and suggestions to improve the manuscript. Furthermore, the authors want to acknowledge the support from the Interreg Österreich-Bayern 2014-2020 Programme funded under grant agreement *(AB292)*. In addition, the research reported in this paper has also been funded by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Federal Ministry for Digital and Economic Affairs (BMDW), and the Province of Upper Austria in the frame of the COMET – Competence Centers for Excellent Technologies Programme managed by Austrian Research Promotion Agency FFG.

## References

[1] Sebastian R. Bader, Irlán Grangel-González, Priyanka Nanjappa, Maria-Esther Vidal, and Maria Maleshkova. A knowledge graph for industry 4.0. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, pages 465–480. Springer, 2020.

[2] Georg Buchgeher, David Gabauer, Jorge Martinez-Gil, and Lisa Ehrlinger. Knowledge graphs in manufacturing and production: A systematic literature review. *IEEE Access*, 9:55537–55554, 2021.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[4] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. Explainable ai in industry. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3203–3204, 2019.

[5] Irlán Grangel-González, Felix Lösch, and Anees ul Mehdi. Knowledge graphs for efficient integration and access of manufacturing data. In *25th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2020, Vienna, Austria, September 8-11, 2020*, pages 93–100. IEEE, 2020.

[6] Paul G. Hawkins and David J. Woollons. Failure modes and effects analysis of complex engineering systems using functional models. *Artif. Intell. Eng.*, 12(4):375–397, 1998.

[7] Longlong He and Pingyu Jiang. Manufacturing knowledge graph: A connectivism to answer production problems query with knowledge reuse. *IEEE Access*, 7:101231–101244, 2019.

[8] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79):1–31, 2004.

[9] Elem Güzel Kalayci, Irlán Grangel-González, Felix Lösch, Guohui Xiao, Anees ul Mehdi, Evgeny Kharlamov, and Diego Calvanese. Semantic integration of bosch manufacturing data using virtual knowledge graphs. In Jeff Z. Pan, Valentina A. M. Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, volume 12507 of *Lecture Notes in Computer Science*, pages 464–481. Springer, 2020.

[10] Salma Mahmoud, Florian Sobieczky, Jorge Martinez-Gil, Patrick Praher, and Bernhard Freudenthaler. Decay-parameter diagnosis in industrial domains by robustness through isotonic regression. *Procedia Computer Science*, 180:466–475, 2021.

[11] Jorge Martinez-Gil. Automated knowledge base management: A survey. *Comput. Sci. Rev.*, 18:1–9, 2015.

[12] Jorge Martinez-Gil, Bernhard Freudenthaler, and Thomas Natschläger. Automatic recommendation of prognosis measures for mechanical components based on massive text mining. *Int. J. Web Inf. Syst.*, 14(4):480–494, 2018.

[13] Mahak Mittal, Mani Bhushan, Shubhangi Patil, and Sushil Chaudhari. Optimal feature selection for svm based fault diagnosis in power transformers. *IFAC Proceedings Volumes*, 46(32):809–814, 2013.

[14] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM*, 62(8):36–43, 2019.

[15] Michael J. Pazzani. Refining the knowledge base of a diagnostic expert system: An application of failure-driven learning. In Tom Kehler and Stanley J. Rosenschein, editors, *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 2: Engineering*, pages 1029–1035. Morgan Kaufmann, 1986.

[16] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, 34(3):1–45, 2009.

[17] Ariam Rivas, Irlán Grangel-González, Diego Collarana, Jens Lehmann, and Maria-Esther Vidal. Unveiling relations in the industry 4.0 standards landscape based on knowledge graph embeddings. In *International Conference on Database and Expert Systems Applications*, pages 179–194. Springer, 2020.

[18] James J Rooney and Lee N Vanden Heuvel. Root cause analysis for beginners. *Quality progress*, 37(7):45–56, 2004.

[19] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[20] Marc Solé, Victor Muntés-Mulero, Annie Ibrahim Rana, and Giovani Estrada. Survey on models and techniques for root-cause analysis. *arXiv preprint arXiv:1701.08546*, 2017.

[21] Bram Steenwinckel, Dieter De Paepe, Sander Vanden Hautte, Pieter Heyvaert, Mohamed Bentefrit, Pieter Moens, Anastasia Dimou, Bruno Van Den Bossche, Filip De Turck, Sofie Van Hoecke, and Femke Ongenae. FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. *Future Gener. Comput. Syst.*, 116:30–48, 2021.

[22] Michael G. Thomason and E. W. Page. Boolean difference techniques in fault tree analysis. *Int. J. Parallel Program.*, 5(1):81–88, 1976.

[23] Ricardo Manuel Arias Velásquez and Jennifer Vanessa Mejía Lara. Root cause analysis improved with machine learning for failure analysis in power transformers. *Engineering Failure Analysis*, 115:104684, 2020.

[24] Ricardo Manuel Arias Velásquez, Jennifer Vanessa Mejía Lara, and Andres Melgar. Converting data into knowledge for preventing failures in power transformers. *Engineering Failure Analysis*, 101:215–229, 2019.

[25] Hehua Yan, Jun Yang, and Jiafu Wan. Knowime: A system to construct a knowledge graph for intelligent manufacturing equipment. *IEEE Access*, 8:41805–41813, 2020.

[26] Yuwen Zhang, X Ding, Y Liu, and PJ Griffin. An artificial neural network approach to transformer fault diagnosis. *IEEE transactions on power delivery*, 11(4):1836–1841, 1996.