Optimising Manufacturing Process with Bayesian Structure Learning and Knowledge Graphs

Tek Raj Chhetri[1(✉)], Sareh Aghaei[1], Anna Fensel[1,2], Ulrich Göhner[3], Sebnem Gül-Ficici[3], and Jorge Martinez-Gil[4]

[1] Semantic Technology Institute (STI) Innsbruck, Department of Computer Science, University of Innsbruck, Technikerstr. 21a, 6020 Innsbruck, Austria
{tekraj.chhetri,sareh.aghaei,anna.fensel}@sti2.at

[2] Wageningen Data Competence Center, Wageningen University and Research, Droevendaalsesteeg 2, 6708 PB Wageningen, The Netherlands

[3] Kempten University of Applied Sciences, Bahnhofstraße 61, 87435 Kempten, Germany
{ulrich.goehner,sebnem.guel-ficici}@hs-kempten.de

[4] Software Competence Center Hagenberg GmbH, Softwarepark 32a, 4232 Hagenberg, Austria
Jorge.Martinez-Gil@scch.at

Abstract. In manufacturing industry, product failure is costly, as it results in financial and time losses. Understanding the causes of product failure is critical for reducing the occurrence of failure and optimising the manufacturing process. As a result, a number of studies utilising data-driven approaches such as machine learning have been conducted to reduce the occurrence of this failure and to improve the manufacturing process. While these data-driven approaches enable pattern recognition, they lack the advantages associated with knowledge-driven approaches, such as knowledge representation and deductive reasoning. Similarly, knowledge-driven approaches lack the pattern-learning capabilities inherent in data-driven approaches such as machine learning. Therefore, in this paper, leveraging the advantages of both data-driven and knowledge-driven approaches, we present a strategy with a prototype implementation to reduce manufacturing product failure. The proposed strategy combines a data-driven technique, Bayesian structural learning, with a knowledge-based technique, knowledge graphs.

Keywords: Manufacturing product failure · Bayesian structural learning · Knowledge graphs · Structure learning

1 Introduction

Small and medium-sized enterprises (SMEs) as significant contributors in the manufacturing and production industry require ensuring a low failure rate of

R. Moreno-Díaz et al. (Eds.): EUROCAST 2022, LNCS 13789, pp. 594–602, 2022.
https://doi.org/10.1007/978-3-031-25312-6_70

products to have a healthy production line [8]. Product failure leads to a loss of market share with the increasing competition and customer expectations in the current era of Industry 4.0. Thus, understanding the causes of product failure is essential in order to eliminate the failures or reduce their effects and optimise the manufacturing process.

While manufacturers have made efforts to reduce the occurrence of product failure in SMEs, analysis of the causes by manual inspections is becoming less efficient, expensive, time-consuming and difficult [7]. To address failures occurring at manufacturing with complex processes, diverse techniques can be employed, including data-driven and knowledge-based (or semantic-based) approaches.

In recent years, data-driven approaches have made progress using machine learning (ML) for monitoring, fault diagnosis, optimisation and control. As a data-driven technique, Bayesian networks (BNs) are widely used to access a comprehensive and accurate analysis of complex systems. BNs are probabilistic graphical models to characterise and analyse uncertainty problems through a directed acyclic graph (DAG). The task of learning the dependency graph from data is called structure learning [11]. Although the state-of-the-art solutions (e.g., using continuous optimisation) have achieved learning the structure of a BN with many variables, they are not easily interpretable and informative about dependencies between the variables. In contrast, semantic-based techniques (e.g., using knowledge graphs (KGs)) allow to define the basic concepts and primary semantic relationships in a domain and provide deductive reasoning.

In this paper, we propose and develop a hybrid model for evaluating and predicting product quality in SMEs' production lines and consequently reducing the failure rates. We utilise structure learning to find and represent probabilistic dependency relationships among the variables and then use KGs to enrich semantic interoperability and exchange information between humans or machines. The main contributions of our paper are summarised as follows: (i) we take advantage of structure learning in BNs to reflect the dependencies among variables in SMEs' manufacturing processes through identifying DAGs; (ii) to overcome the lack of semantic interoperability in the extracted DAGs, we employ the idea of KGs; (iii) we generate and annotate an OWL ontology based on the DAG obtained through the Bayesian structure learning process to create a KG.

The paper is organised as follows. Section 2 provides an overview on the related works. The methodology is discussed in Sect. 3, and Sect. 4 provides a detailed explanation about the implementation. The evaluation is discussed in Sect. 5. Section 6 concludes the paper and gives directions for future research.

2 Related Works

A number of studies, such as [3], [4] and [10], have been conducted on the use of semantic technologies and BNs, demonstrating the advantages of combining the two. Existing research focuses on either integrating BNs into existing ontologies or using ontologies to model BNs. For example, Riali et al. [10] extended the ontology (i.e., fuzzy ontology) with BNs to incorporate probabilistic knowledge present in real-world applications. On the other hand, Chen et al. [4] use ontology to model BNs to represent causal relationships between additive manufacturing. Cao et al. [4], similarly, use ontology and BN to investigate dynamic risk propagation on supply chains. A risk propagation ontology is created (or customised) according to the domain and then it is transformed into a BN.

In summary, the work described above presumes ontology to be existing, which can be viewed as a limitation given the dynamic nature of the settings. In an ontology, for instance, all concepts are predefined, and if there is a change in manufacturing steps, such as the addition or subtraction of certain steps, the ontology must be modified accordingly. Our proposed work can account for these dynamic circumstances, automatically generating the ontology and KG, thus helping industry, especially SMEs that are often limited in resources.

3 Methodology

In this paper, we describe our approach to manufacturing process optimisation in detail. Figure 1 summarises the approach taken in our study, with details provided in the following subsections.

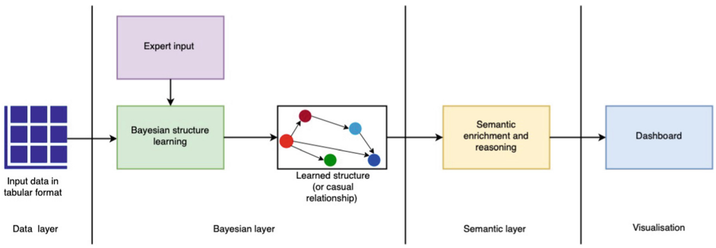


Fig. 1. Proposed methodology

3.1 Data Layer

The data layer is the first component to interact with the data. The data layer reads and preprocesses the input data. Preprocessing is used to ensure the data quality. For example, the input data for some features may be incomplete (i.e., it may contain missing values). Additionally, the data may include values on various scales. This is because missing values and inputs with varying scales result in suboptimal performance. The data layer's preprocessing performs imputation to fill in missing input and scaling values in different ranges to a common range.

3.2 Bayesian Layer

The Bayesian layer provides two major functionalities: learning the dependency graph of a BN from data, which is referred to as structure learning; and integrating expert inputs (or domain knowledge), a feature of BN [6]. The Bayesian layer yields the DAG as shown in Fig. 1 after performing the structure learning, which represents the learned relationships between the features. The Bayesian structure learning is based on [13] and [14], which perform structure learning by formulating combinatorial structure learning problem as continuous optimisation problems, thereby eliminating the combinatorial overhead.

3.3 Semantic Layer

The semantic layer enables capabilities such as reasoning and data enrichment inherent in semantic technology. Reasoning makes use of relationships and the deductive power of logic to generate new inferences (i.e., meaningfulness from the data). Additionally, the use of KG also enables interoperability, which is important when integrating with other external systems. To leverage semantic technology, the semantic layer converts the learned DAG to the corresponding semantic representation, specifically an ontology and a KG. Furthermore, the semantic layer provides reasoning via SPARQL queries.

Algorithm 1: OWL Ontology generator from DAG

```
Input: DAG graph G as an adjacency list
Result: OWL ontology O, ontology class mapper C_m, ontology object property mapper
        O_m, ontology data property mapper D_m
OWL Ontology ← initialise namespace;
for each unique nodes in G do
    create an OWL class as subclass of owl:Thing
end
for each subgraph g in G do
    if g has child nodes then
        create an OWL ObjectProperty class with relation R ;
        assign parent node as domain;
        assign child node(s) as range;
    end
    for each node n in child nodes do
        create OWL DataProperty class;
        assign n as domain ;
        assign data type as range ;
    end
end
Return O, C_m, O_m, D_m;
```

Algorithm 1 (and Algorithm 2) generates (and annotates) an OWL ontology based on the DAG obtained through Bayesian structure learning, in contrast to studies such as [12], which merge the BN into the existing ontology. This is especially advantageous when there is no ontology, which is frequently the case with SMEs. Additionally, this provides benefits, as one can take advantage of semantic technology's benefits without having any prior knowledge of it. Algorithm 1 takes the learned DAG graph G in an adjacency list format as an input. After creating the OWL class as a subclass of the *owl:Thing*, the object property is created, taking into account the connectivity of the nodes in G. In our study, the object property is defined as *isInfluencedByNode*. The *Node* in the object

Algorithm 2: Ontology Annotation

Input: DAG graph G as an adjacency list, OWL ontology O, ontology class mapper C_m, ontology object property mapper O_m, ontology data property mapper D_m
Result: Annotated OWL ontology

```
for each subgraph g in G do
    if g has child nodes then
        for each nodes n in child nodes do
            create an instance i for node n;
            create ObjectProperty restriction mapping O_m to C_m in O for instance i;
            insert value to D_m for instance i;
        end
    end
end
Return Annotated OWL ontology;
```

property *isInfluencedByNode* represents the name of the influencing node (or parent node). Our study consists of the two data properties, namely, *isOriginatedFromNode* and *hasInfluenceFactorOfNode*. *isOriginatedFromNode* is a data property of type xsd:string that contains information about how the relationship was discovered (i.e., based on expert input or learned via structure learning). The *hasInfluenceFactorOfNode* property specifies the degree to which the child node is influenced and is of type xsd:decimal. When the relationship is defined by a domain expert, the *hasInfluenceFactorOfNode* has a weight of 1. Algorithm 2 uses ontology information such as class, object properties and data properties, as well as the ontology itself and the graph G, to annotate the ontology and create the KG.

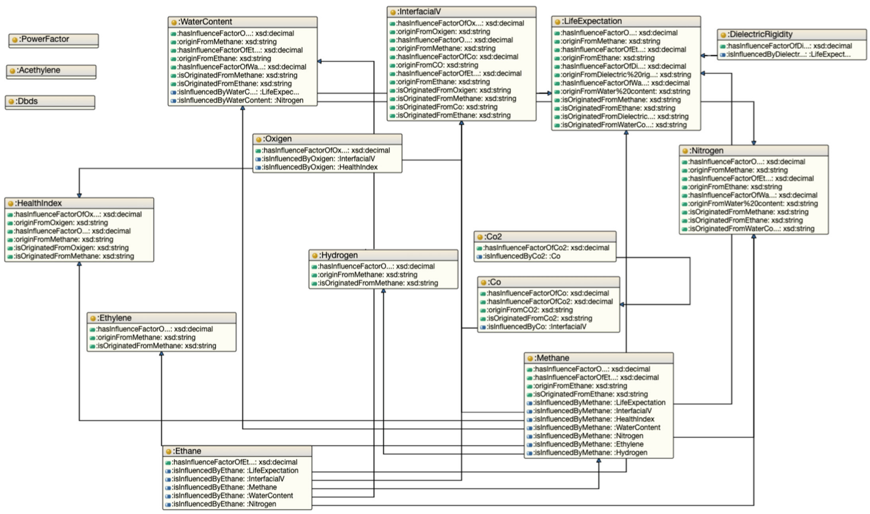
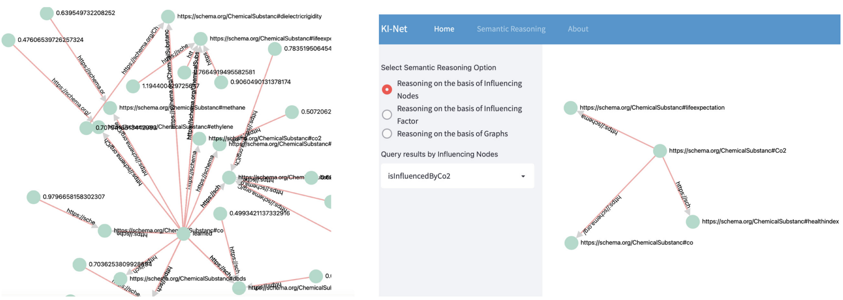


Fig. 2. Automatically generated OWL ontology based on DAG learned from Bayesian structure learning.

Figure 2 shows the ontology generated using Algorithm 1. Each of the algorithms for annotation and ontology generation has a time complexity of $O(n^2)$. The ontology and KG instances that were generated are based on the power transformer dataset [1]. We can see in Fig. 2 (i.e., ontology) that some classes are not connected. The reason for this is, that structured learning was unable to establish a connection between those disjointed classes. This also demonstrates the inherent uncertainty of structure learning.

3.4 Visualisation

The visualisation component provides the user interface for interaction. For example, the visualisation component interactively displays the results of the semantic reasoning, assisting both experts and non-experts in comprehending the variables' relationships. Figure 3 shows the visualisation of the results of the semantic reasoning performed via SPARQL. The semantic reasoning in Fig. 3a shows all the KG instances having an influence factor (or weight) greater than or equal to 0.37 and Fig. 3b shows the nodes that are being influenced by *Co2* nodes. Moreover, the visualisation also provides an interface that allows one to set the hyperparameters, such as DAG filter threshold, L1 and L2 regularisation, for the structure learning.



(a) (b)

Fig. 3. (**a**) Visualisation of semantic reasoning results according to influencing factor (or weights). (**b**) Visualisation of the outcomes of semantic reasoning according to their influencing nodes (or class).

4 Implementation

Python version 3.7 and Streamlit version 1.7.0 were used for the implementation of the proposed work. For the implementation of the Bayesian structure learning algorithms discussed in Sect. 3.2, we use CausalNex [2] version 0.11.0. CausalNex is a Python library for Bayesian Networks and causal reasoning. Owlready2 version 0.37 [9] was used for ontology generation and annotation as discussed in

Sect. 3.3. Similarly, RDFLib version 6.1.1 was used for SPARQL queries in order to interact with the KG, and Streamlit agraph was used for the visualisation if the KG and the DAG. The other libraries scikit-learn version 1.1.1, NetworkX version 2.7.1 and pandas version 1.4.1 were used for handling data such as data preprocessing. The source code is available openly on GitHub[1].

5 Evaluation

When investigating a product failure, investigators would typically want to know the interdependence of the various manufacturing steps, as well as their relationships and effects on other steps. And since our work is focused on minimising product failure, we evaluated our work by evaluating if the generated KG would answer the question that would arise during the failure analysis. Table 1 presents the questions of interest for product failure analysis and their respective answers (i.e., how KG answers the raised questions).

Table 1. Competency questions pertaining to product failure analysis and the corresponding answers.

Questions	Answer
What are the interdependencies between the various manufacturing steps (or what are the interdependencies with step X)?	Interdependencies between various manufacturing steps are answered by the object property *isInfluencedByNode*
How does manufacturing step X influence manufacturing step Y, or what is the effect of manufacturing step X on step Y?	The effect of manufacturing step X on step Y data property *isInfluencedByNode*
How are the interdependencies between the various stages of production determined? Is it based on expert knowledge or independent data-driven learning?	The data property *isOriginatedFromNode* provides an answer to how the interdependencies were deduced

In addition, it is essential that the generated ontology is consistent and error-free. This is due to the fact that inconsistent ontology can lead to problems, such as erroneous inferences. We ran the Hermit[2] reasoner to evaluate the consistency of the generated ontology, which confirmed that the generated ontology had no consistency. The duration of the reasoner was roughly 60 ms.

6 Conclusion and Future Work

In this paper, we presented our work on manufacturing process optimisation using KG and BN, which, to the best of our knowledge, is among the first attempt

[1] Code: https://github.com/tekrajchhetri/ki-net.
[2] http://www.hermit-reasoner.com.

to bridge the gap in the industrial sector's usage of KGs in manufacturing. The use of the KG provides the interoperability, semantics (or meaning) of data and further allows for reasoning, which can be extremely beneficial when analysing failures in sectors such as manufacturing. In addition, the application of KG permits interpretability, a benefit that techniques such as deep learning lack.

Future work would consist of applying the proposed method to other domains or deploying it in industrial environments. In addition, one could extend the work by incorporating additional domain knowledge and applying machine learning to KG for improved results, as we demonstrated in our previous study [5].

Acknowledgements. The research reported in this paper has been funded by European Interreg Austria-Bavaria project KI-Net[3] (grant number: AB292). We would also like to thank Oleksandra Roche-Newton for her assistance in the manuscript preparation and Simon Außerlechner, system engineer at STI Innsbruck, for facilitating servers for experimentation([3] https://ki-net.eu).

References

1. Arias, R.: Data for: Root cause analysis improved with machine learning for failure analysis in power transformers (2020). https://doi.org/10.17632/RZ75W3FKXY.1
2. Beaumont, P., et al.: CausalNex (2021). https://github.com/quantumblacklabs/causalnex. Last Accessed 25 Apr 2022
3. Cao, S., Bryceson, K., Hine, D.: An ontology-based bayesian network modelling for supply chain risk propagation. Indus. Manage. Data Syst. **119**(8), 1691–1711 (2019). https://doi.org/10.1108/IMDS-01-2019-0032
4. Chen, R., Lu, Y., Witherell, P., Simpson, T.W., Kumara, S., Yang, H.: Ontology-driven learning of bayesian network for causal inference and quality assurance in additive manufacturing. IEEE Robot. Autom. Lett. **6**(3), 6032–6038 (2021). https://doi.org/10.1109/LRA.2021.3090020
5. Chhetri, T.R., Kurteva, A., Adigun, J.G., Fensel, A.: Knowledge graph based hard drive failure prediction. Sensors **22**(3) (2022). https://doi.org/10.3390/s22030985
6. Heckerman, D.: A tutorial on learning with bayesian networks (2020). https://doi.org/10.48550/ARXIV.2002.00269
7. Kang, S., Kim, E., Shim, J., Chang, W., Cho, S.: Product failure prediction with missing data. Int. J. Prod. Res. **56**(14), 4849–4859 (2018)
8. Kang, Z., Catal, C., Tekinerdogan, B.: Product failure detection for production lines using a data-driven model. Expert Syst. Appl. **202**, 117398 (2022). https://doi.org/10.1016/j.eswa.2022.117398
9. Lamy, J.B.: Owlready: ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. Artif. Intell. Med. **80**, 11–28 (2017). https://doi.org/10.1016/j.artmed.2017.07.002
10. Riali, I., Fareh, M., Bouarfa, H.: A semantic approach for handling probabilistic knowledge of fuzzy ontologies. In: ICEIS (1), pp. 407–414 (2019)
11. Scanagatta, M., Salmerón, A., Stella, F.: A survey on bayesian network structure learning from data. Prog. Artif. Intell. **8**(4), 425–439 (2019)
12. Setiawan, F.A., Budiardjo, E.K., Wibowo, W.C.: Bynowlife: A novel framework for owl and bayesian network integration. Information **10**(3), 95 (2019). https://doi.org/10.3390/info10030095

13. Zheng, X., Aragam, B., Ravikumar, P., Xing, E.P.: DAGs with NO TEARS: Continuous Optimization for Structure Learning. In: Advances in Neural Information Processing Systems (2018)
14. Zheng, X., Dan, C., Aragam, B., Ravikumar, P., Xing, E.P.: Learning sparse nonparametric DAGs. In: International Conference on Artificial Intelligence and Statistics (2020)