Exploiting Ontology to Build Bayesian Network

Ahmed Mabrouk, Sarra Ben Abbes, Lynda Temal, Ledia Isaj and Philippe Calvez
LAB CSAI ENGIE, 4 Rue Joséphine Baker, 93240 Stains, France

Keywords: Bayesian Network Structure, Ontology, Expert Knowledge, Learning, Dependencies, Renewable Energies.

Abstract: Exploiting experts' domain knowledge represented in the ontology can significantly enhance the quality of the Bayesian network (BN) structure learning. However, in practice, using such information is not a trivial task. In fact, knowledge encompassed in ontologies doesn't share the same semantics as those represented in a BN. To tackle this issue, a large effort has been devoted to create a bridge between both models. But, as far as we know, most state-of-the-art approaches require a Bayesian network-specific ontology for which the BN structure could be easily derived. In this paper, we propose a generic method that allows deriving knowledge from ontology to enhance the learning process of BN. We provide several steps to infer dependencies as well as orientations of some edges between variables. The proposition is implemented and applied to the wind energy domain.

1 INTRODUCTION

Over the last three decades, probabilistic graphical models (PGMs) such as Bayesian networks (Pearl, 1988) are considered as one of the most successful tools for reasoning about beliefs in many real-world applications (cancer diagnosis, robotics, machine diagnosis, etc). Unlike many state-of-the-art predictive models such as random forests, support vector machine (SVM), BN provides a prominent model that enables to represent complex systems using graphical and probabilistic formalisms (easy for humans to understand). The graphical structure encodes a set of dependencies among random variables. The probabilistic part is composed of a set of conditional probability tables to represent uncertainties. Both components are jointly used to perform automated reasoning under uncertainty by enabling to answer efficiently a very wide range of diagnosis queries (Pearl, 2014).

Ontology, also represented by a graphical model, is well known to be the best way to represent knowledge in a domain of discourse according to different points of view and purposes. It allows representing explicitly and formally existing entities in an application domain. Ontology formalization can use Description Logic (DL) language, which is based on First-order logic to describe concepts, relationships, and constraints. It then enables us to make inferences.

Both models appear to be useful to perform reasoning and domain knowledge explanation using different paradigms. Intuitively, we believe that combining BNs and ontologies gives rise to provide high expressiveness and more reliable reasoning under uncertainty. In this sense, several approaches have been proposed to create a bridge between both models. Basically, they can be divided into three categories: (i) introducing additional notations to represent probabilistic values in the ontology (Yang and Calmet, 2005; Zhang et al., 2009; Carvalho et al., 2010; Mohammed et al., 2016), (ii) exploiting the semantic richness of the ontology to guide the BN learning (Fenz, 2012; Jeon and Ko, 2007; Ishak et al., 2011a; Messaoud et al., 2013), and (iii) using BN results to enrich the ontology (Ishak et al., 2011b; Wang, 2007). In this paper, we are interested in exploiting the knowledge in the ontology within the BN structure learning. Unlike classical methods which, naively rely on ontological relationships to identify dependencies between variables, our approach exploits in an intelligent way existing knowledge to deduce more efficient dependencies in the BN. To do so, we propose a generic method that allows deriving key knowledge from ontology. This method is based on a set of rules allowing to infer dependencies and edges orientation in an original way. The rest of this paper is organized as follows. In section 2 we give the main definitions of BN and ontology models. In section 3, we discuss the most common state-of-the-art approaches dedicated to exploit ontologies during the BN learning phase. Then, in section 4, we describe our new approach and justify its correctness within the wind turbine domain. Its efficiency is highlighted through experiments in section 5. Finally, a conclusion and some future works are given in section 6.

Mabrouk, A., Ben Abbes, S., Temal, L., Isaj, L. and Calvez, P.
Exploiting Ontology to Build Bayesian Network.
DOI: 10.5220/0010840400003122
In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)*, pages 578-585
ISBN: 978-989-758-549-4; ISSN: 2184-4313

2 BACKGROUND

In this section, we present the main definitions and semantics of BN and ontology models.

2.1 Bayesian Network

Definition 1 (Bayesian network)**.** *A BN is a pair* $(\boldsymbol{G}, \Theta)$ *where* $\boldsymbol{G} = (\mathbf{V}, E_{\boldsymbol{G}})$ *is a directed acyclic graph (DAG),* $\mathbf{V}$ *represents a set of random variables*[1]*,* $E_{\boldsymbol{G}}$ *is a set of arcs, and* $\Theta = \{\theta_{V_i|\mathbf{Pa}(V_i)}\}_{V_i \in \mathbf{V}}$ *is the set of conditional probability tables (CPTs) of the nodes / random variables* V_i *in* $\boldsymbol{G}$ *given their parents* $\mathbf{Pa}(V_i)$*, i.e.* $\theta_{V_i|\mathbf{Pa}(V_i)} = P(V_i|\mathbf{Pa}(V_i))$*. The BN encodes the joint probability over* $\mathbf{V}$ *as:*

$$P(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} P(V_i|\mathbf{Pa}(V_i)) \qquad (1)$$

Eq.1 is also called the chain rule or the general product rule. BN provides a mechanism for exploiting structure in high-dimensional joint distributions to describe them compactly, and in a way that allows them to be constructed and utilized effectively. By their graphical structure, BNs encode an independence model, i.e., a set of conditional independences between random variables, characterized by the *d-separation* property:

Definition 2 (*d-separation*)**.** *Two nodes* V_i *and* V_j *are said to be d-separated in* $\boldsymbol{G}$ *by a set of nodes* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{V_i, V_j\}$*, which is denoted by* $V_i \perp_{\boldsymbol{G}} V_j | \mathbf{Z}$*, if, for every trail (directed path) linking* V_i *and* V_j *in* $\boldsymbol{G}$*, there exists a node* S *on the trail such that one of the following conditions holds:*

1. *S has converging arrows on the trail and neither S nor any of its descendants are in* $\mathbf{Z}$*;*
2. *S does not have converging arrows on the trail and* $S \in \mathbf{Z}$*.*

In other words, any independence reported by *d-separation* is satisfied by the underlying distribution. When two variables V_i and V_j are not *d-separated* by $\mathbf{Z} \subset \mathbf{V}$, they are said to be *d-connected*. With these definitions, relationships encoded in a BN can be seen as a flow in the graph. The *d-separation* tells us when influence from V_i can “flow” through $\mathbf{Z}$ to affect our beliefs about V_j.

2.2 Ontology

Ontologies are complex artifacts (composed of concepts, hierarchical relations, and roles) that are built according to different points of view and purposes. The most widely cited definition of an ontology is a formal, explicit specification of a shared conceptualization, used to help humans and machines to share common knowledge (Gruber, 1993; Guarino, 1995). This definition highlights the following characteristics of an ontology:

- **Conceptualization:** defines the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them (Genesereth and Nilsson, 1987). A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose (Gruber, 1993) .
- **Explicit:** corresponds to the precise definition of the concepts and the constraints of their use.
- **Formal:** refers to the fact that the expressions must be machine-readable.
- **Shared:** refers to a common understanding of domain knowledge among people or agents.

An ontology is formally defined by $\mathbf{O} = (\mathbf{C}, \leqslant_{\mathbf{C}}, \mathbf{R}, \leqslant_{\mathbf{R}}, \mathbf{A})$

- $\mathbf{C}$ and $\mathbf{R}$ are two disjoint sets whose elements are respectively called *Concepts* (e.g. *Room, Building*) and *Relations* (e.g.*part-of, sub-Zone-of*).
- $\leqslant_{\mathbf{C}}$ is a partial order over $\mathbf{C}$, called concept hierarchy or taxonomy *(is-a)*
- $\leqslant_{\mathbf{R}}$ is a partial order over $\mathbf{R}$, called a hierarchy of relations (e.g.*sub-Zone-Of* $\leqslant_R$ *part-Of*)
- $\mathbf{A}$ a set of axioms and inferences rules (e.g. *Physical Entity part-of Only Physical Entity*)

3 RELATED WORK

In our paper, we shall focus our discussion on the use of ontology to construct a BN structure. In this context, several approaches have been developed for assisting in eliciting knowledge from ontology and then derive a BN (Bucci et al., 2011; Helsper and Van Der Gaag, 2002). In (Devitt et al., 2006), authors expose a direct correspondence between BN formalism and the original domain ontology: (i) concepts $\rightarrow$ nodes, (ii) concepts attributes $\rightarrow$ CPTs, and (iii) inheritance relations $\rightarrow$ arcs. The main drawback of this approach is that it requires BN-specific ontology extensions. (Fenz, 2012) proposed a semi-automatic approach composed of the following steps: (i) selection of relevant classes, individuals, and properties (ii) creation of the BN structure (iii) construction of the CPTs, and (iv) incorporation of existing knowledge

[1]By abuse of notation, we use interchangeably $V_i \in \mathbf{V}$ to denote a node in the BN or the voltage of bus i

facts. In this work, the BN is built from a security ontology describing threat, vulnerability, and control dependencies. The author extends the basic idea of (Devitt et al., 2006) to a more generalized framework by introducing the following analogies: (i) axioms $\rightarrow$ BN nodes scales and weights, and (ii) instances $\rightarrow$ findings. To generate a BN, (Fenz, 2012) developed a Protégé plug-in called Bayesian Network Tab (BNTab)[2]. A key issue in the proposed approach is that it works only with boolean (or binary) variables, so the application scope is limited to restricted real-world domains. Moreover, this approach doesn't take into account the meaning of the ontology relations to identify the more significant ones that fit better with the dependency semantic in the BN. In (Jeon and Ko, 2007), a semi-automatic approach of BN construction for diagnosing diseases in the e-health domain is proposed. This method starts by generating nodes in a BN based on a set of selected concepts from the e-health ontology. Then, developers identified valid links among nodes in the BN based on a meta-model that represents cause-and-effect relationships among ontologies. This approach demonstrates several issues. First, it requires the presence of causal relationships between ontologies which are not always tractable in practice. Second, the links generating step ends up with a complex BN structure in the form of a hierarchical graph. Such topology is not suitable for all applications. In (Messaoud et al., 2013), authors present SemCaDo (Semantical Causal Discovery) algorithm for integrating ontological knowledge for learning causal BNs. The SemCaDo algorithm takes an observational dataset and a corresponding ontology as inputs. In the first phase, causal relations are extracted from the ontology and then integrated in the form of constraints (white lists) during BN structure learning. Then, the second phase optimizes the orientation of the remaining undirected edges in the complete partially directed graph (CPDAG) using a semantic distance calculus provided by the ontology. Finally, the algorithm re-iterates over the second phase if there are still some non-directed edges in the graph. All discovered causal links will be introduced as semantic causal relations between the corresponding ontology concepts. BN and OWL Integration Framework (ByNowLife) (Setiawan et al., 2019) is a framework proposed to integrate BN with OWL by providing an interface for probabilistic reasoning information through SPARQL queries. This approach consists of transforming logical information contained in an ontology into a BN and vice-versa. This framework is composed of three main parts: (1) application, which allows querying the Knowledge Base in SPARQL format for logical reasoning and *hasProbValueOf* special property for probabilistic reasoning, (2) reasoner, which involves two components; logical reasoner and probabilistic reasoner, and (3) knowledge base, contains the domain ontology and BN knowledge. The main drawback of the two last approaches is that causal relations in the ontology are not always available. In addition, semantic characteristics of ontological relations are not taken into consideration during BN construction.

[2]https://protegewiki.stanford.edu/wiki/Bayesian_Network_Tab_(BNTab)_1.1.3

4 PROPOSED APPROACH

In this section, we describe a new approach that exploits in an efficient way the knowledge capitalized in ontologies to enhance BN structure learning results. We justify the correctness of our approach based on the wind turbine example.

4.1 Problem Formalization

Although BNs are seldom constructed entirely by experts, the knowledge given by these latter can still prove to be a useful source of information that should be exploited by BN learning algorithms. In this context, the ontology can be particularly helpful since it represents explicit knowledge. While the complementarity between ontology and BN mights seem trivial in general, it turns out that the joint use of both models can be subject to many issues. In fact, relations between concepts do not necessarily encode a dependence/independence between concepts. Therefore, generate automatically a BN graphical structure from a given ontology, appears completely intractable. To tackle this issue, we propose a semi-automatic solution to facilitate knowledge elicitation from ontology and knowledge-guided BN structure learning. The general pipeline of the proposed approach is summarized as follows:

1. identify a set of relations that encapsulate a dependence property between concepts,
2. compute dependence scores between concepts based on selected relationships,
3. build an initial BN skeleton[3] based on the computed scores,
4. orient some edges in the skeleton using the semantic of selected relationships in the ontology,

[3]a skeleton of the BN is the DAG where arcs are substituted by (undirected) edges.

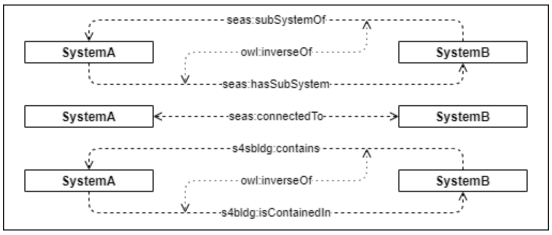

Figure 1: Selected Relations used in the Wind Turbine Ontology.

5. refine the resulted BN through the use of observational data and a score-based approach.

In the following, we discuss in greater detail each of the previous steps. To facilitate understanding, we apply our approach on the wind turbine domain.

4.2 Identification of Key Relationships

As discussed previously, the BN model allows representing conditional independencies as a DAG. In fact, conditional independencies could be directly inferred from the BN graph via the *d-separation* criterion. In this context, we need to extract relevant relations from the ontology that provide novel insights about variable interactions (dependence, causality, etc.). To identify these relations called $\mathbf{R}_{dep} \subseteq \mathbf{R}$, it is quite natural to ask questions to one or multiple experts, following this format: *Given a relation $R_i \in \mathbf{O}$, is our belief about the state of the concept C_i (resp. C_j) is influenced when we have an observation regarding the concept C_j (resp. C_i)*? The binary decision (yes/no) returned by the expert for this question indicates whether the relationship should be considered to capture dependencies between variables. After the accumulation of all of these binary answers, we end up the first step of our approach with the selected relations shown in figure1. As can be easily seen, the selected relationships focus mainly on the location (*s4bldg:contains* (resp. *s4bldg:isContainedIn*)) as well as the set of connections between the different components (*seas:connectedTo*, *seas:subSystemOf* (resp. *seas:hasSubSystem*)) [4]. According to the expert, the lower the semantic distance between two components (or concepts in the ontology) in the system, the greater dependent they are. For example, since the concept *Main Bearing* is connected to the concept *Hub* in the ontology, it is likely that, if the temperature of the *Main Bearing* is high, the temperature of the *Hub* will get higher as well, or vice-versa. The same idea applies to the other selected relations.

[4]Note that *subSystemOf* of can inferred from *hasSubSystem* if it is not explicitly mentioned in **O** Same thing for the relation *contains* according to *iscontainedIn*

In this step, SPARQL queries were used to extract concepts linked by the selected relationships from **O** as described in the algorithm1.

Algorithm 1: Dependency triples extraction.

Input: Ontology **O**, Selected relations $\mathbf{R}_{dep}$
Output: Set **T** of extracted triples
for R_k in $\mathbf{R}_{dep}$ **do**
    Perform SPARQL query:
    SELECT ??C_i ?C_j WHERE {
        ?C_i rdfs:subClassOf ?S .
        ?S a owl:Restriction; owl:onProperty R_k;
        owl:someValuesFrom || owl:allValuesFrom ?C_j.
        }
    Add $< C_i, R_k, C_j >$ to **T**
end for
return T

4.3 Dependence Scores Calculation and BN Skeleton Building

In this section, we explain how the extracted information from **O** can be exploited during the different BN construction steps. Given the extracted triples **T**, we merely generate a graph **G**, where edges are weighted by the strength of dependency w_k that every relation R_k represents, i.e., an undirected edge in **G** is inserted between nodes V_i (corresponds to C_i) and V_j (corresponds to C_j)[5] if $< C_i, R_k, C_j > \in \mathbf{T}$ or $< C_j, R_k, C_i > \in \mathbf{T}$. In the example of wind turbine, the selected relations were ranked by the experts according to the dependency semantic, as follows: the relation *contains*[6] (resp. *isContainedIn*) transcribes a stronger dependency between concepts, as certain components are contained in another one, followed by *subSystemOf* (resp. *hasSubSystem*), and finally *connectedTo*. For these relations, the assigned dependency weights are: $w_{contains}$ (resp. $w_{isContainedIn}$) $\leftarrow 1$, $w_{subSystemOf}$ (resp. $w_{hasSubSystem}$) $\leftarrow 1.2$, and $w_{connectedTo} \leftarrow 1.4$. Figure 2 depicts an example of a weighted graph resulting from the ontology shown in figure3. Based on the resulted graph, we compute the dependence between any two nodes V_i and V_j in **G** (denoted by $dep(V_i,V_j)$) as:

$$dep(V_i,V_j) = \frac{1}{shortPath_{\mathbf{G}}(V_i,V_j)} \tag{2}$$

The shortest path between nodes V_i and V_j ($shortPath_{\mathbf{G}}(V_i,V_j)$) is calculated using the weighted

[5]We use V_i to denote nodes in the weighted graph rather than C_i in order to be consistent with BN notations

[6]For simplicity, in the rest of the paper, we remove the prefix part from all concepts and relationships.

Dijkstra's algorithm(Dijkstra et al., 1959). From the given results, we deduce dependency strengths between all pairs of nodes in **G**. For instance, the dependence strength between *Main Bearing* and *Generator-Bearing* ($dep(MB, GBe)$) is equal to 0.29 (see Fig.2). The obtained results were approved by the wind turbine experts. Using dependency scores, we can therefore build an initial skeleton of the BN. Remind that edges in the BN represent probabilistic dependencies (or correlations) between nodes. In this case, an edge E_{V_i,V_j} is inserted in the BN if and only if $dep(V_i,V_j)$ is greater than some defined threshold k:

$$E_{V_i,V_j} = \begin{cases} Accept & \text{if } dep(V_i,V_j) \geq k \\ Reject & \text{else.} \end{cases}$$

With all these rules, our approach ends up this first step by constructing an initial skeleton of the BN using dependencies information derived from the ontology. In this phase, the expert domain can check and then correct the automatically detected edges between variables if needed.

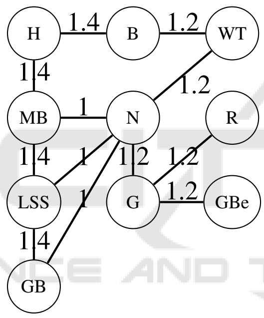

Figure 2: Extracted weighted graph from ontology using the selected relations where H: Hub, B: Blade, WT: Wind Turbine, N: Nacelle, R: Rotor, GBe: Generator bearing, G: Generator, MB: Main Bearing, LSS: Low Speed Shaft, and GB: GearBox.

In the next section, we will describe how we can determine the orientations of some edges and how we exploit such information to make the BN learning, more reliable.

4.4 Knowledge-guided BN Structure Learning

We start by explaining how certain edges in the skeleton are oriented based on the semantic of relationships in **O**. Then, we use a score-based algorithm to refine the BN. For example, if we consider the dependency semantic encapsulated in *subSystemOf* and *contains* relations, an ordering $\prec$ over the associated variables (corresponding to concepts), can be inferred. The relation *subSystemOf* is used to represent the composition of different systems. These systems can be divided into two categories: mechanical and electrical sub-systems. According to the wind turbine ontology, the mechanical component has the following sub-systems: wind turbine blade, the rotor, the nacelle, etc. Regarding the electrical component, it is composed of the generator and the power electronic converter, etc.. Both sub-systems are also composed of a set of smaller sub-systems. Thus, the *subSystemOf* can be considered as a relationship linking a component to smaller ones, which makes it usable *per se* as a prior about corresponding variables (or concepts) ordering. To illustrate this idea, let us consider an extract of the wind turbine ontology as shown in figure3. Based on the *subSystemOf* property, the following ordering can be deduced:

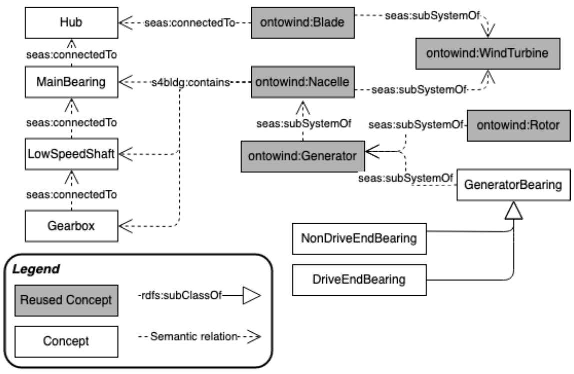

Figure 3: Extract of Wind Turbine ontology.

$$Nacelle \prec Generator \prec Generator\text{-}bearing \prec \ldots$$

Intuitively, the ordering $\prec$ over variables is very useful to define the list of parents for some variables in the constructed skeleton. For this purpose, a node V_i is set as a parent of node V_j (i.e., $V_i \rightarrow V_j$) only when $V_i \prec V_j$ and they are connected in the skeleton. It should be emphasized that the variables ordering deduced from the relation *subSystemOf* is acyclic since a variable V_i cannot be at the same time a *subSystemOf* and *hasSubSystem* with the same variable V_j. According to the expert, this approach is quite reasonable in practice since the impact is generally more important from the global component to the more specific one. For instance, a high temperature of the nacelle implies that the temperature of all its subsystems are also high. The opposite scenario does not necessarily hold in practice. The relation *contains* depicts the set of mechanical/electronic elements that a component may contain. Thus, an ordering over variables that satisfies the position of each component w.r.t. the others can be deduced. Note that both relations *subSystemOf* and *contains* (resp. *hasSubSystem* and *isContainedIn*) may be linked to the same concepts. Importantly, these ordering results do not contradict each other, because their semantics are very close. If two concepts C_i and C_j are linked by *contains* and *subSystemOf*, the following properties are satisfied:

$$< C_i, subSystemOf, C_j > \rightarrow \nexists < C_j, contains, C_i > \tag{3}$$

$$< C_i, contains, C_j > \rightarrow \nexists < C_i, subSystemOf, C_j > \tag{4}$$

With all these properties, our approach converts the skeleton **G** into a partially directed graph (PDAG) using Algorithm2. For the remaining undirected edges, we select arbitrarily orientations. At this point, our algorithm has constructed an initial DAG of the BN. The obtained DAG is then refined through a score-based search algorithm. In our wind turbine case, we used a SCADA data to learn the final BN. During the search phase, we pick the resulted DAG from the ontology as a starting point and we compute its score. After that, we consider all the neighbor graphs of **G** in the search space — all of the legal (without cycle) networks obtained by applying a single operator (edge deletion, edge addition, or edge reversal) to **G** — and compute the score for each of them given the dataset. We then consider the change that leads to the best improvement. We continue this process until no modification improves the score. During this refinement process, there are two ways to consider the knowledge derived from the ontology:

- **scenario 1**: enable the learning algorithm to modify the initial DAG deduced from **O**.
- **scenario 2**: consider the ontology-based oriented edges as constraints that we want to enforce, i.e., the algorithm is not allowed to modify these arcs.

Algorithm 2: Construct an initial PDAG-BN.

Input: BN skeleton **G**, Ontology **O**
Output: PDAG-BN **G**
$\prec_1 = order(\mathbf{O}, subSystemOf)$
$\prec_2 = order(\mathbf{O}, contains)$
$\prec = \{\prec_1 \cup \prec_2\}$
for each E_{V_i,V_j} in $E_{\mathbf{G}}$ **do**
  if $V_i \prec V_j$ **then**
    $E_{\mathbf{G}} \leftarrow (V_i \rightarrow V_j)$
  end if
  if $V_j \prec V_i$ **then**
    $E_{\mathbf{G}} \leftarrow (V_j \rightarrow V_i)$
  end if
end for
return G

5 EXPERIMENTAL RESULTS

This section is dedicated to present the use-case and highlight the impact of ontology's knowledge in the BN learning process.

5.1 Wind Turbine Use Case

In this paper, we evaluate our approach using the data from supervisory control and data acquisition (SCADA) of wind turbines. This data represents a cost-effective way to monitor wind turbine components for early failures and performance issues. In our experiment, we used the SCADA data coming from La Haute Borne wind farm between 2013 and 2016. This data is composed by a massive amounts of time-series that are stored, manipulated, and filtered via the DARWIN Platform of ENGIE[7]. Overall, 210095 records have been generated during this period. In this section, we focus our attention on the study of the main components of the wind turbine such as nacelle, generator, gearbox, etc[8]. To build the BN, we also used the wind turbine ontology (see an extract in Fig.3), designed within H2020 European project. For each concerned concept (nacelle, generator, gearbox, etc.), we extract all the concepts that are related to it with the selected relations as described in our approach (*subSystemOf* (resp. *hasSubSystem*), *contains* (resp. *isContainedIn*), and *connectedTo*).
In the following, both knowledge resources (SCADA data and ontology) are exploited to derive a BN model representing wind turbine system.

5.2 Evaluation of the Approach

We evaluate the performance of our approach w.r.t. the quality of the detected relationships for a set of selected temperature variables (gearbox and nacelle). For each record in the data, we predict the value of the temperature classes given the observations about the explanatory variables in the BN. The evaluation of the BN's classification task is been carried out using the 10-fold cross-validation procedure and the expected loss estimation. We studied the impact of the ontology knowledge integration on the predictions quality and the computation time. To make the comparison evaluation more thorough, in our experiments, we vary the following parameters:

- the size of the dataset,
- the dependence threshold k,

[7]https://digital.engie.com/solutions/darwin

[8]By abuse of notation, we use the word Gear and Gen to denote respectively gearbox and generator.

- the scenario of the score-based setting (see the end of section 4.4).

All experiments are performed on a 1.9GHz Intel Core i7 computer with 16GB of memory running Windows 10. Our approach has been implemented using the bnlearn R package (Scutari, 2009) and Protégé tool[9]. We start by building the initial BN DAG using the information in the ontology. Note that we have done the matching between SCADA temperature variables and the ontological entities (concepts/relations). For example: the generator temperature, which is a variable in SCADA, is represented in the ontology by this triple *<Generator, hasTemperature, Temperature>*. Table 1 mentions an extract of detected dependencies between SCADA variables using the ontology. In these results, we observe a strong dependency between temperatures of nacelle and gearbox oil sump (equal to 1). A high dependency is also observed between rotor bearing and nacelle temperatures.

Table 1: An extract of dependency computed scores.

Var1 name	Var2 name	Score
Nacelle_temp	Gear_oil_temp	1
Gear_inlet_temp	Gear_oil_temp	1
Rotor_bear_temp	Nacelle_temp	1
Converter_torque	Torque	0.83
Gear_bear_1_temp	Gear_oil_temp	0.83
Gear_bear_2_temp	Gear_oil_temp	0.83
Hub_temp	Rotor_bear_temp	0.71
Gear_inlet_temp	Nacelle_temp	0.5
Rotor_bear_temp	Gear_oil_temp	0.5
Gear_inlet_temp	Gear_bear_1_temp	0.45
Gear_inlet_temp	Gear_bear_2_temp	0.45
Nacelle_temp	Gear_bear_1_temp	0.45
Nacelle_temp	Gear_bear_2_temp	0.45
Gen_stator_temp	Gen_bear_1_temp	0.42
Gen_stator_temp	Gen_bear_2_temp	0.42
Gen_stator_temp	Nacelle_temp	0.42
Gen_bear_1_temp	Nacelle_temp	0.42

Figure 4 displays the expected loss for the gearbox inlet temperature prediction using models learned with the hill-climbing algorithm (HC) alone, and our approach without and with constraints, respectively denoted by sc=1 and sc=2. We fixed the threshold $k = 0.2$. As can be observed, our approach always outperforms the simple HC method, whatever the size of the dataset. For a dataset size equal to 10000, the expected loss of our approach with scenarios 1 and 2 is actually lower than that of the HC by $\sim 6\%$ and $\sim 5\%$. These results can be explained by the fact that our approach rules enable us to detect key dependencies with gearbox inlet temperature variable, hence obtaining better predictions.

[9]https://protege.stanford.edu/

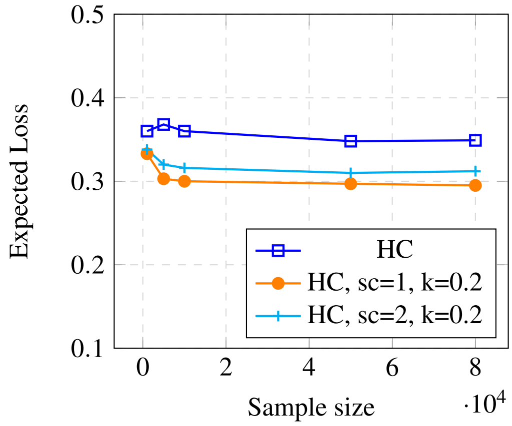

Figure 4: Gearbox inlet temperature prediction errors.

Figure 5 shows results for nacelle temperature predictions. As can be seen, our approach using constraints got better results than the simple use of HC algorithm. Our approach with scenario 1 (sc=1) is always outperformed by the simple use of HC, in this case. These results happened because we enable the refinement score-based algorithm to modify our initial DAG (derived mostly from the ontology), which may lead to loose some key relationships.

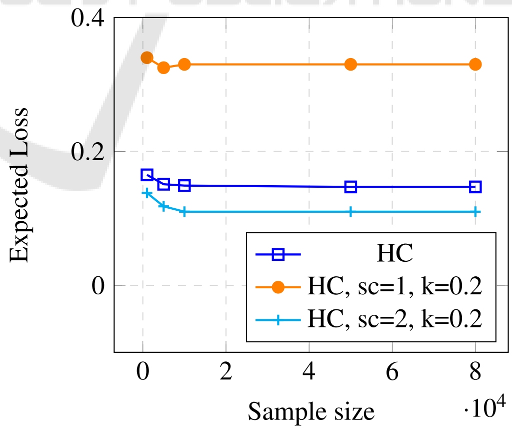

Figure 5: Nacelle temperature prediction errors.

Figure 6 depicts runtimes associated with each algorithm according to the data size parameter. As expected, except for the case where the data size is equal to 10000, our approach (sc=1 and sc=2) is mostly faster than HC. This is explained by the use of initial knowledge derived from the ontology which optimizes the DAG search space. For data size equal to 80000, our approach with scenarios 1 and 2 are respectively about ∼19% and ∼16% faster than the only use of HC.

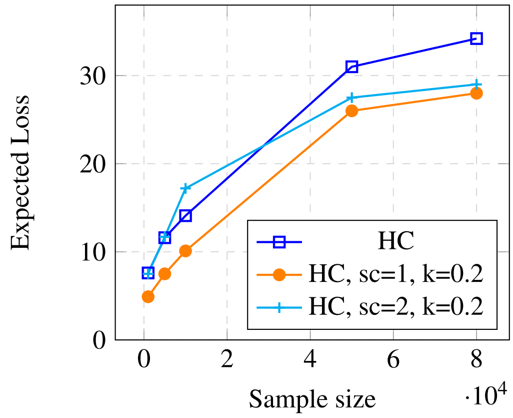

Figure 6: Runtime w.r.t. the data size.

6 CONCLUSION AND FUTURE WORKS

In this paper, we proposed a new generic method for the BN learning structure by exploiting the knowledge in the ontology. Our method is divided into several steps: (i) constructing a weighted graph from the ontology (ii) deriving dependencies and constructing the BN skeleton (iii) orienting edges using the semantic of relations in the ontology (iv) refining the BN via a scored based algorithm. By integrating knowledge from the ontology, our approach leads to a significant improvement in terms of runtime computations and expected loss results. The proposed approach can be easily applied to other domains. As future works, our current approach can be extended by integrating additional knowledge from ontologies and also by refining these latter through the use of BN results (adding relationships, uncertainty, etc.). Additional experimentations on more complex real-world datasets will be also explored.

REFERENCES

Bucci, G., Sandrucci, V., and Vicario, E. (2011). Ontologies and bayesian networks in medical diagnosis. In *2011 44th Hawaii International Conference on System Sciences*, pages 1–8. IEEE.

Carvalho, R. N., Laskey, K. B., and Costa, P. C. (2010). Pr-owl 2.0–bridging the gap to owl semantics. *Uncertainty reasoning for the semantic web II*, pages 1–18.

Devitt, A., Danev, B., and Matusikova, K. (2006). Constructing bayesian networks automatically using ontologies. *Applied Ontology*, 1.

Dijkstra, E. W. et al. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Fenz, S. (2012). An ontology-based approach for constructing bayesian networks. *Data & Knowledge Engineering*, 73:73–88.

Genesereth, M. R. and Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. San Francisco, CA, USA.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220.

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal Human Computer Study*, 43(5-6):625–640.

Helsper, E. M. and Van Der Gaag, L. C. (2002). Building bayesian networks through ontologies. In *ECAI*, volume 2002, page 15th.

Ishak, M. B., Leray, P., and Amor, N. B. (2011a). Ontology-based generation of object oriented bayesian networks. In *BMAW 2011*, pages 9–17.

Ishak, M. B., Leray, P., and Amor, N. B. (2011b). A two-way approach for probabilistic graphical models structure learning and ontology enrichment. In *KEOD 2011*, pages 189–194.

Jeon, B.-J. and Ko, I.-Y. (2007). Ontology-based semi-automatic construction of bayesian network models for diagnosing diseases in e-health applications. In *2007 Frontiers in the Convergence of Bioscience and Information Technologies*, pages 595–602. IEEE.

Messaoud, M. B., Leray, P., and Amor, N. B. (2013). Active learning of causal bayesian networks using ontologies: a case study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Mohammed, A.-W., Xu, Y., and Liu, M. (2016). Knowledge-oriented semantics modelling towards uncertainty reasoning. *SpringerPlus*, 5(1):1–27.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems : networks of plausible inference / Judea Pearl.* Morgan Kaufmann Publishers San Mateo, Calif.

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Scutari, M. (2009). Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*.

Setiawan, F. A., Budiardjo, E. K., and Wibowo, W. C. (2019). Bynowlife: a novel framework for owl and bayesian network integration. *Information*, 10(3):95.

Wang, Y. (2007). Integrating uncertainty into ontology mapping. In *The Semantic Web*, pages 961–965. Springer.

Yang, Y. and Calmet, J. (2005). Ontobayes: An ontology-driven uncertainty model. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 1, pages 457–463. IEEE.

Zhang, S., Sun, Y., Peng, Y., Wang, X., et al. (2009). Bayesowl: A prototype system for uncertainty in semantic web. In *Proceedings of the International Conference on Artificial Intelligence*.