

Semantic Association Rule Learning from Time Series Data and Knowledge Graphs

Erkan Karabulut¹, Victoria Degeler¹ and Paul Groth¹

¹University of Amsterdam, Science Park 904, Amsterdam, 1098 XH, North Holland, The Netherlands

Abstract

Digital Twins (DT) are a promising concept in cyber-physical systems research due to their advanced features including monitoring and automated reasoning. Semantic technologies such as Knowledge Graphs (KG) are recently being utilized in DTs especially for information modelling. Building on this move, this paper proposes a pipeline for semantic association rule learning in DTs using KGs and time series data. In addition to this initial pipeline, we also propose new semantic association rule criterion. The approach is evaluated on an industrial water network scenario. Initial evaluation shows that the proposed approach is able to learn a high number of association rules with semantic information which are more generalizable. The paper aims to set a foundation for further work on using semantic association rule learning especially in the context of industrial applications.

Keywords

rule learning, knowledge graph, time series data, digital twin, internet of things,

1. Introduction

Learning rules and patterns from data is one of the core sub-fields in data analysis and machine learning. Association Rule Mining (ARM) is one specific task in rule learning where the goal is to learn association rules between variables which describe how two or more variables are associated to each other. In an industrial Internet of Things (IoT) setting, ARM methods are used to learn rules from time series sensor data [1, 2].

Association rules are in the form of $X \rightarrow Y$, which means if X holds, then Y also holds where X and Y can be a single or multiple variables with a truth value. In order to apply ARM methods to numerical data, many approaches have been proposed including discretization, optimization and statistical approaches, under the name Numerical ARM or Quantitative ARM [3]. As an example, for time series data produced by IoT devices, a simple association rule based on a statistical approach can be in the form of $\text{mean}(m_t^X) \rightarrow \text{mean}(n_t^Y)$, which is interpreted as “in a time frame t , if mean measurement from sensor X is m , then the mean value of the measurements of sensor Y must be n ”. Discretization methods refer to discretizing numerical data before running an ARM algorithm. Optimization methods refer to evolutionary algorithms where rule quality criteria are generally used as fitness functions to learn rules in a desired format [3].


SemIIM'23: 2nd International Workshop on Semantic Industrial Information Modelling, 7th November 2023, Athens, Greece, co-located with 22nd International Semantic Web Conference (ISWC 2023)

*Corresponding author: Erkan Karabulut

✉ e.karabulut@uva.nl (E. Karabulut); v.o.degeler@uva.nl (V. Degeler); p.t.groth@uva.nl (P. Groth)

🆔 0000-0003-2710-7951 (E. Karabulut); 0000-0001-7054-3770 (V. Degeler); 0000-0003-0183-6910 (P. Groth)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

We hypothesize that for numerical data produced in industrial IoT networks, incorporating semantics of the system components in rule learning can be beneficial including discovering previously unknown relations, e.g. higher number of rules, and helping to generalize association rules of the above form. This hypothesis is tested in a specific type of IoT scenario, a Digital Twin (DT). DTs have many different proposed definitions over the past two decades [4]. The main goal is to create a precise representation (*‘twin’*) of a physical system, often referred as Physical Twin (PT), in a digital environment and to maintain a bi-directional communication in between them. Recently, semantic technologies such as ontologies and Knowledge Graph (KG)s, started to be used in DTs, for system/data modeling, establishing semantic interoperability, extracting semantic relations and/or facilitating reasoning processes [5].

To the best of our knowledge, at the time of writing, there is no approach for learning rules containing semantic information as well as time series data in a DT. This study aims to fill this gap by proposing a first semantic rule learning approach utilizing KGs, based on the well-known FP-Growth [6] algorithm. Concretely, the contributions of this paper are as follows:

- Describing a full pipeline of operations that consists of: i) KG construction in DTs, ii) semantic association rule learning based on the KG and time series data, and iii) making inferences based on the learned rules (Section 2).
- A first approach (Naive SemRL) that extends FP-Growth algorithm to learn rules containing semantic information from KGs and time-series data (Section 3).
- A semantic rule quality measure in order to evaluate rules generated by semantic rule learning algorithms (Section 4).

The proposed approach is evaluated in an industrial use case, water networks, which refer to water distribution systems that bring water to consumers, i.e. apartments, industrial sites. This is an ongoing research with many open issues and research questions emphasized in Section 5.

2. Semantic Rule Learning and Inference Pipeline

This section first motivates utilization of semantic association rule learning techniques in DTs, and then describes a pipeline of operations.

In a best case scenario, a DT has the full knowledge of its PT. In this situation, we say that the PT is 100% *“twinned”*, or the *“twinning ratio”* is 100%. Low twinning ratio intuitively might affect the performance of any reasoning or learning algorithm running in the DT, e.g. too many missing values. A major reason that can cause low twinning ratio is to have *discrepancies* in between PT and its DT. A *discrepancy* refers to a state or attribute of a PT component that has incorrect or inaccurate representation in its DT. For instance, in a water network scenario, an undetected leakage in a pipe is considered a discrepancy. Instead of using separate solutions for each such issue, this study proposes a discrepancy detection method that is a generalization over any such discrepancy. The proposed approach consists of a pipeline (Figure 1) of three operations: i) KG construction, ii) semantic rule learning, iii) making inferences.

Knowledge Graph Construction in DTs. KGs are already being used for information modeling in DTs [7]. We hypothesize that DTs with high dependency among its sub-components, e.g. DT of a water network, can benefit from KGs, not only in information modeling but also in rule learning and making inferences. KGs for DTs can be constructed using a top-down approach

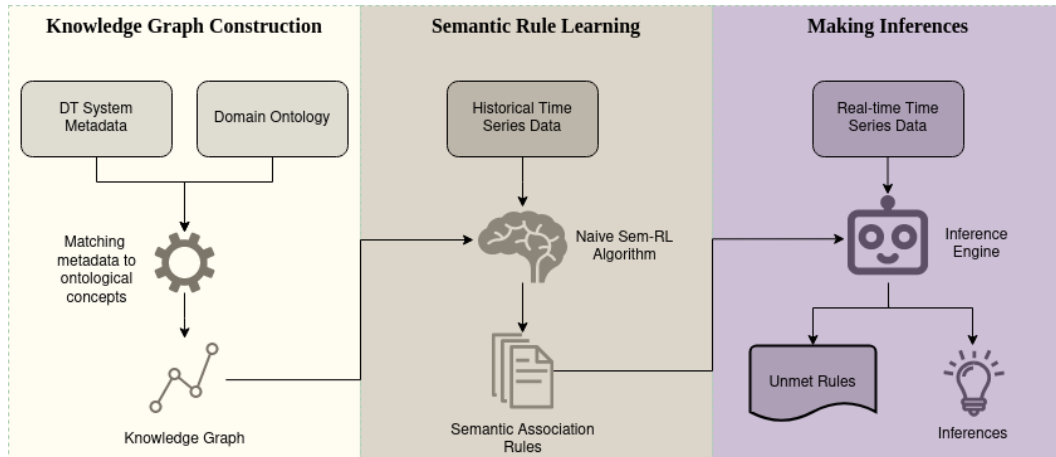


Figure 1: Semantic association rule learning and inference pipeline.

from DT metadata, and a domain ontology that shows how the components are related to each other [8]. Types of entities in DTs are never obscure, meaning that when a representation for a physical object is created in the digital environment, e.g. a water pipe, the type of the object is also explicitly or implicitly assigned, e.g. by putting metadata in a “*water_pipe*” table. Then, an ontology or a data schema can be used to label each entity and the relations. Figure 2 shows KG construction from a partial water network metadata given in EPANET input¹ format.

Semantic Rule Learning refers to learning association rules with semantic information so that the learned rules are not only applicable to specific entities, but applicable to a set of entities with certain characteristics. The proposed semantic rule learning algorithm Naive SemRL, described in Section 3, utilizes a KG constructed in the previous step, and historical time series data. A simple example of a rule that does not contain semantic information is ‘*if sensor1 measures V1, then sensor2 measures V2*’. The goal of the proposed approach is to generate rules

¹<https://www.epa.gov/water-research/epanet>

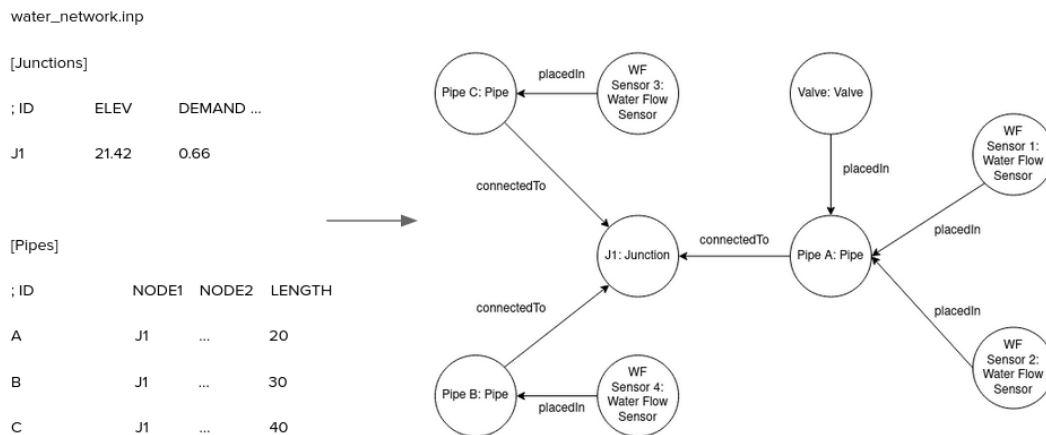


Figure 2: KG construction example for a drinking water network scenario.

Algorithm 1 Naive SemRL

```
1: procedure NAIVEMSEMRL(knowledge_graph, disc_hist_time_series, k_neighbors)
2:   enriched_transactions = []
3:   for transaction in disc_hist_time_series do
4:     topology = graph.topology(transaction.sensor_list(), k_neighbors)
5:     attributes = graph.attributes(transaction.sensor_list(), k_neighbors)
6:     enriched_transactions.append(transaction + topology + attributes)
7:   end for
8:   return FP-Growth(enriched_transactions)
9: end procedure
```

in the form of ‘if a sensor with type T placed in a pipe $P1$ with attribute $1 > A1$ measures $V1$, then the sensor with type $T2$ that is placed in a Junction $J1$ connected to $P1$ measures $V2$ ’.

Making Inferences Based on Semantic Rules. In this phase of the pipeline, real-time time series data is analyzed based on the previously obtained semantic association rules. An inference engine gathers the rules that are not met, e.g. for a certain period of time, and makes inferences based on the unmet rules. The methodology to be used in this step remains to be among future work, while the focus of this study is on the first and second phases of the pipeline.

3. A Naive Semantic Rule Learning Approach - Naive SemRL

The main intuition behind the proposed approach, Algorithm 1, is that instead of learning association rules for individual sensors, it generalizes sensor data using its metadata from the KG. The algorithm does that by extending transactions in a transaction database with semantic information extracted from the KG. As an example, rather than seeing sensor data as ‘*sensor X measured value Y in a time frame T*’, it generalizes to ‘*a sensor with these attributes and neighboring components measured value Y in a time frame T*’.

Naive SemRL requires a KG (*knowledge_graph*), discretized historical time series data (*disc_hist_time_series*, from now on ‘TS’), and number of neighbors (*k_neighbors*) to be analysed for semantic relations as input. TS is a set of transactions where each transaction contains discretized sensor data (items) for a certain time frame. It goes through each of the transaction in TS (lines 3-7), and first finds the topology of the items based on the *k_neighbors* variable (line 4). As an example, in Figure 2, value of the topology variable for *J1:junction* would be [*Pipe_C_ConnectedTo_J1, Pipe_B_ConnectedTo_J1, Pipe_A_ConnectedTo_J1*], assuming the value of *k_neighbors* is equal to 1. A list of attributes for each of the components and links are extracted in line 5, and a new transaction is created in line 6. FP-Growth algorithm is run with the new set of transactions to discover association rules with semantic info in line 8.

4. Preliminary Evaluation and Industrial Use Case

This section presents a semantic rule quality criteria that measures generalizability of semantic association rules, and an industrial use case where the proposed approach is applied.

4.1. A Semantic Rule Quality Criterion

Many quality criteria for association rules are proposed with the most fundamental ones being support and confidence [3]. However, we were unable to identify an association rule quality criterion that is specific to evaluating the semantic aspect of the learned rules. Therefore, we propose the following “*semantic expressivity*” association rule quality measure:

Definition 4.1 (Semantic Expressivity). Let $C = \{c_1, c_2, \dots, c_n\}$ be a set of classes in an ontology (or a data schema). $\forall c \in C(\text{has_attributes}(c, \{a_1^c, a_2^c, \dots, a_m^c\}))$, with $\text{has_attributes}(x, a_m^x) =$ class x has set of a_m^x attributes, and assume $\text{attributes}(x) =$ number of attributes in class x . Let $I = \{i_1, i_2, \dots, i_b\}$ be a set of items. $\forall i \in I(i = (a^c \# z))$, where $a^c =$ an attribute of a class c , $\# =$ any comparison operation, and z any value. $X \rightarrow Y$ is an implication (association rule) where $X, Y \subseteq I$. Finally, let $\text{instances}(X)$ be the different class instances in X and let $\text{attr_count}(X, c)$ be number of items which have attributes of instance of a class c in X .

$$\text{attr_ratio}(X) = \prod_p^{\text{instances}(X)} \frac{\text{attr_count}(X,p)}{\text{attributes}(p)}$$

$$\text{Semantic_Expressivity}(X \rightarrow Y) = \frac{(1-\text{attr_ratio}(X)) \times (1-\text{attr_ratio}(Y))}{(\text{instances}(X) + \text{instances}(Y))/2}$$

Intuition. Learned rules may contain different levels of semantic information. Including too much semantics in the rule makes it less generalizable, hence less ‘*semantically expressive*’. For instance, “*Junctions with 3 pipes have 1500-2000Pa water pressure*” is more general than “*Junctions with 3 pipes where each of the pipes is 50-100m long, and has 2-3m diameter have 1500-2000Pa water pressure*”. The main goal of the proposed quality criterion is to understand how semantically expressive a rule is by giving each rule a score between 0 and 1. Assume X in $X \rightarrow Y$ is ‘ $\{pipe_diameter > 2\}$ ’, with $pipe$ being a class in a water network ontology that can have 3 attributes: diameter, length and elevation. In this case $\text{attr_ratio}(X) = 1/3$ since X is about one of the attributes only. Low attribute ratio leads to high semantic expressivity as the formula includes $(1 - \text{attr_ratio}(X)) \times (1 - \text{attr_ratio}(Y))$. Average number of instances is included in the divisor part for the purpose of incorporating topology into the formula. As an example, an association rule about a node with 3 neighbors will have a higher divisor than a node with 2 neighbors which is more general.

4.2. Industrial Use Case

The proposed algorithm is demonstrated on LeakDB dataset [9], an artificially created realistic leakage dataset for water distribution networks. It contains metadata of 31 junctions, 1 reservoir, 34 pipes and 1,716,960 sensor measurements. Water network related classes in EPYNET Python package² is used as a data schema while creating a KG. MLxtend’s [10] FP-Growth implementation is used while implementing the Naive SemRL algorithm. For simplicity, the proposed approach tested using a straightforward discretization method of lowering sensor measurement precisions and calculating daily averages.

A sample rule learned from the described dataset: ‘ $\{(WaterPressureSensor: WPS, placed_in, Junction: J1), (Junction: J1, connected_to, Pipe: P1), (WaterPressureSensor: WPS, measures, 43)\} \rightarrow \{(WaterConsumptionSensor: WCS, placed_in, Junction: J2), (Junction: J2, connected_to, Pipe: P2), (Junction: J2, connected_to, Pipe: P3), (WaterConsumptionSensor: WCS, measures, 38)\}$ ’.

²<https://github.com/Vitens/epynet>

Table 1

Effect of support threshold on the number of rules and semantic expressivity (SE) (confidence = 0.9)

Support	Naive SemRL			FP-Growth
	Number of Rules	Max SE	Min SE	Number of Rules
0.2	7819	0.66	0.14	28
0.3	816	0.66	0.22	9
0.4	50	0.66	0.33	2
0.5	50	0.66	0.33	2

Interpretation of the rule: ‘When there is a water pressure sensor WPS placed inside a junction J1, and J1 is connected to a Pipe P1, and WPS measures 43, then a water consumption sensor WCS placed in a Junction J2 that is connected to Pipes P2 and P3 must measure 38’. The semantic expressivity of the rule is 0.28, as it does not contain any attribute and based on 3 instances on the antecedents side, and 4 instances on the consequents side. In this experiment, Naive SemRL is run with topology info only without node attributes, as it increases runtime of the algorithm exponentially. Currently, an intuition/search-based approach is investigated that can tell when and where to include semantics in order to avoid exponential increase in the runtime. Table 1 shows how incorporating semantics in the rule learning process increases the number of rules, together with min and max semantic expressivity values. Besides finding new rules about the same nodes extended with semantics, when run with low support thresholds, Naive SemRL can find new rules which FP-Growth can not find. Having more rules is not necessarily good as it may increase the time required for post-processing and making inferences. In order to overcome this hurdle, incorporating semantics within evolutionary or other approaches that can directly learn rules satisfying certain rule quality criteria is being investigated as part of future work.

5. Conclusions and Future Work

This study proposed a semantic association rule learning pipeline for Digital Twins. The pipeline consists of knowledge graph construction, semantic association rule learning from knowledge graphs and time series data, and making inferences. An initial naive approach for semantic rule learning, Naive SemRL, and a first semantic rule quality criterion is proposed. The new approach is evaluated in a water network use case and the results show that incorporating knowledge graphs allows us to learn rules with semantic information which are more generalizable.

There are many open issues and research questions yet to be answered. KG construction for DTs from system metadata and a domain ontology will be automatized. FP-Growth will be replaced by novel rule learning methods with different perspectives, e.g. statistical vs. optimization-based NARM methods. And these methods will be compared based on applicability of the proposed approach and quality criterion. Lastly, an inference mechanism that can detect and find root-causes of discrepancies from semantic rules will be developed.

Acknowledgments

This work has received support from The Dutch Research Council (NWO), in the scope of Digital Twin for Evolutionary Changes in water networks (DiTEC) project, file number 19454. We would like to thank Vitens N.V. for providing us a historical water network sensor dataset to test our approach.

References

- [1] P. Sunhare, R. R. Chowdhary, M. K. Chattopadhyay, Internet of things and data mining: An application oriented survey, *Journal of King Saud University-Computer and Information Sciences* 34 (2022) 3569–3590.
- [2] V. Degeler, A. Lazovik, F. Leotta, M. Mecella, Itemset-based mining of constraints for enacting smart environments, in: *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (Percom Workshops)*, 2014, pp. 41–46. doi:10.1109/PerComW.2014.6815162.
- [3] M. Kaushik, R. Sharma, I. Fister Jr, D. Draheim, Numerical association rule mining: A systematic literature review, *arXiv preprint arXiv:2307.00662* (2023).
- [4] R. D. D’Amico, J. A. Erkoyuncu, S. Addepalli, S. Penver, Cognitive digital twin: An approach to improve the maintenance management, *CIRP Journal of Manufacturing Science and Technology* 38 (2022) 613–630.
- [5] E. Karabulut, S. F. Pileggi, P. Groth, V. Degeler, Ontologies in digital twins: A systematic literature review, 2023. *arXiv:2308.15168*.
- [6] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, *ACM sigmod record* 29 (2000) 1–12.
- [7] J. Akroyd, S. Mosbach, A. Bhave, M. Kraft, Universal digital twin-a dynamic knowledge graph, *Data-Centric Engineering* 2 (2021) e14.
- [8] G. Tamašauskaitė, P. Groth, Defining a knowledge graph development process through a systematic review, *ACM Transactions on Software Engineering and Methodology* 32 (2023) 1–40.
- [9] S. G. Vrachimis, M. S. Kyriakou, et al., Leakdb: a benchmark dataset for leakage diagnosis in water distribution networks:(146), in: *WDSA/CCWI Joint Conference Proceedings*, volume 1, 2018.
- [10] S. Raschka, Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack, *The Journal of Open Source Software* 3 (2018). URL: <https://joss.theoj.org/papers/10.21105/joss.00638>. doi:10.21105/joss.00638.