2023 European Conference on Computing in Construction
40th International CIB W78 Conference
Heraklion, Crete, Greece
July 10-12, 2023

SEMANTIC WEB-ENABLED OUTLIER AND MISSING VALUE DETECTION AND REPLACEMENT IN SMART BUILDINGS

Alex Donkers[1], Dujuan Yang[1], and Bauke de Vries[1]
[1]Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract

Digital twins have the potential to leverage AI in buildings. The quality of AI algorithms is dependent on the quality of the input data and its preprocessing. This paper discusses the potential of using semantic web technologies in preprocessing tasks. After reviewing state-of-the-art initiatives in this field, a data integration method is introduced based on semantic web technologies. This integrated data is used in approaches to find outliers and missing values in time series data and in two semantic similarity-based imputation methods. The paper shows that semantic web technologies can enhance preprocessing tasks, by applying both explicit and implicit reasoning.

Introduction

Construction digital twins have the potential to take artificial intelligence (AI) in the construction industry to a next level (Boje *et al.*, 2020). Leveraging AI could lead to better simulations of our buildings, potentially leading to predictions and optimization of systems in the built environment. Solutions to improve building energy use and indoor comfort, detect operational inefficiencies and HVAC (heating, ventilation and air conditioning) failures, understanding occupant behavior, and improving safety and scheduling are within reach (Petrova *et al.*, 2019; Boje *et al.*, 2020).

Various authors mention data interoperability as a requirement to reach this potential and mention semantic web technologies as a solution (Pauwels, Zhang and Lee, 2017; Boje *et al.*, 2020). Gary Marcus (2020) boldly states that semantic models are a requirement for robust intelligence, and that AI systems should be able to reason over cognitive models of our real world in order to draw knowledge and be reliable.

Recent research initiatives followed the claims of Marcus and used semantic web technologies to support AI in the construction industry. Petrova et al. (2019) used data mining techniques to find patterns in indoor sensor data. Esnaola-Gonzalez et al. (2018) applied data mining to predict future indoor temperature states. Semantic web technologies were applied in data selection, preprocessing, and transformation stages.

The quality of AI models is dependent on the quality of the data that we feed them. Data preparation is therefore an important, but also time-consuming task (Perez-Rey, Anguita and Crespo, 2006). Kang (2013) lists various techniques for handling missing data. These range from simple deletion (listwise deletion, pairwise deletion) to simple imputation (mean substitution, regression imputation, last observation carried forward) to more complex imputation methods (maximum likelihood, expectation-maximization, multiple imputation, and sensitivity analysis). As Kang (2013) mentions, these data preparation methods require expert knowledge about the specific system and its context. Especially in the architecture and construction industry, typified as a mass customization industry, a single piece of code is not likely to provide accurate data preparation results. Buildings are unique objects, causing the context of sensors to be different from case to case. Simultaneously, sensors, databases, and end-user applications differ per building, resulting in a wide range of protocols that need to be understood by the expert. Paradoxically, while the cleaning process requires expert knowledge, the development of IoT systems causes data in the built environment to come in growing volumes, making it harder for experts to understand and process this data.

In this work, we aim to find the potential of using semantic web technologies in the data preparation phase. The work specifically focuses on three practical tasks: 1. What values in the time series database are missing values? 2. What values in the time series database are outliers? 3. By what values should those missing values and outliers be replaced?

The paper starts with reviewing related work, after which the data integration method is presented. Sections 4 and 5 respectively present methods to find missing values and outliers. Section 6 presents a method to replace these values based on semantic similarity, after which this paper ends with a discussion and conclusion.

Related work

Semantic web and data preprocessing

Fayyad et al. (1996) described a five-step model for knowledge discovery in databases: *1. Selection*, *2.*

Preprocessing, *3. Transformation*, *4. Data mining*, and *5. Evaluation and interpretation*. There has been an uptake in research aiming to combine semantic web technologies with preprocessing tasks, as covered by the literature review of Ristoski and Paulheim (2016). They found that ontologies can help in performing preprocessing tasks, such as data validity checks and data cleaning, and categorized these ontologies into two groups. The first category contains domain-independent specifications for preprocessing tasks. The second category contains domain-specific knowledge that can be used to reason on how to preprocess data.

Perez-Rey et al. (2006) created the OntoDataClean ontology to represent a generic, domain-independent model for data preprocessing in the semantic web. Every data source in the knowledge graph is linked to an instance of the CleaningModel-class, which then explicitly describes how that data source should be cleaned in case of a missing value.

Gao et al. (2013) used their Correlated Environmental Sensor Properties ontology to manually add information about the correlation between two sensor properties. They could for example model that the temperature of a space is strongly correlated to the relative humidity of that space. This information is then queried and combined with the physical distance between two sensors to find sensors that are neighboring and correlated. Outliers and suspicious values are detected by comparing sensor streams with the streams of those similar sensors.

The SemOD framework (Esnaola-Gonzalez *et al.*, 2017) also aims to find suspicious sensor values by reusing information from knowledge graphs. Contextual information about the sensors, in this case the solar radiation of an outdoor temperature sensor, is added to the knowledge graph. Compared to Gao et al. (2013), SemOD (Esnaola-Gonzalez *et al.*, 2017) aims to be more expressive in identifying the root cause of the outlier by using different types of information than just physical distance and correlation. Rules to find outliers are then transformed into SPARQL queries that return outliers based on the information in the knowledge graph.

Esnaola-Gonzalez et al. (2021) argue that the performance of machine learning methods to impute data decreases if they are used outside their training environments, and that the capabilities of semantic web technologies to create structured representations of metadata can improve imputation methods. They proposed a semantics-based data imputation approach to replace missing values. An office room is represented using the BOT (Rasmussen *et al.*, 2020) ontology and time-series data is converted to RDF (resource description framework) using SOSA/SSN (Janowicz *et al.*, 2019). Missing values of a sensor are replaced by values of the most similar sensor. The similarity is calculated based on physical distance and the hosting building element of the sensor.

Some of the examples in this section require transforming time series data into RDF triples. Fürber and Hepp (2013) argue that this adds complexity to the preprocessing stage, while earlier research (Esnaola-Gonzalez and Javier Diez, 2019; Petrova *et al.*, 2019) argues that time series data should be stored in time series databases for performance reasons.

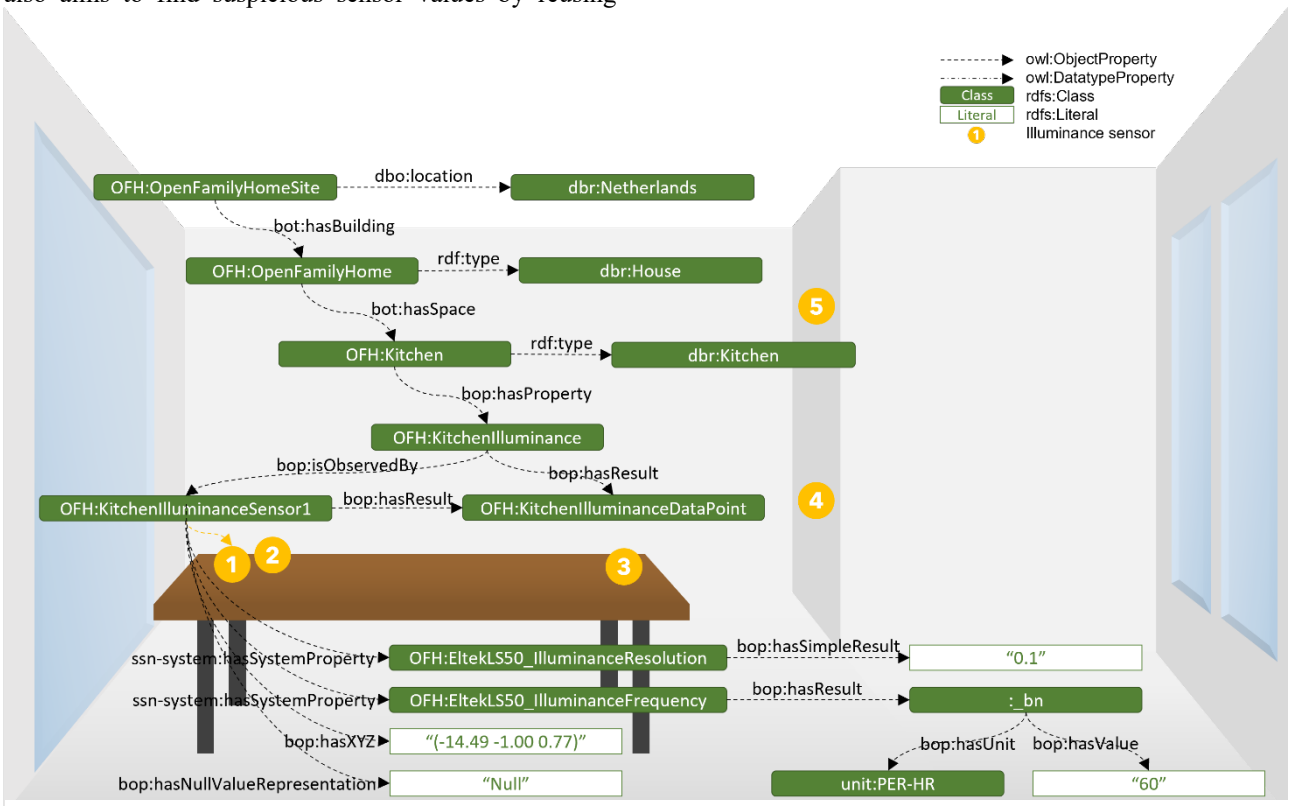

Figure 1: Structure of the knowledge graph

Semantic web and building information

Integrating data in the construction industry is a challenge. It typically requires integrating cross-domain knowledge from multiple stakeholders that speak their own domain language, produce files in different software packages, and store them in different locations. The de facto standard for Building Information Models (BIM), IFC (Industry Foundation Classes), is limited in its data integration capabilities (Pauwels, Zhang and Lee, 2017).

In response to data integration challenges similar to those in BIM, semantic web technologies are proposed as a solution, first introduced by Tim Berners-Lee (Berners-Lee, Hendler and Lassila, 2001). In the semantic web, information is structured as triples following the resource description framework (RDF) model. Multiple RDF graphs can be linked with each other, eventually resulting in a web of data. Data in this web can be classified using ontologies, that give a semantic meaning and structure to the data.

Since the introduction of the semantic web, various initiatives took place to represent building information using semantic web technologies (Pauwels, Zhang and Lee, 2017). This led to the development of a central ontology for the AEC sector, the ifcOWL ontology (Pauwels and Terkaj, 2016). Critiques on this ontology – it being too complex, difficult to extend – were answered by the development of a range of smaller domain ontologies, capturing for example a building's topology (Rasmussen *et al.*, 2020), static and dynamic properties (Donkers *et al.*, 2022), and geometry (Wagner *et al.*, 2019).

Methodology

To test our approach, a digital representation of a residential building is created using semantic web technologies. Figure 1 shows a simplified representation of the semantic representation and Figure 2 shows a graphical summary of how the data are integrated and used in this paper. The building – the Open Family Home – is created using Revit 2020 and converted to RDF Turtle, based on methods presented in earlier work (Donkers *et al.*, 2021). It follows the BOT (Rasmussen *et al.*, 2020) and BOP (Donkers *et al.*, 2022) ontologies to represent the buildings' topology and static and dynamic properties, respectively. Indoor environmental quality parameters (temperature, relative humidity, CO2, indoor air quality, and illuminance) were measured for 4 weeks in January and February 2022 with a frequency of one measurement per minute. The sensor data is stored in an InfluxDB cloud storage, and basic metadata of this storage is added to the knowledge graph to automatically identify the right data point in the InfluxDB database.

The core topological nodes of the building are enriched with DBPedia (a linked open data encyclopedia) resources. This enrichment can be used to perform human-like reasoning. As DBPedia is an open-source knowledge graph, multiple stakeholders in multiple projects can link concepts to the same DBPedia resources, so that queries can be reused in the future. In this work, we reused location information from DBPedia.

The sensor nodes in the knowledge graph are extended with system capabilities provided by the manufacturer. To do so, the ssn-system ontology[1] was reused. The ontology introduces concepts to describe a range of system capabilities, such as measurement range, resolution, frequency, and drift.

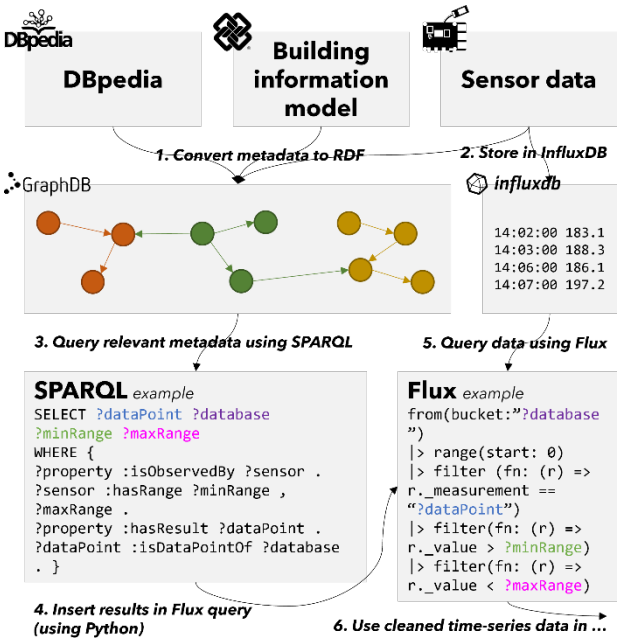

Figure 2: Graphical summary of the research method

Results

Missing value detection

Sensors and databases have different methods to represent missing values (or null values). This adds complexity to the preprocessing phase, as the data scientist (or algorithm) should know this representation before processing these specific values. The architecture of systems and databases is different for most projects, which is why simple filters to find null values require manual input from the data scientist. This information can however be stored in the knowledge graph, so that data scientists can query that information and feed this into their algorithms. Figure 1 shows how the null value representation of a sensor can be added to the graph by using a simple datatype property. Listing 2 queries this representation using SPARQL, after which it can be used to automatically find the null values from a specific sensor. Similar to adding a null value representation to the sensor, one could also add this information to (or deduce it from) the data point node in the graph, as databases might also have their own representation of null values.

[1] https://www.w3.org/ns/ssn/systems/

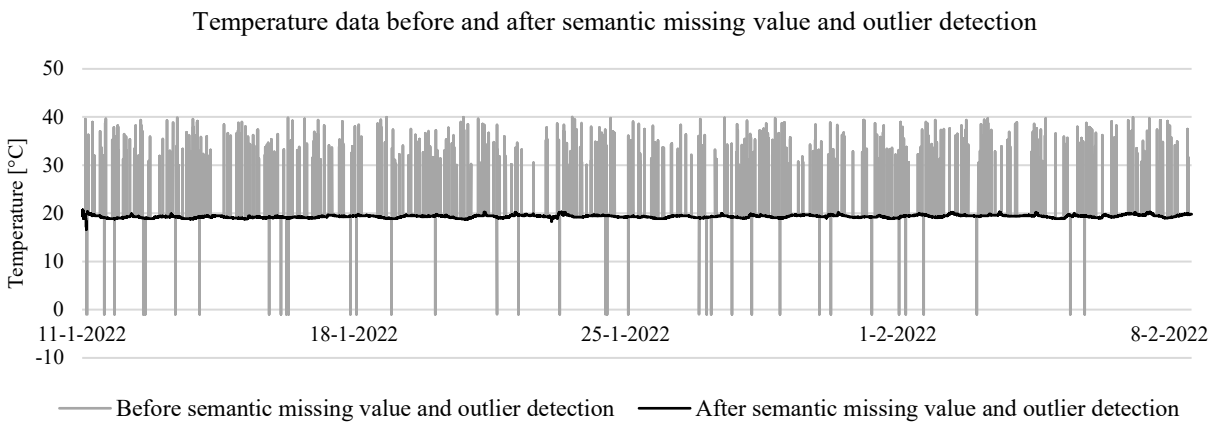

Figure 3: Temperature data before and after the missing value and outlier detection

The explicit reasoning-based missing value detection requires that the database contains a timestamp with a missing value representation. In case of communication failures, sensors might not be able to communicate with a database at all, possibly causing a gap in the timestamps. The previously introduced approach will not be sufficient in those cases.

To find whether there should be a value at a certain timestamp, we can perform implicit reasoning using the information in the knowledge graph. Listing 1 shows how the frequency of a sensor can be queried using SPARQL. Using this frequency, one could identify gaps in a time series stream. Consider the time series data in Listing 2. Based on the frequency of this sensor (Figure 1) and the type of sensor, we would expect data at 14:04 and 14:05. In this implicit reasoning-based missing value detection, we calculate the time difference of two consecutive time series measurements. If this time difference is two or more times the sensor's frequency, we are expecting missing data points. Algorithms can then impute new timestamps by taking the timestamp before the gap (14:03 in Listing 2) and adding timestamps based on the queried frequency until it reaches the timestamp after the gap (14:06 in Listing 2).

```
PREFIX OFH:
<http://github.com/AlexDonkers/OpenFamilyHome#>
PREFIX bop: <https://w3id.org/bop#>
PREFIX ssn-system:
<http://www.w3.org/ns/ssn/systems/>
SELECT * WHERE {
    OFH:Kitchen bop:hasProperty ?property .
    ?property a quantitykind:Temperature .
    ?property bop:isObservedBy ?sensor .
    ?sensor ssn-system:hasSystemProperty ?range ,
        ?frequency .
    ?sensor bop:hasNullValueRepresentation
        ?nullValueRepresentation .
    ?range bop:hasSimpleMinimum |
        bop:hasSimpleMaximum ?rangeValue .
    ?frequency a ssn-system:Frequency .
    ?frequency bop:hasSimpleResult ?frequencyValue .
}
```

Listing 1: Querying the measurement range and null value representation of a temperature sensor using SPARQL

```
2022-11-23T14:02:00Z 183.1 KitchenIlluminanceSensor1
2022-11-23T14:03:00Z 188.3 KitchenIlluminanceSensor1
2022-11-23T14:06:00Z 186.1 KitchenIlluminanceSensor1
2022-11-23T14:07:00Z 197.2 KitchenIlluminanceSensor1
```

Listing 2: Time series data containing two missing values

Outlier detection

A data scientist can classify a value as an outlier because it is technically impossible, or because it is very unlikely given the context of the sensor. This section aims to tackle both challenges using the available data in the RDF graph.

Sensor systems are typically designed to only measure values between a certain measurement range. This range can be explicitly described in the knowledge graph using the ssn-system ontology. This measurement range contains a minimum and maximum value. Listing 1 shows how these values can be queried using SPARQL.

Next to applying explicit reasoning to find outliers, knowledge graphs enable implicit reasoning to apply more complex knowledge patterns. In practice, experts might classify values as outliers not only based on the system boundaries but also based on expert knowledge. For example, a certain temperature sensor might technically be able to measure temperatures between -30 and 50 °C, but if this sensor measures the temperature of an indoor space in a residential building in the Netherlands, this technical range is way larger than the reasonable range of values that we can expect. Experts, therefore, use the available contextual information to determine if a value is reasonable.

Listing 3 and Listing 4 show how implicit reasoning based on contextual information can be used to filter out outliers. Listing 3 first selects all sensors that observe the temperature in a residential building in the Netherlands and adds a new custom measurement range to these sensors. This custom range can then be queried using SPARQL (Figure 2) and inserted in a Flux query (Listing 4) using a filter function, to only query the values that fall within this range.

Figure 3 shows the result of both the missing value detection and outlier detection on a temperature data

Table 1: Results of the similarity search

	Entity	Score	bop:isHostedBy	bop:hasXYZ
1	OFH:KitchenIlluminanceSensor1	1.0000	OFH:KitchenTable	"(-13.59 1.09 0.77)"
2	OFH:KitchenIlluminanceSensor2	0.8996	OFH:KitchenTable	"(-13.49 1.00 0.77)"
3	OFH:KitchenIlluminanceSensor3	0.7923	OFH:KitchenTable	"(-14.49 -1.00 0.77)"
4	OFH:KitchenIlluminanceSensor4	0.6542	-	"(-11.32 -2.55 0.77)"
5	OFH:KitchenIlluminanceSensor5	0.6507	-	"(-11.32 -2.55 2.00)"

stream, measured in the Open Family Home. Both the outliers and missing values are removed based on the information in the knowledge graph.

```
PREFIX OFH:
<http://github.com/AlexDonkers/OpenFamilyHome#>
PREFIX bot: <https://w3id.org/bot#>
PREFIX ssn-system:
<http://www.w3.org/ns/ssn/systems/>
PREFIX bop: <https://w3id.org/bop#>
PREFIX quantitykind:
<http://qudt.org/vocab/quantitykind/>
PREFIX dbo: <https://dbpedia.org/ontology/>
PREFIX dbr: <https://dbpedia.org/resource/>
INSERT {
    ?sensor ssn-system:hasSystemProperty
        OFH:CustomRange .
    OFH:CustomRange rdf:type ssn-system:Range,
        bop:Property , ssn-system:CustomRange.
    OFH:CustomRange bop:hasSimpleMinimum "10" .
    OFH:CustomRange bop:hasSimpleMaximum "30" .
} WHERE {
    ?property bop:isObservedBy ?sensor .
    ?property a quantitykind:Temperature .
    ?sensor bop:isHostedBy ?host .
    ?zone bot:containsElement ?host .
    ?building bot:hasSpace ?zone .
    ?building a dbr:House .
    ?site bot:hasBuilding ?building .
    ?site dbo:location dbr:Netherlands .
}
```

Listing 3: Insert a custom range for temperature sensors in residential buildings in the Netherlands

```
from(bucket: "?database")
    |> range(start: v.timeRangeStart, stop:
        v.timeRangeStop)
    |> filter(fn: (r) => r["_measurement"] ==
        "?dataPoint")
    |> filter(fn: (r) => r._value > ?minRange)
    |> filter(fn: (r) => r._value < ?maxRange)
```

Listing 4: Query data within the new custom range using Flux

Semantic similarity-based imputation

After finding the missing values and outliers, the knowledge graph is used to replace those values with more realistic ones. Two methods are proposed: the semantic similarity approach and the semantic shape follower approach. In both methods, values are replaced based on the values of similar sensors. To find those similar sensors, a predication-based semantic indexing (PSI) (Cohen, Schvaneveldt and Rindflesch, 2009) algorithm was applied in GraphDB. In PSI, semantic predications are encoded based on the information in the knowledge graph and presented in a vector space. PSI creates a vector for each entity in the graph and uses these vectors to calculate similarity. Four training cycles were performed to improve the outcome of the predictions.

Table 1 shows the results of the similarity search when searching for similar objects as OFH:KitchenIlluminanceSensor1. The location of the five sensors is visible in Figure 1. As expected, OFH:KitchenIlluminanceSensor2 has the highest similarity score. We, therefore, use the time series data of this sensor in the semantic similarity-based imputation.

In the first method – the semantic similarity approach – missing values and outliers of sensor 2 are directly replaced by the value of sensor 1. In the second method – the semantic shape follower approach – missing values and outliers of sensor 2 are computed by taking the previous value and incrementing the same value as the increment of sensor 1. The imputed values in the second method thus follow the same shape as the time series of sensor 1 but have a different starting value. The first approach hypothetically works well if two sensors are expected to have similar values, while the second approach hypothetically works well if the time series of two sensors have similar shapes.

These approaches were tested on OFH:KitchenIlluminanceSensor1 and OFH:KitchenIlluminanceSensor2. Two case studies are tested. In the first case study (MV), we randomly replaced single values with outliers and missing values in the time series of sensor 2. In the second case study (MP), we randomly added missing periods of 30 successive values. Next to the two developed approaches, a linear interpolation and a multiple imputation (Kang, 2013) that combines the other three methods are tested. The sensor's resolution (Figure 1) is queried from the knowledge graph using SPARQL and all newly added values are rounded based on this resolution.

Table 2 shows the average value and the root mean squared error for all four methods and two case studies. It shows that for this specific case, the semantic shape follower approach is the most accurate. Figure 4 shows the shape of the imputed data, where missing values were replaced between 09:28 and 09:58. The semantic shape follower approach seamlessly follows the original data.

Table 2: Results of the different imputation methods

		Linear interpolation		Semantic similarity		Sematic shape follower		Multiple imputation	
	Actual	MV	MP	MV	MP	MV	MP	MV	MP
Average	28.21	28.18	28.30	28.27	28.38	28.19	28.22	28.21	28.30
RMSE	0	1.23	1.49	1.37	0.93	1.23	0.27	1.24	0.59

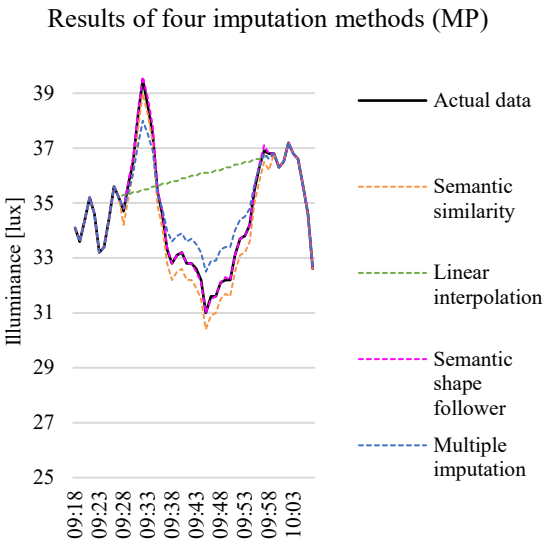

Figure 4: Results of four imputation methods (MP)

While the semantic shape follower does perform the best in this use case, it has a limitation, namely that it requires a similar sensor to be installed near the original sensor. This is often not the case in practice. Therefore, a second study is performed to test if the sensor values could be replaced by an illuminance sensor on the second floor of the Open Smart Home. The sensor is placed in an office space that is located on the same side of the building as the kitchen. Next to this, the same experiment is performed with two temperature sensors with similar locations. Table 3 shows the results of this experiment. The *actual* values represent the values by the kitchen sensors on floor 0, while the values in the semantic similarity and semantic shape follower columns make use of the office sensors on floor 1 to replace missing values and outliers. The similarity score shows the similarity of the two sensors, based on the PSI algorithm as also presented in Table 1.

Using the illuminance sensor in the office to replace missing values results in a low performance of the algorithms compared to the performance in Table 2. Zooming into the same timeframe as Figure 4 shows that the shape of the imputed data is nowhere near the actual data. However, imputing the missing values of the temperature sensor with the sensor in the office room – especially when using the semantic shape follower approach - shows very good performance. The difference is explainable. The office room and the kitchen are used differently over time, and the lighting sources are individually controlled for both spaces. The temperature, however, is controlled by a central heating system. Even though the temperature in the kitchen and the office space might not be the same, their patterns show the same behavior.

Discussion

This paper presented an approach to apply semantic web technologies in the data preprocessing phase in the architecture and construction industry. Since a one-size-fits-all solution is unlikely to perform well in the construction industry, this paper applies data from knowledge graphs to enhance preprocessing tasks.

To the best of our knowledge, this is the first work that uses implicit reasoning on cross-domain (and potentially federated) knowledge graphs to perform both missing value and outlier detection and replacement in time series databases. By applying semantic web technologies in the data preparation phase, automate certain parts of the reasoning process and reduce the human effort of data preparation compared to existing methods (as mentioned by Kang (2013)).

Various challenges remain unanswered in this paper. This work shows an example of applying both implicit and explicit reasoning-based algorithms in a residential home in the Netherlands. The exact algorithms, the data that needs to be stored in the knowledge graph, and the

Table 3: Performance of the semantic imputation approaches when using sensors from another floor

	Illuminance			Temperature		
	Actual	Semantic similarity (MP)	Semantic shape follower (MP)	Actual	Semantic similarity (MP)	Semantic shape follower (MP)
Average	28.21	43.98	**28.37**	19.52	19.45	**19.52**
RMSE	0	76.01	**14.87**	0	0.26	**0.05**
Similarity score	1.000	0.3238	0.3238	1.000	0.2054	0.2054

reasoning performed in this study are highly context-specific. Researchers are encouraged to redo this study for other building types, other sensors, and other locations in the world.

For this reason, the results of the imputation methods are only valid for this specific use case. Researchers should distinguish the best imputation methods for various cases. If more knowledge in this domain becomes available, the knowledge graph might again help the data scientist in finding the most suitable method based on the available contextual information.

We applied a semantic similarity algorithm that found the most similar sensor. However, this work does not provide insight into when a similarity score is high enough to replace values. In fact, while the similarity between the two temperature sensors in Table 3 was lower than between the two illuminance sensors, the imputation algorithms performed better. The search query to find similar objects can be extended to show more contextual information (such as the host and coordinates in Table 1). Performing more practical use cases could give more insight into acceptable similarity scores, and the extended search query might give practitioners more insight into why the scores are as such.

Finally, this research is reactive and focuses on problem-solving. The results currently give no insights into why certain sensors show more outliers or missing values. By extracting certain data quality metrics from the algorithms in this research, and feeding them back into the knowledge graph, we might be able to inform facility managers about failing systems or find root causes for suspicious behavior of sensors.

Conclusion

Data preprocessing is a time-consuming task that requires expert knowledge. To reach the full potential of AI, approaches to ease these tasks should be developed. Due to the nature of the construction industry – a mass customization industry – single pieces of code are not likely to perform accurate preprocessing tasks on a large scale. Semantic web technologies have proven to enable the integration of cross-domain building information in so-called semantic digital twins. This paper applied these semantic web technologies to enhance preprocessing tasks, more specifically, the following three practical tasks: 1. What values in the time series database are missing values? 2. What values in the time series database are outliers? 3. By what values should those missing values and outliers be replaced?

Reviewing the state-of-the-art shows us that there is an uptake in combining semantic web technologies and AI, however, the practice of applying semantic web technologies in preprocessing tasks in the construction industry is limited.

Based on methods in earlier research, heterogeneous building information was integrated by converting data from a building information model, product data from various Eltek sensors, and metadata related to the placement of those sensors to an RDF turtle format and combining these files in GraphDB. The knowledge graph was enriched by DBpedia data.

The paper then introduces methods to find outliers and missing values. First, explicit reasoning is used to find those values. A sensor's null value representation and measurement range were added to the knowledge graph. After querying them using SPARQL, they were used to filter data on missing values and outliers, respectively. As not all missing values and outliers let themselves be caught with explicit reasoning, more complex, implicit reasoning approaches are introduced. Missing values are found by finding gaps in the time series data based on the resolution of the sensor, while outliers are found by adding custom measurement ranges based on contextual information about the sensor in the knowledge graph.

Finally, the paper introduces two methods to replace the missing values and outliers with new values. First, semantic similarity is calculated using a predication-based semantic indexing algorithm. The sensor data of similar sensors are then used in two approaches: the semantic similarity approach and the semantic shape follower approach. These approaches were tested for sensors in the same room, but also for sensors on a different floor. The performance of the approaches is highly context-specific, strengthening our views that contextual information in semantic digital twins can enhance preprocessing tasks. Following the viewpoint of Marcus (Marcus, 2020), semantic models of our real world do have the potential to enhance AI systems, at least in the construction industry.

Acknowledgments

The authors would like to gratefully acknowledge the support from Eindhoven University Technology, KPN (TKI-HTSM 19.0162), and the Netherlands Enterprise Agency, as part of the 'SmartTWO: Maverick Telecom Technologies as Building Blocks for Value Driven Future Societies' project (TK|L912P06).

References

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The semantic web', Scientific American, 284(5), pp. 35-43. doi: 10.1038/scientificamerican0501-34.

Boje, C., Guerriero, A., Kubicki, S. and Rezgui, Y. (2020) 'Towards a semantic Construction Digital Twin: Directions for future research', Automation in Construction, 114, 103179. doi: 10.1016/j.autcon.2020.103179.

Cohen, T., Schvaneveldt, R. W. and Rindflesch, T. C. (2009) 'Predication-based semantic indexing: permutations as a means to encode predications in semantic space', AMIA Annual Symposium proceedings, 2009, pp. 114-118.

Donkers, A., Yang, D., De Vries, B. and Baken, N. (2022) 'Semantic Web Technologies for Indoor Environmental Quality: A Review and Ontology

Design', Buildings, 12(10), 1522. https://doi.org/10.3390/buildings12101522.

Donkers, A., Yang, D., De Vries, B. and Baken, N. (2021) 'Real-Time Building Performance Monitoring using Semantic Digital Twins', Proceedings of the 9th Linked Data in Architecture and Construction Workshop, Luxembourg, Luxembourg, pp. 55-66.

Esnaola-Gonzalez, I., Bermúdez, J., Fernández, I., Fernández, S. and Arnaiz, A. (2017) 'Towards a semantic outlier detection framework in wireless sensor networks', in ACM International Conference Proceeding Series, Amsterdam, The Netherlands, pp. 152-159. doi: 10.1145/3132218.3132226.

Esnaola-Gonzalez, I., Bermúdez, J., Fernández, I. and Armaiz, A. (2018) 'Semantic prediction assistant approach applied to energy efficiency in Tertiary buildings', Semantic Web, 9(6), pp. 735-762. doi: 10.3233/SW-180296.

Esnaola-Gonzalez, I., Garciarena, U. and Bermúdez, J. (2021) 'Semantic Technologies Towards Missing Values Imputation', Proceedings of the 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Kuala Lumpur, Malaysia, pp. 191-196. doi: 10.1007/978-3-030-79457-6_16.

Esnaola-Gonzalez, I. and Javier Diez, F. (2019) 'Integrating building and IoT data in demand response solutions', Proceedings of the 7th Linked Data in Architecture and Construction Workshop, Lisbon, Portugal, pp. 92-105.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) 'From data mining to knowledge discovery in databases', AI Magazine, 17(3), 37. doi: 10.1609/aimag.v17i3.1230.

Fürber, C. and Hepp, M. (2013) 'Using semantic web technologies for data quality management', in Sadiq, S. (eds.) Handbook of Data Quality: Research and Practice, Springer, pp 141-161. doi: 10.1007/978-3-642-36257-6_7.

Gao, L., Bruenig, M. and Hunter, J. (2013) 'Semantic-based detection of segment outliers and unusual events for wireless sensor networks', Proceedings of the 18th International Conference on Information Quality, ICIQ 2013, Little Rock, United States, pp. 127-144.

Janowicz, K., Haller, A., Cox, S.J.D., Le Phuoc, D., Lefrançois, M. (2019) 'SOSA: A lightweight ontology for sensors, observations, samples, and actuators', Journal of Web Semantics, 56, pp. 1-10. doi: 10.1016/j.websem.2018.06.003.

Kang, H. (2013) 'The prevention and handling of the missing data', Korean Journal of Anesthesiology, 64(5), pp. 402-406. doi: 10.4097/kjae.2013.64.5.402.

Marcus, G. (2020) The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. http://arxiv.org/abs/2002.06177.

Pauwels, P. and Terkaj, W. (2016) 'EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology', Automation in Construction, 63, pp. 100-133. doi: 10.1016/j.autcon.2015.12.003.

Pauwels, P., Zhang, S. and Lee, Y. C. (2017) 'Semantic web technologies in AEC industry: A literature overview', Automation in Construction, 73, pp. 145-165. doi: 10.1016/j.autcon.2016.10.003.

Perez-Rey, D., Anguita, A. and Crespo, J. (2006) 'OntoDataClean: Ontology-based integration and preprocessing of distributed data', Proceedings of the 7th International Symposium on Biological and Medical Data Analysis, Thessaloniki, Greece, pp. 262-272. doi: 10.1007/11946465_24.

Petrova, E., Pauwels, P., Svidt, K. and Jensen, R.L. (2019) 'In Search of Sustainable Design Patterns: Combining Data Mining and Semantic Data Modelling on Disparate Building Data', in Advances in Informatics and Computing in Civil and Construction Engineering, Proceedings of the 35th CIB W78 2018 Conference: IT in Design, Construction and Management, Chicago, Illinois, United States, pp. 19-26. doi: 10.1007/978-3-030-00220-6_3.

Rasmussen, M. H., Lefrançois, M., Schneider, G.F. and Pauwels, P. (2020) 'BOT: The building topology ontology of the W3C linked building data group', Semantic Web, 12(1), pp. 143-161. doi: 10.3233/sw-200385.

Ristoski, P. and Paulheim, H. (2016) 'Semantic Web in data mining and knowledge discovery: A comprehensive survey', Journal of Web Semantics, 36, pp. 1-22. doi: 10.1016/j.websem.2016.01.001.

Wagner, A. et al. (2019) 'Relating geometry descriptions to its derivatives on the web', in Proceedings of the 2019 European Conference on Computing in Construction, Chania, Greece, pp. 304-313. doi: 10.35490/ec3.2019.146.