Semantic Web 1 (0) 1–5
IOS Press

FIDES: An Ontology-based approach for making Machine Learning systems Accountable

Iker Esnaola-Gonzalez [a,*], Jesús Bermúdez [b]

[a] *TEKNIKER, Basque Research and Technology Alliance (BRTA), Iñaki Goenaga 5, 20600 Eibar, Spain*
E-mail: iker.esnaola@tekniker.es
[b] *E-mail: jesus.bermudez@ehu.eus*

Editor:
Solicited review:
Open review:

Abstract. Although the maturity of the artificial intelligence technologies is rather advanced nowadays, its adoption, deployment and application is not as wide as it could be expected. This could be attributed to many barriers, where the lack of trust of users stands out. Accountability is a relevant factor to advance in this trustworthiness aspect, as it enables discovering the causes that derived a given decision or suggestion made by an artificial intelligence system. In this article, the use of ontologies is conceived as a way for making machine learning systems accountable, thanks to their conceptual modelling capabilities to describe a domain of interest, as well as formality and reasoning capabilities. The feasibility of the proposed approach has been demonstrated in a real-world energy efficiency scenario and it is expected to pave the way towards raising awareness of the possibilities of semantic technologies in different factors that may be key in the trustworthiness of artificial intelligence-based systems.

Keywords: Accountability, Ontology, Artificial Intelligence, Machine Learning

1. Introduction

Even though the maturity of the artificial intelligence (AI) technologies is rather advanced nowadays, according to McKinsey[1], its adoption, deployment and application is not as wide as it could be expected. This could be attributed to many barriers including cultural, economic and technical [1, 2], as well as social barriers, where the lack of trust of potential end-users in AI systems is remarkable [3, 4]. As a matter of fact, there are many concerns that derive in this lack of trust such as potential safety issues that may lead to harm humans [5, 6] and biases towards the penalisation of certain social groups [7–9]. However, this lack of trust, if carefully managed, can be overcame thus contributing to the acceptance of AI systems [2].

AI trustworthiness can be defined as "the extent to which a user is confident in, and willing to act on the basis of, the recommendations, actions, and decisions of an artificially intelligent decision aid" [10]. There are many factors that affect this lack of trust [11, 12], including the explainability. This factor has been addressed by the so-called explainable artificial intelligence (XAI), which refers to the "techniques that enable human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners" [13]. XAI was intensively studied from the 1970s to the 1990s [14], although a resurgence of the topic has been seen recently

*Corresponding author. E-mail: iker.esnaola@tekniker.es
[1] https://www.mckinsey.com/featured-insights/artificial-intelligence/ai-adoption-advances-but-foundational-barriers-remain

due to the current technological advancements in the various fields of the AI [15].

The explainability is necessary but far from sufficient for achieving the desired trustworthiness in AI systems. In order to do so, not only should the developed AI systems be explainable, but also accountable [16, 17]. As a matter of fact, the ability to hold them accountable by explaining their inner workings, their results and the causes of failures to users, regulators and citizens, is critical to achieve trust [18].

The accountability can be defined as the ability to determine whether a decision was made in accordance with procedural and substantive standards and to hold someone responsible if those standards are not met [17]. This means that with an accountable AI system, the causes that derived a given decision can be discovered, even if its underlying model's details are not fully known or must be kept secret. In other words, the person, group or company in charge of the AI system should be able to answer questions that are related, not only to the obtained outputs (e.g. what the output result is or when the output is generated), but also to the AI procedures that led to such outputs (e.g. which data set(s) are used to train the AI system or how well the AI system performs in terms of accuracy).

However, the information needed to answer these questions is hardly ever accessible in a straightforward way. This information is scattered across multiple files, repositories and systems, and in the worst-case scenario, is not even registered. That means that, if the person, group or company in charge of the AI system wanted to answer the aforementioned questions, it would be very time consuming, as it would be needed to be an expert or have the help of experts in different frameworks, systems, data models, repositories and query languages. As a matter of fact, the regular performance these accountancy tasks would be infeasible.

Therefore, it seems reasonable to consider that the adequate representation of data, processes and workflows involved in AI systems could contribute to make them accountable in an easier and systematic manner. There are a variety of technologies that offer conceptual modelling capabilities to describe a domain of interest, but only ontologies combine this feature with Web compliance, formality and reasoning capabilities [19].

Since the AI is a field that comprises a variety of fields ranging from natural language processing to knowledge representation [20], this article focuses on a specific branch: the machine learning (ML). Namely, an ontology-based approach is proposed towards achieving the accountability of ML systems. The rest of the article is structured as follows. Section 2 presents the related work. The proposed ontology-based approach is described in Section 3 and demonstrated in a real-world use case in Section 4. It is evaluated and discussed in Section 5 and finally, conclusions of this work are shown in Section 6.

2. Related Work

Although the usage of Semantic Technologies towards the achievement of Trustworthy AI has been researched in the literature, their full potential is yet to be exploited.

[21] provides a literature-based overview of the usage of Semantic Technologies alongside ML methods in order to facilitate their explainability. According to the reviewed literature, the main role of the Semantic Technologies is, on the one hand, to make Neural Networks explainable, and on the other, to create explainable embeddings with knowledge graphs. As for the domains of application, the healthcare domain has attracted a lot of attention, although they are also present in the entertainment or commercial field.

[22] presents an approach for creating more understandable post-hoc explanations of decision tree algorithms. In this approach, ontologies that model the concerned domain knowledge are used in the process of generating such explanations. Results showed that decision trees generated with the support of domain ontologies are more understandable than those generated without them. The downside of this approach is that the used ontologies are manually created ad-hoc for each problem, which definitely hinders its usability.

[23] proposes an explanation ontology that can be used by designers to support the generation of different explanation types into their AI-enabled systems. Nine different explanation types are identified, each with different needs, and the proposed ontology can encode them as OWL restrictions. This provides a means for system designers to translate their user requirements gathered from user studies to explanations that can be generated by their systems.

Doctor XAI is presented in [24], an ontology-based approach for producing post-hoc explanations of black-box sequential data classification methods. The application of the approach is focused on the medical domain, but since the method is agnostic with regards to the black box model, the possible applications cover several scenarios where a sequence of events linked

to ontology concepts can be identified, including an online market basket analysis or Wikipedia user behaviour forecast.

[25] proposes an ontology-based knowledge representation and reasoning framework for human-centred transfer learning explanations. This approach exploits the reasoning capabilities offered by the Semantic Technologies and makes use of external knowledge bases to infer different kinds of human understandable explanatory evidence, allowing common users without ML expertise to have a good insight of transfer learning explanations. In [26], semantic reasoning and ML have been combined for explaining the rationale of classification predictions in an informative manner to human users, which is expected to in turn strengthen the trust relationship between human decision makers and intelligent systems making the prediction.

Knowledge graphs can provide an explainable layer that may act an effective way to interpret the black-box answers given by neural models [27, 28], which have been proved to be useful in the field of conversational agents [29] and recommender systems [30]. Furthermore, the existing approaches, limitations and opportunities for knowledge graphs in XAI are analysed in [31]. In this article, knowledge graphs are envisioned to bring XAI to the right level of semantics and interpretability, supporting explanations that may overtake existing limitations in different AI fields, ranging from computer vision to natural language processing.

In [32], it is stated that semantic representations for explainability can evolve from existing representations for provenance and context. Therefore, the strengths of the Semantic Web, coupled with ML methods, will be a significant contributor to hybrid explainable AI systems. To the extent of knowledge of author, so far, the main focus of the usage of Semantic Technologies has been placed on explainability, although accountability is considered a key requirement that should be met to achieve trustworthy AI systems [33, 34]. [35] makes a first contribution on the usage of ontologies to support the accountability of ML systems, proposing a method to know which predictive model was responsible for making a given forecast, but also, to understand where such forecasts come from, that is, which is their underlying rationale. However, many fundamental aspects that could contribute to making the ML systems accountable remain unaddressed, such as the description of the procedure followed to develop the predictive models.

All this evidence reinforces the discourse that the Semantic Technologies could play a more important role in achieving trustworthy AI systems in general, and in solving the accountability challenge for ML systems in particular.

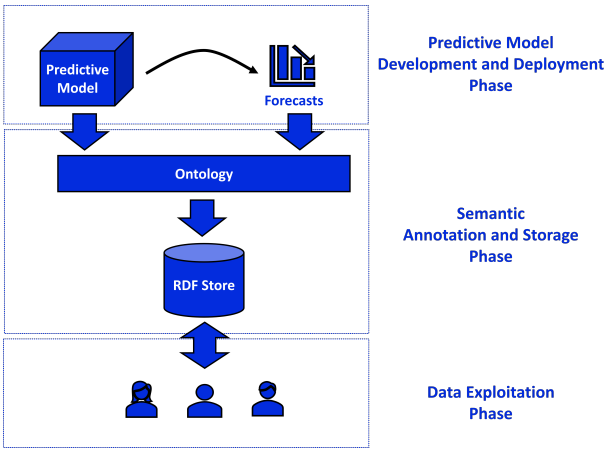

Fig. 1. Outline of FIDES.

3. FIDES: Making Machine Learning Systems Accountable

Towards the achievement of accountable ML systems, this article presents FIDES[2]. FIDES is a tool that leverages ontologies for representing, structuring and setting formal relations among the predictive models and the forecasts that conform a ML system, and provides end-users with the necessary means to exploit this knowledge and answer the pertinent questions.

3.1. FIDES at a glance

This approach consists of three phases as shown in Figure 1.

In the first phase, the predictive model that will solve the problem at hand is developed. Depending on the type of the problem addressed, and the quality and the amount of the data available, some algorithms may provide better results than others. Furthermore, the adequate fine-tuning of the hyperparameters of the algorithms may have a direct effect on the performance of the final model. Therefore, at this stage, the data scientists in charge of developing the model will need to make the opportune choices. Once the model is generated, it is deployed into production in order to generate the aimed forecasts. FIDES is model-agnostic with

[2]Fides was the Roman goddess of trust.

a view to be valid for the wide variety of existing ML algorithms, and it works with predictive models developed in R programming language and deployed in RServe[3], a server that allows to execute R implementations.

In the second phase, the relevant information related to the developed predictive model and the generated forecasts are retrieved from R and Rserve respectively, and annotated with the adequate ontological terms. Then, the resulting RDF triples are automatically stored in an Openlink Virtuoso[4] repository. Both the retrieval and semantic annotation of the data, and the storage of the RDF triples is fully automated with a service based on Apache Jena[5], so there is no need for human intervention. The main goal of this service is to minimise potential performance issues and errors derived from manual practices. More details on the ontologies used to annotate the relevant data are provided in Section 3.2.

In the third phase, the end-users are provided with a GUI (Graphical User Interface) that facilitates the retrieval of the information that helps to make ML systems accountable. This GUI implements a set of API methods developed in Apache Jena, which in turn execute a set of predefined parameterizable SPARQL queries, thus abstracting end-users from the underlying query language.

3.2. Ontologies for Accountability

In order to make ML systems accountable, FIDES envisions the representation of two main knowable topics: the forecast made by the predictive model, and the procedure followed by such a predictive model for making the forecast. In order to formalise the information requirements for each knowable topic, the Competency Questions (CQ) were used as proposed by different ontology engineering methodologies such as NeOn [36] or LOT [37].

Regarding the forecasts made by a predictive model, it may be of interest to represent the information that may answer to CQs of the following style:

- Which is the value of a given forecast?
- When was a given forecast generated?
- What or who generated a given forecast?

As for the procedure followed by a given predictive model for making forecasts, the information that may be of interest can be divided in, on the one hand, the information addressing the data used to train the predictive model, and on the other, the information concerning the details of the procedure implemented by the predictive model.

The training data can be characterised by its features, including the amount of data used, the dependent and independent variables considered, and the statistical characteristics such as the variance, mean or median of the data. Therefore, the CQs of the following style could be of interest:

- Which is the frequency of a given predictive model's training data?
- Which is amount of observations used for training a given predictive model?
- When was the last data point within a given predictive model's training data collected?

Regarding the predictive model's procedure details, information related to the algorithm used and its hyperparameters may be of interest, as well as the performance assessed in development time. This can be procured by CQs of the following style:

- Which is the base algorithm of the predictive model?
- Which is are the hyperparameter values of the predictive model?
- Which is the RMSE of the predictive model?

The answering of this kind of CQs contributes to, on the one hand, the identification of potential causes that may have led to undesirable outcomes produced by AI systems, and on the other, the evaluation and assurance that AI systems are legally, ethically and technologically robust, while respecting democratic values, human rights and the rule of law, for example, as requested by the EU Artificial Intelligence Act[6].

3.2.1. Ontology selection

Once the requirements that the ontology needs to satisfy were identified, the adequate set of ontology terms were developed or selected and used to annotate such information. In this regard, the ontology reuse best practice [38] was followed and the approach for selecting the potential ontologies to be reused was inspired by the Ontological Resource Reuse Pro-

[3] https://www.rforge.net/Rserve/
[4] https://virtuoso.openlinksw.com/
[5] http://jena.apache.org/
[6] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206&from=EN

cess [39]. For the search of relevant ontologies, four reputed sources were consulted: LOV and LOV4IoT ontology catalogues, and Google Scholar[7] and ScienceDirect[8] research databases. Additionally, for the assessment, comparison and the final selection of the ontologies to be reused, the following set of ontology quality criteria defined by [40] were followed:

- Having an explicit license that specifies that they can be used and under which conditions.
- Having enough documentation to understand the ontology purpose, domain and fundamentals, and determine whether it describes this domain appropriately or not.
- Having a minimum metadata to help human users and computer applications understand the data as well as other important aspects that describe a data set.

3.2.2. Ontologies for representing forecasts

Currently, there are many ontologies which could be used for representing events or activities, the result of which is an estimate of the value of a quality of the feature of interest, obtained using a specific procedure. A thorough analysis of ontologies covering such a domain can be found in [40], and it can be concluded that the SOSA/SSN ontology[9] proposed by [41, 42] may be one of the most appropriate ontologies for representing forecasts due to, for example, its nice documentation, complete metadata and alignments to related domain ontologies. However, SOSA/SSN ontology's admission of different models to represent the same state of affairs may derive in interoperability problems, which is why it was discarded in FIDES. Instead, the EEPSA ontology[10] proposed by [43] was selected to be reused, as it was developed on the basis that a proper axiomatisation shapes the set of admitted models better, and therefore, establishes the ground for a better interoperability.

Although being developed for supporting a data analyst assistant in energy efficiency and thermal comfort problems in buildings [44], the backbone of the EEPSA ontology is defined as a combination of three Ontology Design Patterns (ODP) that can be used as basic building blocks to address similar problems in different domains. These ODPs try to be minimal in the number of classes and properties offered, but include appropriate ontology axioms that allow proper inferences. Namely, the three ODPs are the AffectedBy ODP[11], the Execution-Executor-Procedure (EEP) ODP[12] and the Result-Context (RC) ODP[13]. Thanks to the great flexibility provided by this ODP-based ontology engineering modelling solution, the combination of these three ODPs have led to the development of ontologies covering other domains such as the agrifood as presented in [45].

The AffectedBy ODP defines two classes representing features of interest (*aff:FeatureOfInterest*) and their qualities (*aff:Quality*) and three object properties: *aff:belongsTo*, *aff:affectedBy* and *aff:influencedBy*. The *aff:belongsTo* object property supports the notion that every quality belongs to the feature of interest it is intrinsic to (i.e. a quality cannot belong to different features of interest), thus following the conceptualisation defined in the DOLCE upper level ontology proposed by [46]. The *aff:affectedBy* object property relates a quality with another quality that it affects, and the *aff:influencedBy* object property relates a quality with the feature of interest that it influences.

The EEP ODP imports the AffectedBy ODP and its two classes, and additionally, it defines three more classes: *eep:Execution*, *eep:Executor*, and *eep:Procedure*. An individual of *eep:Execution* is an event (e.g. a forecast) upon a quality of a feature of interest, produced by an agent by performing a procedure. As for an individual of *eep:Executor*, it is an agent (e.g. a predictive model) capable of performing tasks by following procedures. Lastly, an individual of *eep:Procedure* (e.g. the procedure implemented by a predictive model) describes the workflow, protocol, plan, algorithm, or computational method to be executed by agents to produce an event. Furthermore, the *eep:madeBy*, *eep:usedProcedure*, and *eep:onQuality* object properties are introduced in the EEP ODP. The *eep:madeBy* object property links an execution to the agent that performs the action, the *eep:usedProcedure* object property links an execution to the procedure that describes the task to be performed; and the *eep:onQuality* object property links an execution to the quality concerned by the execution. These three functional object properties, combined with a set of property chain axioms defined in the EEP ODP, allow the inference of the remaining object properties *eep:implements* linking executors to proce-

[7] https://scholar.google.com
[8] https://www.sciencedirect.com
[9] http://www.w3.org/ns/ssn
[10] http://w3id.org/eepsa
[11] https://w3id.org/affectedBy
[12] https://w3id.org/eep
[13] https://w3id.org/rc

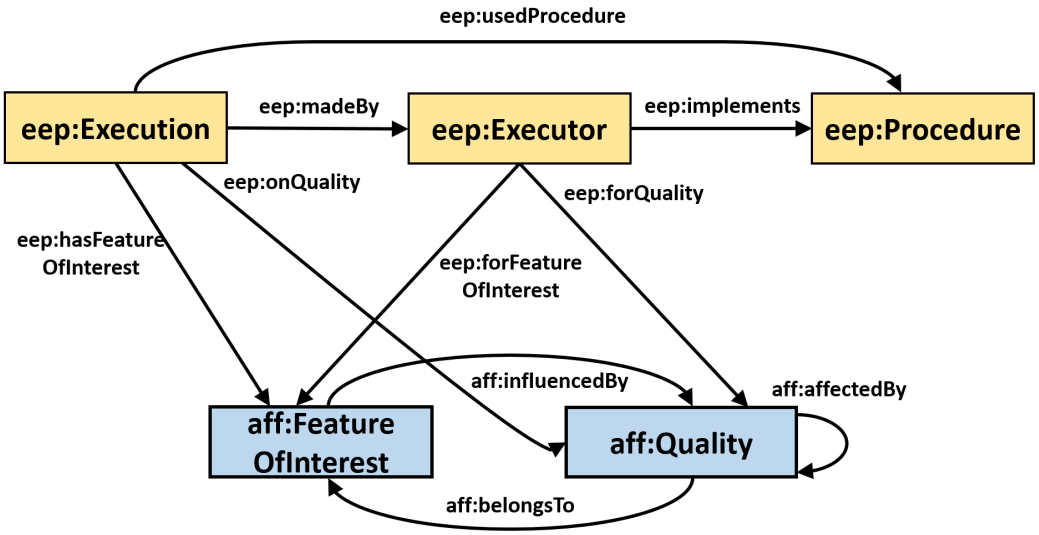

Fig. 2. The main classes and properties of the AffectedBy, EEP and RC ODPs used for the annotation of forecasts.

dures, *eep:hasFeatureOfInterest* linking executions to features of interest, *eep:forQuality* linking executors to qualities, and *eep:forFeatureOfInterest* linking executors to features of interest (see Figure 2).

The RC ODP aims at representing the results of the executions defined in the EEP ODP as well as their contexts. These results can be complex objects that usually include units of measurement, the measurement value, and some other optional parameters, but sometimes, a simple representation with a literal type value may suffice. Both complex and simple results can be modelled with the *rc:hasResult* object property and the *rc:hasSimpleResult* datatype property respectively. Furthermore, temporal and spatial aspects of a result are represented in the RC ODP with the *rc:hasGenerationTime* data property and the *rc:hasTemporalContext* object property for the former, and the *rc:hasSpatialContext* object property for the latter.

These three ODPs are published in the ODP repository OntologyDesignPatterns.org[14] and they are available online with a CC BY 4.0 license. They have a well-presented documentation, careful metadata with explanatory descriptions of the intended meanings of their terms, and alignments to other domain ontologies such as the SOSA/SSN ontology or W3C's PROV-O ontology[15] to ensure clarity in modelling and avoid errors that may have unintended reasoning implications [47]. Hence, they satisfy the ontology quality criteria established in Section 3.2.1.

3.2.3. *Ontologies for representing predictive procedures*

The other knowable topic that needs to be represented with adequate ontological terms is the one concerning the predictive procedures used for achieving forecasts. The existing ontologies in this domain are not as abundant as for the previous one, although there are still ontologies covering ML experiments and different areas of the data mining, such as the OntoDM-core ontology described in [48] or the DMOP ontology presented in [49]. However, there is a gap between these ontologies, which definitely hampers an ideal interoperable scenario. Towards reducing such a gap and achieving a higher level of interoperability among those resources, the ML-Schema[16] [50] was developed within the W3C Machine Learning Schema Community Group[17]. And this is the ontology selected to be reused for FIDES. It is an ontology that provides a set of classes, properties, and restrictions to represent different aspects of ML processes. On the one hand, resources to describe the data used as input and their characteristics and quality are offered. On the other, resources to describe the implementations, algorithms used to develop models and their hyperparameters are defined. Finally, the developed models, their characteristics and the evaluation obtained in the training phase

[14] http://ontologydesignpatterns.org/
[15] https://www.w3.org/TR/prov-o/
[16] http://www.w3.org/ns/mls
[17] https://www.w3.org/community/ml-schema/

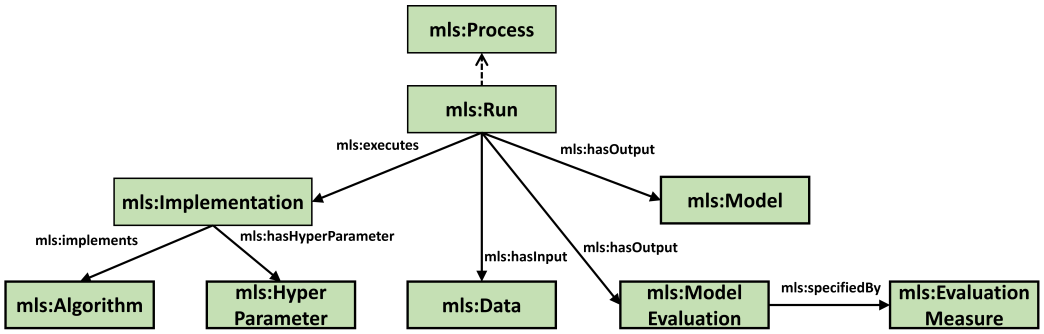

Fig. 3. The main classes and properties of the ML-Schema used for the annotation of predictive models.

can also be represented with this ontology. Figure 3 shows the main classes and relationships defined in the ML-Schema.

The ML-Schema is published in the LOV catalogue and it is available online with a W3C Community Contributor License Agreement. Even though it has a complete documentation page, the metadata associated to the resources described in the ontology are incomplete. As a matter of fact, the guidelines proposed by [51], which are considered one of the most complete ontology metadata guidelines to date, are not met in many cases. For example, there is no human readable label for any of the defined terms, and additionally, there are classes (e.g. *mls:DatasetCharacteristic*) and properties (e.g. *mls:hasValue*) which have no metadata associated at all, thus their intended meaning is not clear and their functionality is open to interpretation. Therefore, the W3C Machine Learning Schema Community Group should work on this issue to improve the vocabulary's reusability[18]. The ML-Schema is also mapped to more specific ontologies and vocabularies focused on ML such as the MEX vocabulary presented by [52].

Summarising, the ontologies leveraged by FIDES for the representation of the relevant information are, on the one hand, the AffectedBy, the EEP and the RC ODPs for representing forecasts, and on the other, the ML-Schema for representing predictive procedures. The alignment between these ontologies is rather straightforward thanks to their design with a view to be easily extended and complemented with other ontological resources. As a matter of fact, two RDF triples suffice to integrate the aforementioned ontologies:

$$mls{:}Process \sqsubseteq eep{:}Procedure$$

where the ML-Schema's *mls:Process* class is defined as a subclass of EEP ODP's *eep:Procedure* class, and

$$mls{:}Model \sqsubseteq eep{:}Executor$$

where the ML-Schema's *mls:Model* class is defined as a subclass of EEP ODP's *eep:Executor* class.

4. FIDES in use

In order to illustrate the validity of FIDES, it has been implemented in a real-world energy efficiency scenario. In this scenario, a total of 122 residential and small commercial buildings located in the island of Lanzarote (Spain) have participated, and for each of them, a predictive model that forecast the electric demand of the next 24 hours had to be developed. Then, the generated forecasts would be used as an input for the overall energy efficiency solution, which is out of scope. This scenario has been running for a total of 62 days, therefore, a total of 181,536 forecasts (24 forecast per day per building unit x 122 building units x 62 days) have been generated in total by the 122 predictive models developed.

These forecasts had to be accountable as it was an explicit requisite of the energy efficiency solution that they were part of. With a classic approach, the information describing the relationship between the predictive models and the building units to which they correspond, is gathered in an Excel file. As for the forecasts, they are stored in a relational database where specific SQL queries have to be executed to retrieve the desired ones. Likewise, some minor details of the predictive models such as its performance, are stored in another table of the same database. Finally, other information of the predictive models is not collected, thus different

[18] An issue related to this matter is opened at the moment of writing this article in https://github.com/ML-Schema/core/issues/25.

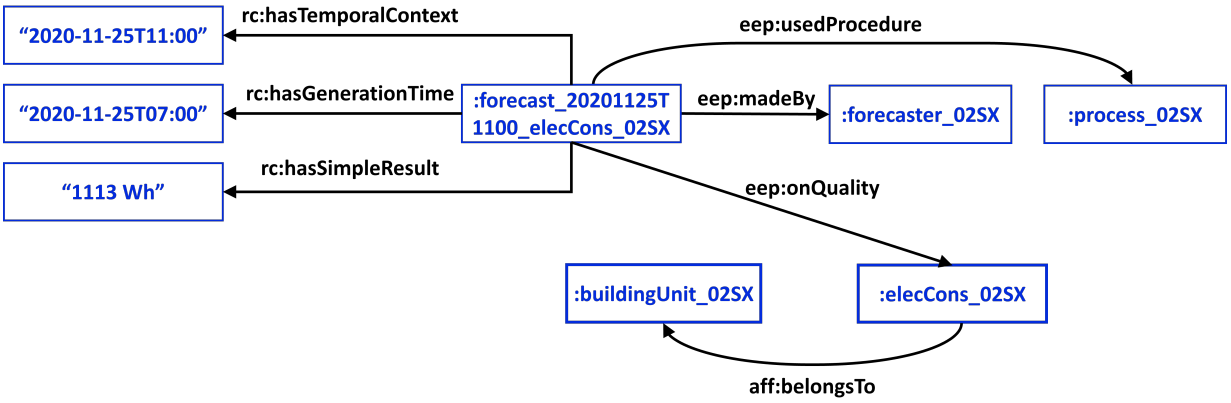

Fig. 4. Simplified graphic representation of the triples representing the 02SX building unit's electric consumption forecast.

functions have to be executed in the R framework over each predictive model to access this information.

Therefore, it is evident that managing the accountability in the scenario described is not a trivial task and that an approach supported by technologies that enable the management of the semantics and interrelationships of data, as well as the knowledge representation could ease this process. This is why FIDES has been implemented.

4.1. *Predictive Model Development and Deployment phase*

The 122 predictive models have been developed by a team of data scientists in the R programming language, and all of them have been implemented using the same KNN algorithm of the caret[19] package. However, since the training data available for each building unit differed due to various reasons (e.g. lost of data derived from the malfunctioning of electricity meters), the value of the *k* hyperparameter also differed, and it was assigned by using the forward-chaining time series cross validation method. Furthermore, the forecasting effectiveness of the developed models has been evaluated with the RMSE (Root Mean Squared Error) metric.

The developed predictive models have been exported in the form of *.rds* files and put into production in an Rserve version 3.2.5 deployed in a Docker[20] container. These predictive models have been automatically executed once a day during 62 days, using periodical tasks executed by a *cron daemon* process.

[19]http://caret.r-forge.r-project.org/
[20]https://www.docker.com/

4.2. *Semantic Annotation and Storage phase*

Once all the 122 predictive models have been generated and deployed in Rserve, their corresponding RDF triples have been automatically generated and stored in the Virtuoso Open-Source Edition version 7.2.5.1 repository. Likewise, each time a forecast has been generated, its corresponding RDF triples have been automatically generated and stored in the same repository.

For the sake of demonstrating the automatic semantic annotation within FIDES, let us consider the following simplified use case. A given predictive model was executed on 2020/11/25 at 07:00 and forecast that the building unit 02SX would have an electric consumption of 1,113 Wh on 2020/11/25 at 11:00. This predictive model was trained with 7,423 data points collected from the 02SX building unit. The features of the training set included, apart from the electric consumption, the hour when the measurement was made, the weekday and whether it was a working day or not. This predictive model was based on the R language's caret package's KNN algorithm implementation with the hyperparameter *k* set to 7, and obtained an RMSE of 242.03 Wh. The triples describing the predictive model developed for the 02SX are represented by Figure 5

The forecast has been defined as an instance of the *eep:Execution* class. It has been made by (*eep:madeBy*) a given predictive model (*eep:Executor*) and produced by (*eep:usedProcedure*) following a given procedure represented as individual of the *eep:Procedure* class. The properties defined in the RC ODP have been used for representing the actual value of the forecast (*rc:hasSimpleResult*), the instant when the forecast has

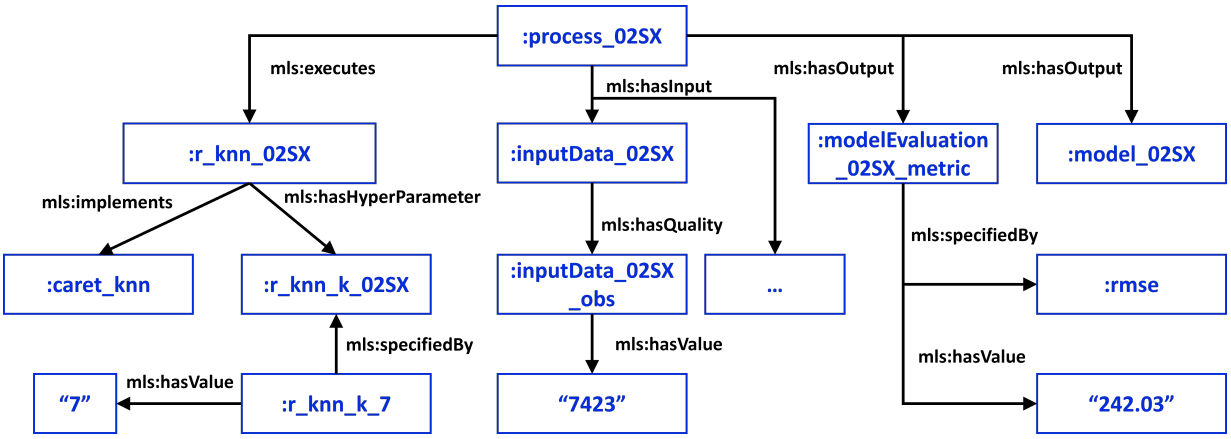

Fig. 5. Simplified graphic representation of the triples representing the example scenario's predictive model.

been generated (*rc:hasGenerationTime*) and the time for when the forecast is valid (*rc:hasTemporalContext*). Additionally, the forecast has been related with the electric consumption of the building unit 02SX that predicts (represented as individual of the *aff:Quality* class) via the *eep:onQuality* object property, and this quality has been linked with the building unit 02SX (represented as an individual of the *aff:FeatureOfInterest* class) it belongs to. The triples describing the electric consumption forecast made for the 02SX are represented by Figure 4.

Regarding the procedure used for obtaining such a forecast, it has been represented as an individual of the *mls:Run* class, which is a subclass of the broader *mls:Process* class. This procedure has executed an R environment implementation (*mls:Implementation*) of the KNN algorithm (*mls:Algorithm*) with the *k* hyperparameter (*mls:Hyperparameter*) value set to 7. Additionally, the procedure has been related to the data set used for the training process (*mls:Dataset*) via the *mls:hasInput* object property. This data set's features have been represented with individuals of the *mls:DatasetCharacteristic* class and linked with the *mls:hasQuality* object property. Finally, the resulting predictive model has been represented as an individual of the *mls:Model* class and it has been related with the procedure that generated it via the *mls:hasOutput* object property. Likewise, the predictive model's evaluation (*mls:ModelEvaluation*) has been specified by an individual of the *mls:EvaluationMeasure* class (in this case representing the RMSE) via the *mls:specifiedBy* object property and with a value of 242.03 Wh. The predictive model and its features are characterised by the triples represented in Figure 5. In addition, the RDF triples representing both the forecast and the predictive model's procedure can be found in Appendix A.

4.3. Data Exploitation phase

Once the forecasts and details of the procedure used by predictive models to generate such forecasts are semantically annotated and stored in the RDF Store, FIDES make use of a GUI to let end-users interact with this information. First of all, a list with all the participant building units is displayed as shown in Figure 6. This list is obtained automatically by calling an API method that executes a predefined SPARQL query.

For the sake of demonstrating the different data exploitation functionalities offered by FIDES, let us consider that the manager of the energy efficiency solution may want to know:

- Which is the performance obtained in the training of the model that forecasts the electric consumption of the building unit 02SX?

This information can be discovered by clicking the 'Performance' button of the row belonging to the 02SX building unit, which in turn calls an API method that instantiates and executes the predefined parameterizable SPARQL query shown in Listing 1. When clicking the option, wild card *$FORECAST_QUALITY* is automatically replaced with the forecast quality's URI (in this example, *:elecCons_02SX*). The results obtained are shown in Table 1.

```
PREFIX eep: <https://w3id.org/eep#>
PREFIX mls: <http://www.w3.org/ns/mls#>
PREFIX rdf: <http://www.w3.org/1999/02/
      22-rdf-syntax-ns#>
```

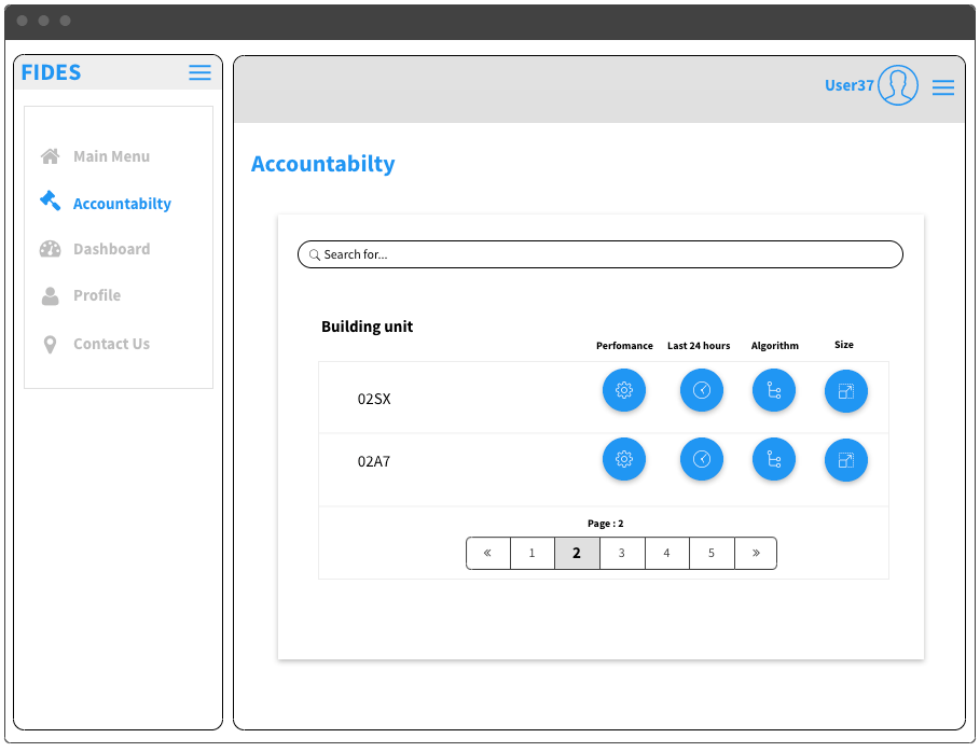

Fig. 6. FIDES GUI.

```
PREFIX rdfs: <http://www.w3.org/2000/01/
        rdf-schema#>

SELECT ?performanceMetric
       ?performanceValue
WHERE {
?forecast eep:onQuality
         $FORECAST_QUALITY;
  eep:usedProcedure ?procedure.

?procedure mls:hasOutput ?modelEval.

?modelEval rdf:type mls:ModelEvaluation;
  mls:specifiedBy ?performanceMetricURI;
  mls:hasValue ?performanceValue.

?performanceMetricURI rdfs:label
  ?performanceMetric.
}
```

Listing 1: SPARQL query for retrieving the performance obtained in the training of the model that forecast a certain quality for a certain instant of time.

Likewise, the manager may also wonder:

Table 1

Results obtained after running the SPARQL query shown in Listing 1, parameterised with the desired values.

?performanceMetric	?performanceValue
RMSE	242.03

– Which is the algorithm and hyperparameters used by the model that forecast the electric consumption of the building unit 02SX?

This information can be discovered by clicking the 'Algorithm' button of the row belonging to the 02SX building unit, which in turn calls another API method that instantiates and executes the predefined parameterizable SPARQL query shown in Listing 2. Just like in Listing 1, wild card *$FORECAST_QUALITY* and is automatically replaced with the corresponding values (in this example, *:elecCons_02SX*)). The results obtained are shown in Table 2.

```
PREFIX rdfs: <http://www.w3.org/2000/01/
        rdf-schema#>
PREFIX eep: <https://w3id.org/eep#>
PREFIX mls: <http://www.w3.org/ns/mls#>
PREFIX rc: <https://w3id.org/rc#>
```

```
SELECT ?algorithm ?hyperparameter
       ?hyperparamValue
WHERE {
?forecast eep:onQuality
          $FORECAST_QUALITY;
  rc:hasTemporalContext $FORECAST_TIME;
  eep:usedProcedure ?procedure.

?procedure mls:executes ?implementation.

?implementation mls:implements
          ?algorithmURI;
  mls:hasHyperParameter
          ?hyperparameterURI.

?hyperparameterSetting mls:specifiedBy
          ?hyperparameterURI;
  mls:hasValue ?hyperparamValue.

?algorithmURI rdfs:label ?algorithm.

?hyperparameterURI rdfs:label
          ?hyperparameter.
}
```

Listing 2: SPARQL query for retrieving the algorithm and hyperparameters of the model that forecast a certain quality for a certain instant of time.

Table 2
Results obtained after running the SPARQL query shown in Listing 2, parameterised with the desired values.

?algorithm	?hyperparameter	?hyperparamValue
knn	k	7

Apart from the two API methods and corresponding predefined parametrisable SPARQL queries shown above, FIDES has two additional functionalities that may contribute to hold the ML accountable but, for the sake of simplicity, are not shown in this article. Namely, functionalities that answer the questions:

- Which are the last 24 forecasts generated by a given predictive model?
- Which are the starting and ending dates of the training data of a given predictive model?

The answer to these questions can be obtained by clicking in the 'Last 24 hours' and 'Size' buttons of the GUI respectively, which will in turn call the corresponding API methods that will trigger the parametrisation of the SPARQL queries, following the same process as shown for the previous two questions.

5. Evaluation and Discussion

The validity of FIDES has been evaluated for the presented scenario from three different points of view: usability, functionality and scalability. Likewise, three different people with different backgrounds participated in the evaluation process: two data scientists and one system manager. It is worth noting that, although they had different backgrounds, all of them had previous experience with energy efficiency problems. Additionally, the classical approach a person should follow for holding a ML system not supported by FIDES is also explained, for the sake of evaluating the accessibility to the information.

5.1. Usability

The usability of FIDES has been measured with the SUS (System Usability Scale) [53]. It consists in a questionnaire with ten questions, where participants are asked to score them with one of five responses that range from Strongly Agree (5 points) to Strongly disagree (1 point). It allows to evaluate a wide variety of products and services, including hardware, software, mobile devices, websites and applications, and it has become an industry standard. The SUS questionnaire has been used after participants have interacted with FIDES at least once and before any discussion took place. Furthermore, as suggested by the methodology itself, participants have been asked to record an immediate response to each question, rather than thinking about items for a long time.

The average score obtained was 80.8 out of 100, so it can be concluded that the overall usability of FIDES is very good. The most remarkable outcomes of this questionnaire are that all participants think that they would use FIDES frequently as it is easy to use and quick to learn. The average score for each question is shown in Figure 7.

5.2. Functionality

At the moment of writing this article, FIDES offers four functionalities with a view to help end-users holding ML systems accountable:

- Obtain the performance of a given predictive model.
- Obtain the algorithm and hyperparameter values of a given predictive model.
- Obtain the last 24 forecasts generated by a given predictive model.

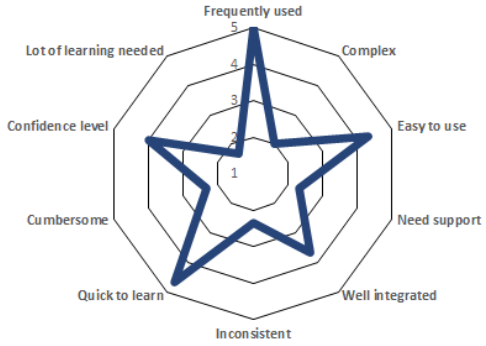

Fig. 7. Average SUS score of FIDES.

– Obtain the starting and ending dates of the data set used to train a given predictive model.

Participants have been asked about the functionalities offered by FIDES (to which degree these four functionalities satisfied their needs) and whether they would require from further functionalities. Two of the interviewees responded that the functionalities satisfied all their needs and that they did not have additional requests at the moment. The third one responded that a functionality that provided additional information related to the training data's quality dimensions like completeness (related to the number of missing values) or time uniqueness (related to the duplicated data) could be helpful.

Thanks to the fine-grained semantic representation of both the forecasts made by the predictive model and the procedure followed by such a predictive model for making the forecasts, additional queries are at hand. However, these queries are not currently implemented in the FIDES interface, and should therefore be made by executing specific SPARQL queries in the RDF Store's SPARQL endpoint. This means that end-users should, on the one hand, have enough knowledge to design SPARQL queries, and on the other, be aware of the network of ontologies used to represent the collected information. This would indeed diminish the advantage of FIDES, which consists in making ML systems accountable in a straightforward manner and without the need of having specific technical skills.

Therefore, FIDES is expected to extend the functionalities offered in further releases by developing API methods that execute predefined parameterisable SPARQL queries, instead of letting users access the SPARQL endpoint directly.

5.3. Scalability

The scalability of FIDES has been evaluated considering the number of triples it needs to scale to the usage of the proposed real-world energy efficiency scenario over the time.

FIDES uses around 35 triples to represent a predictive model, which could increase depending on the number of hyperparameters described of the algorithm used for building the model and the required inputs. At first, once the predictive models that conform an ML systems are represented, the number of triples should remain stable. However, under normal conditions, predictive models' performance degrade over time due to a change in the environment that violates the models assumptions [54], and models need to be retrained. This retraining consists in re-running the process that generated the previous model on the new set of data available but, since according to [55] the electric consumption patterns are strongly influenced by occupants' behaviour and lifestyle, a change in their habits (e.g. a family member leaving the household or a commercial building extending their working hours) may make the typical retraining process insufficient and new predictive models may be necessary. This means that each building unit may need more than a single predictive model over time, specially amidst this COVID-19 situation [56], which leads to a linear growth of triples.

As for the representation of each forecast, it needs around 12 triples. In this kind of ML systems where forecasts are generated with a rather high frequency, the amount of triples generated after a certain period of time increases in a linear way but can end up growing in an exponential way, as the number of predictive models increases. A direct consequence of having such an increasing amount of triples are the high query latencies. A similar situation has already been addressed in the literature for IoT data, which is characterised by its abundance, and it is recommended to be stored in suitable storage systems like Time Series Database (TSDB) [40, 57, 58]. These databases are optimised for time series data, thus being able to manage such an amount of data while ensuring a high performance. Therefore, in order to ensure storage efficiency and a reduced query latency, FIDES should consider storing forecasts in TSDBs, at a cost of losing some semantically annotated information.

5.4. Accessibility

As mentioned in Section 4.3, two of the questions that could help holding the ML system of the real-world energy efficiency scenario accountable could be:

- Which is the performance obtained in the training of the model that forecasts the electric consumption of the building unit 02SX?
- Which is the algorithm and hyperparameters used by the model that forecast the electric consumption of the building unit 02SX?

If this scenario were no supported by FIDES, in order to answer the first question, an end-user would need to go through an Excel file where information related to each house is stored. Once the internal identifier of the building unit O2SX were retrieved, this information would be used to execute a specific SQL query in a PostgreSQL where this information were stored. This means that, not only the end-user needs to have SQL skills, but also, to be aware of the underlying data model used for representing the information. In contrast, with FIDES, the end-user would need to select the building unit 02SX from the list of houses displayed, and click in the 'Performance' button.

As for the second question, since this information is not being stored, the process to access the information would be more complicated. First of all, the end-user would need to retrieve the predictive model file belonging to the building unit 02SX. This model is stored in a folder within the Docker container from where it is periodically executed. Once this file is retrieved, it needs to be analysed in the R environment by executing different functions. This means that, for answering this question, the end-user needs to have R skills. On the contrary, FIDES allows the access to this information by simply clicking in the 'Algorithm' button of the corresponding building unit.

6. Conclusions

The current adoption, deployment and application of AI systems is not as wide as it could be expected, mainly due to a lack of trust of users. Nowadays, there are some scenarios where certain legal, ethical and technological compliance requirements must be satisfied and where the potential causes that may lead to undesirable outcomes must be identified. The ontology-based approach proposed in this article is expected to, on the one hand, address these needs and hold ML systems accountable, and on the other, contribute to helping in the overcoming of these adoption barriers.

FIDES is based on ontologies for representing, structuring and setting formal relations among the predictive models and the forecasts that conform a ML system, and provides end-users with the necessary means to exploit this knowledge for answering relevant questions for making such a ML system accountable. Furthermore, following the Semantic Web best practice, FIDES reuses existing ontologies that follow certain quality criteria to the extent possible.

The validity of FIDES has been demonstrated in a real-world energy efficiency scenario where more than 120 buildings participated. After evaluating the tool in such a demonstration scenario, it can be concluded that the overall usability of the system is good, that the current functionalities may satisfy most managers' requirements, and that the access to the information needed for holding systems accountable is much more straightforward compared with a traditional approach.

The potential of Semantic Technologies to fill existing gaps and address unsolved challenges towards trustworthy AI is high, even though it is not fully exploited yet. The contributions presented in this article try to, on the one hand, pave the way for future research in the usage of ontologies for holding AI systems accountable, and on the other, raise awareness of the possibilities of Semantic Technologies in different factors that may contribute to achieving trustworthy AI systems. Therefore, apart from accountability, the research of the Semantic Technologies as a whole for solving other related factors such as fairness, explainability or transparency is also of interest, and they should receive a bigger attention from the Semantic Web community.

6.1. Future Work

To the extent of knowledge of author, at the moment of writing this article, FIDES is the first tool that exploits Semantic Technologies towards holding ML systems accountable. Just as it happens in these situations, this tool has potential points for improvement that could help upgrading and enhance FIDES as a whole.

The feasibility of FIDES should be tested with ML systems aimed at solving complex multiobjective problems, where the outputs of some given ML systems are the input for other ML systems. It is possible that the interaction between different components of such systems may require from additional ontolog-

ical resources that are not covered by the ontologies currently considered in the presented approach. Furthermore, a wider variety of algorithms and libraries should also be tested.

Likewise, the new functionalities required by one of the testers should be considered for future versions of FIDES. Namely, the training data quality information. The quality of the training data determines whether it meets a standard set by the data scientist or not, and it can be measured in terms of its completeness, accuracy or conformity among others. For the purpose of representing this information, other ontologies such as the Data Quality Vocabulary[21] [59] or the DQTS (Data Quality for Time Series)[22] [60] ontology should be considered, as well as the integration of FIDES with other tools that automatically calculate such metrics [61].

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme project AI-PROFICIENT under grant agreement no. 957391. Furthermore, the Protégé resource has been used, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

[21] https://www.w3.org/TR/vocab-dqv
[22] https://w3id.org/dqts

Appendix A. RDF examples

This appendix shows the RDF representation of the examples used in the article. For the sake of understandability the Turtle serialisation format has been used.

```
@prefix : <http://example.com/> .
@prefix aff: <https://w3id.org/affectedBy#> .
@prefix cdt: <http://w3id.org/lindt/custom_datatypes#> .
@prefix eep: <https://w3id.org/eep#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rc: <https://w3id.org/rc#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:buildingUnit_02SX rdf:type aff:FeatureOfInterest.
:elecCons_02SX rdf:type aff:Quality.
:forecast_20201125T1100_elecCons_02SX rdf:type eep:Execution.
:forecaster_02SX rdf:type eep:Executor.
:process_02SX rdf:type eep:Procedure.

:elecCons_02SX aff:belongsTo :buildingUnit_02SX.
:forecast_20201125T1100_elecCons_02SX eep:onQuality :elecCons_02SX;
        eep:madeBy :forecaster_02SX;
        eep:usedProcedure :process_02SX;
        rc:hasGenerationTime "2020-11-25T07:00"^^xsd:dateTime;
        rc:hasTemporalContext "2020-11-25T11:00"^^xsd:dateTime;
        rc:hasSimpleResult "1113 W.h"^^cdt:energy.
```

Listing 3: RDF representation of the example scenario's forecast.

```
@prefix : <http://example.com/> .
@prefix cdt: <http://w3id.org/lindt/custom_datatypes#> .
@prefix mls: <http://www.w3.org/ns/mls#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:process_02SX rdf:type mls:Run;
    mls:executes :r_knn_02SX;
    mls:hasInput :inputData_02SX;
    mls:hasInput :feature00_02SX;
    mls:hasOutput :model_02SX;
    mls:hasOutput :modelEvaluation_02SX_metric.

:r_knn_02SX rdf:type mls:Implementation;
    mls:implements :caret_knn;
    mls:hasHyperParameter :r_knn_k_02SX;
        mls:hasQuality :r_knn_02SX_model_type_regression;

:caret_knn rdf:type mls:Algorithm;
    rdfs:label "knn"^^xsd:String;
    rdfs:comment "k-Nearest Neighbors"^^xsd:String.

:r_knn_k_02SX rdf:type mls:HyperParameter;
```

```
    rdfs:label "k"^^xsd:String.

:r_knn_02SX_model_type_regression rdf:type mls:ImplementationCharacteristic;
        rdfs:label "Regression"^^xsd:String.

:r_knn_k_7 rdf:type mls:HyperParameterSetting;
        mls:specifiedBy :r_knn_k_02SX.
        mls:hasValue "7"^^xsd:int.

:inputData_02SX rdf:type mls:Dataset;
    mls:hasQuality :inputData_02SX_obs.

:inputData_02SX_obs rdf:type mls:DataCharacteristic;
        rdfs:comment "obs."^^xsd:String;
        rdfs:comment "Number of observations"^^xsd:String;
        mls:hasValue "7423"^^xsd:int.

:feature00_02SX rdf:type mls:Feature;
    rdfs:label "sinmonth"^^xsd:String.

:model_02SX rdf:type mls:Model;
    rdfs:label "Model 02SX"^^xsd:String.

:modelEvaluation_02SX_metric rdf:type mls:ModelEvaluation;
    mls:specifiedBy :rmse;
    mls:hasValue "242.03 W.h"^^cdt:energy.

:rmse rdf:type mls:EvaluationMeasure;
        rdfs:label "RMSE"^^xsd:String.
```

Listing 4: RDF representation of the example scenario's Predictive Model process.

References

[1] S.S. ÓhÉigeartaigh, J. Whittlestone, Y. Liu, Y. Zeng and Z. Liu, Overcoming Barriers to Cross-cultural Cooperation in AI Ethics and Governance, *Philosophy & Technology* **33**, 571–593–. doi:10.1007/s13347-020-00402-x.

[2] M. Cubric, Drivers, barriers and social considerations for AI adoption in business and management: A tertiary study, *Technology in Society* **62** (2020), 101257. doi:https://doi.org/10.1016/j.techsoc.2020.101257.

[3] G. Baryannis, S. Validi, S. Dani and G. Antoniou, Supply chain risk management and artificial intelligence: state of the art and future research directions, *International Journal of Production Research* **57**(7) (2019), 2179–2202. doi:10.1080/00207543.2018.1530476.

[4] E. Broadbent, R. Stafford and B. MacDonald, Acceptance of healthcare robots for the older population: review and future directions, *International journal of social robotics* **1**(4) (2009), 319. doi:10.1007/s12369-009-0030-6.

[5] L. Laranjo, A.G. Dunn, H.L. Tong, A.B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A.Y. Lau et al., Conversational agents in healthcare: a systematic review, *Journal of the American Medical Informatics Association* **25**(9) (2018), 1248–1258. doi:10.1093/jamia/ocy072.

[6] J.T.M. Ingibergsson, U.P. Schultz and M. Kuhrmann, On the use of safety certification practices in autonomous field robot software development: A systematic mapping study, in: *International Conference on Product-Focused Software Process Improvement*, Springer, 2015, pp. 335–352. doi:10.1007/978-3-319-26844-6_25.

[7] M. Raghavan, S. Barocas, J. Kleinberg and K. Levy, Mitigating bias in algorithmic hiring: Evaluating claims and practices, in: *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 469–481. doi:10.1145/3351095.3372828.

[8] J. Sánchez-Monedero, L. Dencik and L. Edwards, What Does It Mean to 'solve' the Problem of Discrimination in Hiring? Social, Technical and Legal Perspectives from the UK on Automated Hiring Systems, in: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 458–468–. ISBN 9781450369367. doi:10.1145/3351095.3372849.

[9] S. Chuang and C.M. Graham, Embracing the sobering reality of technological influences on jobs, employment and human resource development, *European Journal of Training and Development* (2018). doi:10.1108/EJTD-03-2018-0030.

[10] M. Madsen and S. Gregor, Measuring human-computer trust, in: *11th Australasian conference on information systems (ACIS)*, Vol. 53, 2000, pp. 6–8.

[11] J.-Y. Jian, A.M. Bisantz and C.G. Drury, Foundations for an empirically determined scale of trust in automated systems, *International journal of cognitive ergonomics* **4**(1) (2000), 53–71. doi:10.1207/S15327566IJCE0401_04.

[12] B. Cahour and J.-F. Forzy, Does projection into use improve trust and exploration? An example with a cruise control system, *Safety science* **47**(9) (2009), 1260–1270. doi:10.1016/j.ssci.2009.03.015.

[13] D. Gunning, Explainable artificial intelligence (xai), *Defense Advanced Research Projects Agency (DARPA), nd Web* **2**(2) (2017).

[14] S.T. Mueller, R.R. Hoffman, W. Clancey, A. Emrey and G. Klein, Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI, *arXiv preprint arXiv:1902.01876* (2019).

[15] J. van der Waa, E. Nieuwburg, A. Cremers and M. Neerincx, Evaluating XAI: A comparison of rule-based and example-based explanations, *Artificial Intelligence* (2020), 103404. doi:10.1016/j.artint.2020.103404.

[16] J. Fox, The uncertain relationship between transparency and accountability, *Development in practice* **17**(4–5) (2007), 663–671. doi:10.1080/09614520701469955.

[17] J.A. Kroll, S. Barocas, E.W. Felten, J.R. Reidenberg, D.G. Robinson and H. Yu, Accountable algorithms, *U. Pa. L. Rev.* **165** (2016), 633–705.

[18] V. Beaudouin, I. Bloch, D. Bounie, S. Clémençon, F. d'Alché-Buc, J. Eagan, W. Maxwell, P. Mozharovskyi and J. Parekh, Flexible and context-specific AI explainability: a multidisciplinary approach, *Available at SSRN 3559477* (2020). doi:10.2139/ssrn.3559477.

[19] D. Oberle, How ontologies benefit enterprise applications **5**(6) (2014), 473–491. doi:10.3233/SW-130114.

[20] S.J. Russell and P. Norvig, *Artificial intelligence: A Modern Approach*, Prentice Hall, 2009. ISBN 978-0-13-604259-4.

[21] A. Seeliger, M. Pfaff and H. Krcmar, Semantic Web Technologies for Explainable Machine Learning Models: A Literature Review, in: *Joint Proceedings of PROFILES 2019 and SEMEX 2019, 1st Workshop on Semantic Explainability (SemEx 2019), co-located with the 18th International Semantic Web Conference (ISWC '19)*, PROFILES-SEMEX 2019, Vol. 2465, CEUR-WS, 2019, pp. 30–45. ISSN 1613-0073. http://ceur-ws.org/Vol-2465/semex_paper1.pdf.

[22] R. Confalonieri, T. Weyde, T.R. Besold and F.M. del Prado Martın, TREPAN Reloaded: A Knowledge-driven Approach to Explaining Black-box Models **325** (2020), 2457–2464. doi:10.3233/FAIA200378.

[23] S. Chari, O. Seneviratne, D.M. Gruen, M.A. Foreman, A.K. Das and D.L. McGuinness, Explanation Ontology: A Model of Explanations for User-Centered AI, in: *Lecture Notes in Computer Science*, Vol. 12507, Springer, 2020, pp. 228–243. doi:10.1007/978-3-030-62466-8_15.

[24] C. Panigutti, A. Perotti and D. Pedreschi, Doctor XAI: an ontology-based approach to black-box sequential data classification explanations, in: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 629–639. doi:10.1145/3351095.3372855.

[25] F. Lécué, J. Chen, J.Z. Pan and H. Chen, *Knowledge-Based Explanations for Transfer Learning*, in: *Studies on the Semantic Web*, Vol. Volume 47: Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, IOS Press, 2020, pp. 180–195. doi:10.3233/SSW200018.

[26] F. Lécué and J. Wu, Semantic explanations of predictions, *arXiv preprint arXiv:1805.10587* (2018).

[27] P. Hitzler, F. Bianchi, M. Ebrahimi and M.K. Sarker, Neural-symbolic integration and the Semantic Web, *Semantic Web* **11**(1) (2020), 3–11. doi:10.3233/SW-190368.

[28] F. Bianchi, G. Rossiello, L. Costabello, M. Palmonari and P. Minervini, *Knowledge Graph Embeddings and Explainable AI*, in: *Studies on the Semantic Web*, Vol. Volume 47: Knowledge Graphs for eXplainable Artificial Intelligence: Founda-

tions, Applications and Challenges, IOS Press, 2020, pp. 49–72. doi:10.3233/SSW200011.

[29] S. Moon, P. Shah, A. Kumar and R. Subba, Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 845–854. doi:10.18653/v1/P19-1081.

[30] J. Huang, W.X. Zhao, H. Dou, J.-R. Wen and E.Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, in: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 505–514. doi:10.1145/3209978.3210017.

[31] F. Lecue, On the Role of Knowledge Graphs in Explainable AI, *Semantic Web* **11**(1) (2020), 41–51. doi:10.3233/SW-190374.

[32] S. Chari, D.M. Gruen, O. Seneviratne and D.L. McGuinness, *Foundations of Explainable Knowledge-Enabled Systems*, in: *Studies on the Semantic Web*, Vol. Volume 47: Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, IOS Press, 2020, pp. 23–48. doi:10.3233/SSW200010.

[33] H.L.E.G.o.A.I. HLEG-AI, Ethics Guidelines for Trustworthy AI, 2019, last visited on 2021-02-08. https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

[34] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* **58** (2020), 82–115. doi:10.1016/j.inffus.2019.12.012.

[35] I. Esnaola-Gonzalez, Semantic Technologies Towards Accountable Artificial Intelligence: A Poultry Chain Management Use Case (2020), 215–226. ISBN 978-3-030-63799-6. doi:10.1007/978-3-030-63799-6_17.

[36] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, *The NeOn Methodology for Ontology Engineering*, in: *Ontology Engineering in a Networked World*, M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 9–34. ISBN 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1_2.

[37] M. Poveda-Villalón, A. Fernández-Izquierdo and R. García-Castro, Linked Open Terms (LOT) Methodology, Zenodo, 2019. doi:10.5281/zenodo.2539305.

[38] E. Simperl, Reusing ontologies on the Semantic Web: A feasibility study, *Data & Knowledge Engineering* **68**(10) (2009), 905–925. doi:10.1016/j.datak.2009.02.002.

[39] M. Fernández-López, M.C. Suárez-Figueroa and A. Gómez-Pérez, *Ontology Development by Reuse*, in: *Ontology Engineering in a Networked World*, M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 147–170. ISBN 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1_7.

[40] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, Ontologies for Observations and Actuations in Buildings: A Survey, *Semantic Web* **11**(4) (2020), 593–621. doi:10.3233/SW-200378.

[41] A. Haller, K. Janowicz, S. Cox, M. Lefrançois, K. Taylor, D.L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson and C. Stadler, The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation, *Semantic Web* **10**(1) (2019), 9–32. doi:10.3233/SW-180320.

[42] K. Janowicz, A. Haller, S.J. Cox, D.L. Phuoc and M. Lefrançois, SOSA: A lightweight ontology for sensors, observations, samples, and actuators, *Journal of Web Semantics* **56** (2019), 1–10. doi:10.1016/j.websem.2018.06.003.

[43] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, EEPSA as a core ontology for energy efficiency and thermal comfort in buildings, *Applied Ontology* **16**(2) (2021), 193–228. doi:10.3233/AO-210245.

[44] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, Semantic prediction assistant approach applied to energy efficiency in Tertiary buildings, *Semantic Web* **9**(6) (2018), 735–762. doi:10.3233/SW-180296.

[45] I. Esnaola-Gonzalez, I. Fernandez, E. García, S. Ferreiro, M. Gomez, I. Lázaro and A. García, Towards Animal Welfare in Poultry Farms through Semantic Technologies, in: *IoT Connected World & Semantic Interoperability Workshop (IoT-CWSI) 2019*, 2019. https://www.researchgate.net/publication/336848367_Towards_Animal_Welfare_in_Poultry_Farms_through_Semantic_Technologies.

[46] S. Borgo and C. Masolo, Foundational choices in DOLCE, in: *Handbook on ontologies*, Springer, 2009, pp. 361–381. doi:10.1007/978-3-540-92673-3_16.

[47] S. Cox, Ontology for observations and sampling features, with alignments to existing models, *Semantic Web* **8**(3) (2016), 453–470. doi:10.3233/SW-160214.

[48] P. Panov, L. Soldatova and S. Džeroski, Ontology of core data mining entities, *Data Mining and Knowledge Discovery* **28**(5–6) (2014), 1222–1265. doi:10.1007/s10618-014-0363-0.

[49] C.M. Keet, A. Ławrynowicz, C. d'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens and M. Hilario, The data mining optimization ontology, *Journal of web semantics* **32** (2015), 43–53. doi:10.1016/j.websem.2015.01.001.

[50] G.C. Publio, D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren and H. Zafar, ML-Schema: Exposing the Semantics of Machine Learning with Schemas and Ontologies, 2018.

[51] D. Garijo and M. Poveda-Villalón, *Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web*, in: *Applications and Practices in Ontology Design, Extraction, and Reasoning*, G. Cota, M. Daquino and G.L. Pozzato, eds, Studies on the Semantic Web, Vol. 49, IOS Press, 2020, pp. 39–54. ISBN 978-1-64368-142-9. doi:10.3233/SSW200034.

[52] D. Esteves, D. Moussallem, C.B. Neto, T. Soru, R. Usbeck, M. Ackermann and J. Lehmann, MEX vocabulary: a lightweight interchange format for machine learning experiments, in: *SEMANTICS 15': Proceedings of the 11th International Conference on Semantic Systems*, 2015, pp. 169–176. doi:10.1145/2814864.2814883.

[53] J. Brooke et al., *SUS-A quick and dirty usability scale*, in: *Usability evaluation in industry*, Vol. 189, CRC Press, 1996, pp. 4–7. ISBN 9780429157011.

[54] G. Widmer and M. Kubat, Learning in the presence of concept drift and hidden contexts, *Machine learning* **23**(1) (1996), 69–101. doi:10.1007/BF00116900.

[55] P. Lusis, K.R. Khalilpour, L. Andrew and A. Liebman, Short-term residential load forecasting: Impact of calendar effects and forecast granularity, *Applied Energy* **205** (2017), 654–669. doi:10.1016/j.apenergy.2017.07.114.

[56] M. Gomez-Omella, I. Esnaola-Gonzalez and S. Ferreiro, Short-term Forecasting Methodology for Energy Demand in Residential Buildings and the Impact of the COVID-19 Pandemic on Forecasts (2020), 227–240. ISBN 978-3-030-63799-6. doi:10.1007/978-3-030-63799-6_18.

[57] E. Petrova, P. Pauwels, K. Svidt and R.L. Jensen, 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*, Springer, 2019, pp. 19–26. doi:10.1007/978-3-030-00220-6$_3$.

[58] I. Esnaola-Gonzalez and F.J. Diez, Integrating Building and IoT data in Demand Response solutions, in: *Proceedings of the 7th Linked Data in Architecture and Construction Workshop (LDAC 2019)*, Vol. 2389, CEUR-WS, 2019, pp. 92–105. http://ceur-ws.org/Vol-2389/07paper.pdf.

[59] R. Albertoni and A. Issac, Introducing the Data Quality Vocabulary (DQV), *Semantic Web* **12**(1) (2021), 81–97. doi:10.3233/SW-200382.

[60] I. Esnaola-Gonzalez, Towards publishing ontology-based data quality metadata of Open Data, in: *Proceedings of SGAI 2021: Artificial Intelligence XXXVIII*, 2021.

[61] M. Gómez-Omella, B. Sierra and S. Ferreiro, On the Evaluation, Management and Improvement of Data Quality in Streaming Time Series, *Advanced Engineering Informatics* (To-be-published).