Web Semantics: Science, Services and Agents on the World Wide Web 79 (2023) 100808

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

FIDES: An ontology-based approach for making machine learning systems accountable

Izaskun Fernandez [a,*], Cristina Aceta [a], Eduardo Gilabert [a], Iker Esnaola-Gonzalez [b]

[a] TEKNIKER, Basque Research and Technology Alliance (BRTA), Parke Teknologikoa, c/Iñaki Goenaga, 5, Eibar, 20600, Spain
[b] BASF Digital Solutions S.L., P. de la Castellana, 77, Planta 14, Madrid, 28046, Spain

ARTICLE INFO

Keywords:
Accountability
Ontology
Trustworthy artificial intelligence
Machine learning

ABSTRACT

Although the maturity of technologies based on Artificial Intelligence (AI) is rather advanced nowadays, their adoption, deployment and application are not as wide as it could be expected. This could be attributed to many barriers, among which the lack of trust of users stands out. Accountability is a relevant factor to progress in this trustworthiness aspect, as it allows to determine the causes that derived a given decision or suggestion made by an AI system. This article focuses on the accountability of a specific branch of AI, statistical machine learning (ML), based on a semantic approach. FIDES, an ontology-based approach towards achieving the accountability of ML systems is presented, where all the relevant information related to a ML-based model is semantically annotated, from the dataset and model parametrisation to deployment aspects, to be exploited later to answer issues related to reproducibility, replicability, definitely, accountability. The feasibility of the proposed approach has been demonstrated in two scenarios, real-world energy efficiency and manufacturing, and it is expected to pave the way towards raising awareness about the potential of Semantic Technologies in different factors that may be key in the trustworthiness of AI-based systems.

1. Introduction

Despite the fact that Artificial Intelligence (AI) is a field that has reached an advanced stage of maturity nowadays, its adoption, deployment and application is not as wide as it could be expected [1]. This could be attributed to many barriers such as cultural, economic, technical and social [2,3] . As for the latter, the lack of trust of potential end users in AI systems is remarkable [4,5]. As a matter of fact, there are many concerns that derive from this lack of trust, such as potential safety issues that may lead to harm humans [6,7] and biases towards the penalisation of certain social groups [8–10]. However, this lack of trust, if carefully managed, can be overcome, thus contributing to the acceptance of AI systems [3].

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" [11]. There are many factors that affect this lack of trust [12,13], including 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" [14]. XAI was intensively studied from the 1970s to the 1990s [15], although a resurgence of the topic has been seen recently due to the current technological advancements in various disciplines of AI [16].

Explainability is necessary but far from sufficient for achieving trustworthiness in AI systems. In order to do so, not only should the developed AI systems be explainable, but also accountable [17,18]. As a matter of fact, the ability to hold them accountable by explaining their inner workings, their results and the causes of failure to users, regulators and citizens, is critical to achieve trust [19].

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 [18]. This means that, with an accountable AI system, the causes that led to 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 would 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 being used to train the AI system or how well the AI system performs in terms of accuracy).

Nevertheless, the information needed to answer these questions is hardly ever accessible in a straightforward way [20]. This information

* Corresponding author.
E-mail address: izaskun.fernandez@tekniker.es (I. Fernandez).

https://doi.org/10.1016/j.websem.2023.100808
Received 29 March 2023; Received in revised form 28 July 2023; Accepted 26 October 2023
Available online 4 November 2023

is often 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 dealing with the AI system wanted to answer the aforementioned questions, it would be a very time-consuming task, as they would have to be an expert or have the help of experts in different frameworks, systems, data models, repositories and query languages. Therefore, the regular performance of these accountancy tasks, thus, would be infeasible.

Taking the above into account, 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 is 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 [21].

Since AI is a field that comprises a variety of disciplines ranging from natural language processing to knowledge representation [22], this article focuses on a specific branch: statistical machine learning (ML, from now on). FIDES, an ontology-based approach towards achieving the accountability of ML systems, is proposed. The approach relies on annotating all the relevant information related to a ML-based model semantically, from the dataset and model parametrisation to deployment aspects, according to FIDES ontology, to be exploited later on to answer issues related to reproducibility, replicability, in summary, accountability aspects.

The rest of the article is structured as follows. Section 2 presents the related work. FIDES, the proposed semantic approach, is described in Section 3, demonstrated through a energy efficiency and two manufacturing real-world scenarios in Section 4, and evaluated and discussed in Section 5. 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.

To the extent of knowledge of the authors so far, the main focus of the usage of Semantic Technologies has been placed on explainability [23–26], although accountability is considered a key requirement that should be met to achieve trustworthy AI systems [27,28]. Thus, this section will refer to both aspects, although the focus in this paper is directed towards accountability.

As for explainability, [29] provide a literature-based overview of the usage of Semantic Technologies alongside ML methods in order to facilitate their explainability. According to this source, the main role of 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 fields.

In [30], 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, would be a significant contributor to hybrid explainable AI systems.

In [31], the *Explainable ML* ontology to represent explainable ML experiments is presented, enhancing a better understanding of the ML process while improving its explainability. More specifically, this work focuses on a post-hoc approach that represents relevant information about the whole ML-based model development, such as, the data used to train (including details on pre-processing), the selected algorithm or the evaluations, and its output. This ontology covers many critical accountability aspects, including the ones noted above. However, due to its orientation towards explainability, there are some aspects that remain unaddressed, like the knowledge on development and deployment environments.

In terms of accountability, [32] makes a first contribution on the usage of ontologies to support the accountability of ML systems, proposing a method that allows 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 ML systems accountable remain unaddressed, such as the description of the procedure followed to develop the ML-based models, even the examination of their correct functioning [33].

There are other recent works based on Semantic Technologies that deal with accountability aspects, but are more focused on supporting the definition and management of accountability plans – including relevant accountability information –, although limited to the design stage of AI systems [34] or, in the case of [35], on the data accountability through the WellFort approach. This approach provides a semantic-enabled architecture for auditable, privacy-preserving data analysis through a secure storage for users' sensitive data with explicit consent and the collection of sufficient information in an automated way to support auditability at the same time.

All this evidence reinforces the hypothesis that Semantic Technologies could play an important role in achieving trustworthy AI systems in general and in solving the accountability challenge for ML systems in particular in all stages: from design to exploitation in production environments.

3. FIDES: Making machine learning systems accountable

Towards the achievement of accountable ML systems, this article presents FIDES.[1] FIDES is a framework that leverages ontologies for representing, structuring and setting formal relations among the procedures for developing and deploying ML-based models and the estimations obtained from those models . In this sense, end users are thus provided with all the necessary mechanisms to easily deal with accountability-related issues regarding ML-based models in all stages, starting from the generation of relevant accountability-related information to model exploitation.

The core element of FIDES is the FIDES ontology, which enables the representation of all the relevant aspects of ML-based models for their accountability in a human- and machine-readable format. Furthermore, it adds reasoning capabilities for the exploitation phase of this model-related data.

3.1. FIDES ontology

The FIDES ontology aims to be a representational framework that ensures and enhances the accountability of ML-based models to, on the one hand, easily identify the potential causes of undesirable outcomes of AI systems and, on the other, the evaluation and assurance that AI systems are legally, ethically and technologically robust while respecting democratic values, such as, human rights and the rule of law, as requested by the EU Artificial Intelligence Act.[2]

For developing the FIDES ontology, the Linked Open Terms (LOT) methodology has been used. This methodology follows two main steps for development – *requirements specification* and *implementation* –, alongside two additional steps for ensuring the ontology's *publication* and *maintenance* aspects. The aim of the *requirements specification* process is to state why the ontology is being built and to identify and define the requirements the ontology should fulfil. Taking as input the documentation and data provided by domain experts and users, a set of ontological requirements – written in the form of competency questions (CQs) or statements – is generated by the ontology development team. In the

[1] Fides was the Roman goddess of trust.

[2] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206&from=EN

implementation stage, the ontology is built using a formal language according to the requirements identified by the domain experts, while emphasizing on the reuse of existing ontologies whenever it is possible. In this phase, an evaluation of relevant aspects of the ontology, such as, its syntax, modelling and semantics, along with a revision on how the ontology fulfils all the requirements, is also performed. The *publication* process aims to ensure that the ontology is accessible through both human-readable, comprehensive documentation and a machine-readable format from its URI and, finally, the *maintenance* phase ensures that the ontology is updated when new requirements appear or errors are identified.

3.1.1. FIDES ontology requirements

The FIDES ontology envisions the representation of three main knowable topics for ensuring the accountability of ML-based models: the *procedure* followed to construct a ML-based model, the *deployment* aspects to put it into production and the *estimations* made by the ML-based model itself. To formalise the information requirements for each knowable topic (as proposed by LOT), a set of CQs was defined by a team of 4 data scientists, experts in ML-methods, and 2 ontologists.

For the procedure followed to construct ML-based model for making forecasts, the team established that the relevant information could be divided in, on the one hand, the information regarding the data used to train the ML-based model and, on the other, the information concerning the details of the procedure implemented by the ML-based model. More specifically, the latter deals with the characterisation of the training data in terms of features such as the amount of data used, the dependent and independent variables considered or statistical characteristics (e.g. the variance, mean or median of the data). Some CQ examples on this topic are listed below.

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

Regarding the ML-based model's procedure details, information related to the algorithm used and its hyperparameters was identified as relevant, as well as the performance assessed in development time. This can be procured by CQs such as the following:

- Which is the base algorithm of the ML-based model?
- Which are the hyperparameter values of the ML-based model?
- Which is the validation metric used and its value (e.g. Which is the root-mean-square error (RMSE) of the ML-based model?)

For the second knowable topic, the deployment aspects, the team established that the characteristics of the server where the model is deployed and the model's strategy for execution triggering and result storage are crucial for accountability purposes. The following CQs are a set of examples related to this topic:

- Which is the operative system of the deployment server?
- Where are the estimations (derived from the ML-based model executions) stored?
- How is the model triggered, on event or under request?

For the last topic, the estimations made by a ML-based model, the team determined that both the details of the estimation and when did it occur should be available. The following CQs are some examples in this regard:

- Which is the value of a given estimation?
- When was a given estimation generated?
- What is the estimation's error metric/value?

After several rounds, a total of 37 CQs – a set of 28 CQs for model development procedures, 4 for the deployment aspects and 5 CQs for the estimations part – were established as the starting CQs set. The complete list can be consulted in Appendix B.

3.1.2. FIDES ontology implementation

Starting from the 37 CQs from the requirements specification step, a list of 20 terms was created to represent the main concepts that should be present in the FIDES ontology. The terms are listed below.

- *Software*
- *Version*
- *Creator*
- *Contributor*
- *Docker container*
- *Source*
- *Feature*
- *Response feature*
- *Run*
- *Input*
- *Dataset characteristic*
- *Operating System*
- *Procedure*
- *Triggered on*
- *Stores*
- *Prediction value*
- *Prediction error value*
- *Generation time*
- *Temporal context*
- *Result*

Considering those terms, and following the ontology reuse best practices in [36], the research for potential ontologies to be reused was carried out by consulting different resources: the LinkedOpenVocabularies(LOV) and LOV4IoT ontology catalogues and the Google Scholar and ScienceDirect research databases. The approach for selecting the potential ontologies to be reused was inspired by the Ontological Resource Reuse Process [37], and the following set of ontology quality criteria defined by [38] 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.

Ontologies for Estimations and Procedures

Currently, there are many ontologies that could be used for representing events or activities, the result of which is an estimate of the value of a quality of a feature of interest that is obtained using a specific procedure.

A thorough analysis of ontologies covering such a domain can be found in [38], and it has been observed that the SOSA/SSN ontology,[3] proposed by [39,40], may be one of the most appropriate ontologies for representing forecasts due to its comprehensive 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 [41] and, therefore, was discarded for the FIDES ontology.

Instead, the EEPSA ontology,[4] proposed by [42], was selected to be reused, as it was developed on the basis that a proper axiomatisation shapes the set of admitted models better and, thus, 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 [43], the backbone of the EEPSA ontology is defined as a combination of three Ontology Design Patterns (ODP) — the AffectedBy ODP,[5] the Execution-Executor-Procedure

[3] http://www.w3.org/ns/ssn
[4] http://w3id.org/eepsa
[5] https://w3id.org/affectedBy

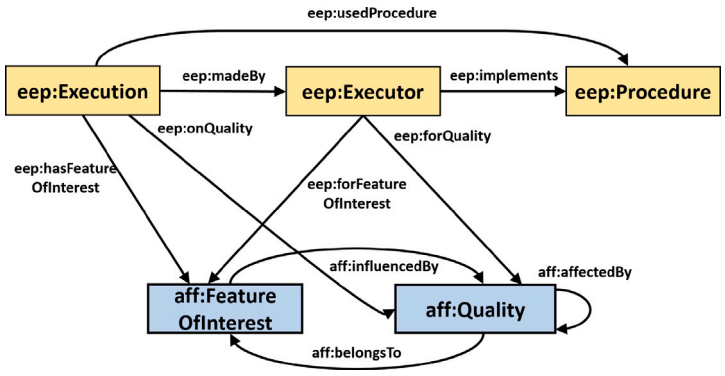

Fig. 1. The main classes and properties of the AffectedBy and EEP ODPs used for the annotation of forecasts.

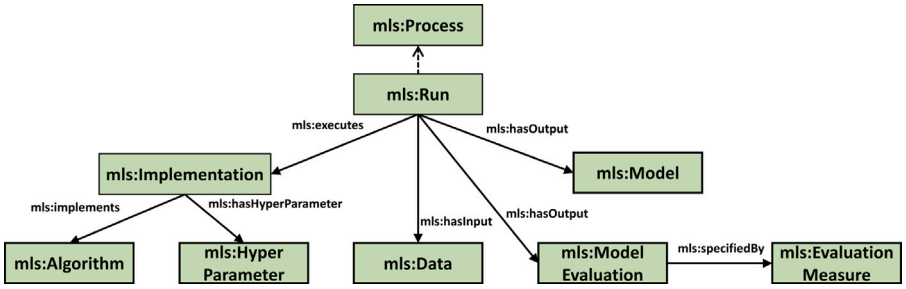

Fig. 2. The main classes and properties of the ML-Schema used for the annotation of predictive models.

(EEP) ODP[6] and the Result-Context (RC) ODP[7]– that can be used as basic building blocks to address similar problems in different domains, dealing with estimations of qualities of some features of interest following specific procedures.

More specifically, 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* to indicate the relations among both classes. This ODP is imported into the EEP ODP (a graphical representation of the interactions between both ODPs can be seen in Fig. 1), which defines three more classes: *eep:Execution*, *eep:Executor*, and *eep:Procedure* for representing events (equivalent to an estimation action), agents (comparable to a ML-based model) and procedures (development and deployment aspects) in charge of estimating the values upon a quality of a feature of interest, respectively. Finally, the RC ODP aims to represent the results of the executions defined in the EEP ODP as well as their contexts.

These three ODPs are published in the ODP repository OntologyDesignPatterns.org[8] 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[9] to ensure clarity in modelling and avoid errors that may have unintended reasoning implications [44]. Hence, the ontology quality criteria established in Section 3.1.2 are satisfied.

Ontologies for the Machine Learning Domain

The existing ontologies in the ML domain and, more specifically, for the development and deployment aspects of ML-based models are not as abundant as for the previous topic. Nevertheless, there are still some ontologies that cover ML experiments and different areas of data mining, such as the OntoDM-core ontology described in [45] or the DMOP ontology presented in [46]. However, there is a gap between these ontologies that would affect the interoperability between both. Towards reducing such a gap and achieving a higher level of interoperability among those resources, the ML-Schema ontology[10] [47] was developed within the W3C Machine Learning Schema Community Group.[11] This ontology, the main classes and relationships of which can be seen in Fig. 2, includes resources to describe, on the one hand, the data used as input and their characteristics and quality and, on the other hand, the implementations, algorithms used to develop models and their hyperparameters. Furthermore, the developed models, their characteristics and the evaluation obtained in the training phase can also be represented with this ontology. ML-Schema is published in the LOV catalogue and it is available online with a W3C Community Contributor License Agreement. Although according to the guidelines proposed by [48],[12] the metadata associated to the resources described in the ontology are incomplete,[13] impacting in the reuse of the ontology negatively, it has a complete documentation page that softens this issue.

Considering the remarks above, ML-Schema allows to specialise the generic procedures in EEP for the ML domain – which is of special relevance in the context of FIDES – and, for this reason, this ontology was finally selected to be reused for the FIDES ontology.

To sum up, the ontologies leveraged by the FIDES ontology for the representation of the relevant information are, on the one hand, the AffectedBy, the EEP and the RC ODPs for estimations and generic aspects about how they have been obtained and, on the other hand, ML-Schema for representing more detailed procedures for ML-based models. The alignment between these ontologies is rather straightforward thanks to their design with a view to be easily extended and complemented with

[6] https://w3id.org/eep
[7] https://w3id.org/rc
[8] http://ontologydesignpatterns.org/
[9] https://www.w3.org/TR/prov-o/
[10] http://www.w3.org/ns/mls
[11] https://www.w3.org/community/ml-schema/
[12] The most complete ontology metadata guidelines to date.
[13] An issue related to this matter is opened at the moment of writing this article in https://github.com/ML-Schema/core/issues/25.

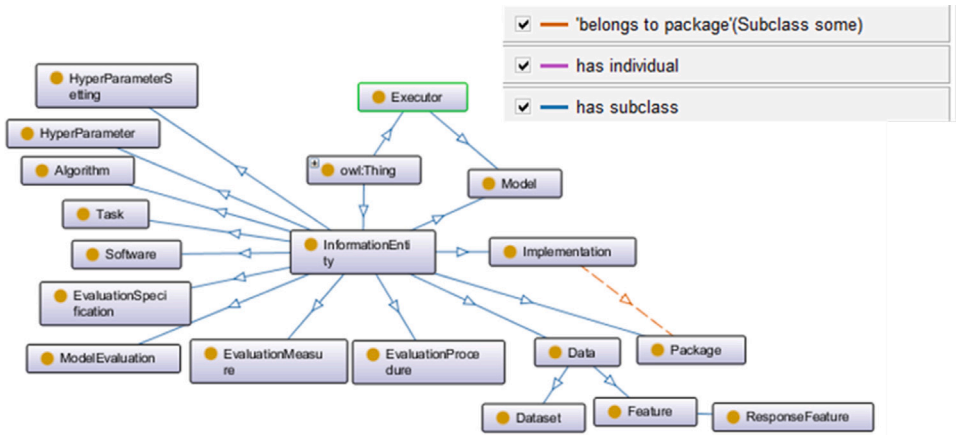

Fig. 3. FIDES ontology excerpt.

other ontological resources. As a matter of fact, two RDF triples suffice to integrate the aforementioned ontologies:

mls:Process ⊑ *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 ⊑ *eep:Executor*

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

With these resources, an important part of the necessary concepts and relations for the FIDES ontology were covered. As for the necessary classes or properties that were not covered by the selected ontologies, those were modelled and properly described in the FIDES ontology. The result is an ontology with 43 classes, 32 object properties and 34 data properties. An excerpt can be seen in Fig. 3.

Once the FIDES ontology was complete, it was evaluated with OOPS! [49] and FOOPS! [50] to monitor potential pitfalls and to determine its FAIRness, respectively. In both cases, the results obtained were positive, as no pitfalls were detected by OOPS![14] and the score obtained in FOOPS! was a 73%.[15]

The FIDES ontology, along with its documentation, is available in https://w3id.org/fides, and it has been submitted for revision to be included in LOV. For the generation of the documentation, the WIDOCO [51] tool was used.

3.2. FIDES at a glance

With the FIDES ontology as core component, this semantic approach consists of three phases: (i) ML-based model development and deployment, (ii) information regarding semantic annotation and storage and (iii) data exploitation. The first phase is related to the development of the ML-based model that will solve the problem at hand, which will be deployed to get the desired estimations. For a proper development of ML-based models, the data scientist, depending on the type of the problem to address and the quality and the amount of data available, should test different algorithms with the adequate fine-tuning of their hyperparameters and select the optimal configuration for the given problem and context. All this parametrisation and configuration information, together with the estimations that the model will be providing in the production environment and the corresponding contextual information, such as, the time when the estimation was produced, is crucial for the accountability of the models, as it has been shown in the CQs described in Section 3.1.1.

For that matter, the second phase is in charge of, on the one hand, semantically annotating the relevant information collected during the first phase according to the FIDES ontology and, on the other hand, publishing it in an RDF store to be accessible for its later exploitation. For FIDES, Openlink Virtuoso[16] is the selected repository for the storage of semantic information.

Each time a new model is created and deployed in a production environment, a new RDF will be created and uploaded to the Virtuoso repository. To automate this process as much as possible, and to make the semantic representation issues transparent to data scientists, a CSV template has been defined to gather the necessary accountability-related information, which will be automatically translated to RDF according to the FIDES ontology and published in the Virtuoso repository through a Python service, as it is shown in Fig. 4. Thus, the only task that data scientists must perform at this step is to provide for any model its corresponding filled-out CSV. This could be done manually or even through the ML-based model training script, according to the expert preferences. More specifically, and regarding the latter, many algorithm implementations (provided by different libraries in languages such as R or Python) allow to easily export model-related information, e.g., hyperparameters, which can be used to fill the CSV file. For example, for the R language, a function that exploits the information from the Caret package [52] has been implemented for this CSV data exportation.

For the instantiation of the information related to the estimations, a similar approach to the one for static information is followed: each time an estimation is produced by the model, its details are dumped into a previously-defined CSV template. By using a Python service, this information will be automatically converted to RDF according to the FIDES ontology and published in the Virtuoso repository, together with the rest of the model accountability information.

Finally, in the data exploitation phase, end users are able to consume the accountability information in the RDF repository through a Graphical User Interface (GUI). This GUI makes use of a REST API that includes methods that answer the main CQs identified by the experts in the field, which in turn execute a set of predefined parametric SPARQL queries over the Virtuoso endpoint, abstracting end users from the underlying semantic query language. In order to simplify the REST API, the different CQs have been grouped into 5 categories,[17] which in turn result in the same number of SPARQL queries that return all the information related to that classification. Following these categories,

[14] In fact, the tool returns a critical pitfall which is detected by an external tool that, in this case, is a false positive.

[15] This score will substantially improve when the ontology is published in the LOV repository. At the moment of writing this paper, FIDES has been submitted for revision.

[16] https://virtuoso.openlinksw.com/

[17] The complete list of CQs and their corresponding category are detailed in Appendix B.

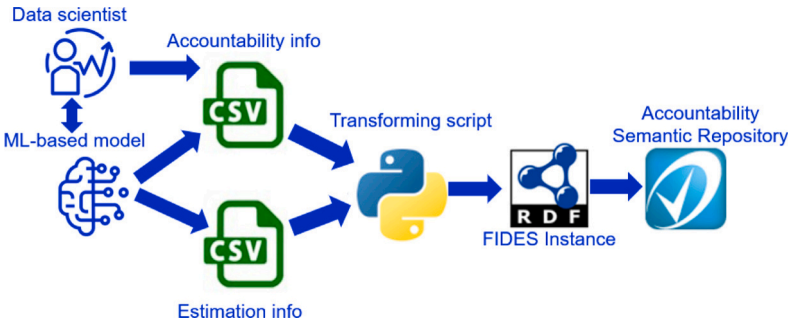

Fig. 4. FIDES semantic annotation automatic process.

every SPARQL query has been encapsulated into a specific method in the REST API, as well as the main functionalities in the GUI:

- ***Development***. Details about the model development environment and the developer.
- ***Model***. Detailed information about the model and associated parameters.
- ***Data***. Information about the data used to train the model.
- ***Deployment***. Model in production details.
- ***Execution***. Model estimations including the corresponding contextual information.

A future version of the FIDES application will include a SPARQL endpoint for advanced users, giving the possibility of executing more precise, specific SPARQL queries.

4. FIDES in use

In order to illustrate the validity of FIDES, it has been applied in a real-world energy efficiency scenario and two manufacturing scenarios in a plant of CONTINENTAL (that is, related to tire manufacturing): obtaining information on when to restart an extrusion process (and with which parameters) and finding the optimal moment for blade replacement in cutting machines.

In the energy efficiency scenario, a predictive model that forecast the electric demand of the next 24 h was developed for each of the total of 122 residential and small commercial buildings involved, which were located in the island of Lanzarote (Spain). Then, the generated forecasts would be used as an input for an overall energy efficiency solution, which is out of scope in this work.

As for the manufacturing scenarios, and as noted above, they are related to extrusion processes. The extrusion process consists on mixing a set of different materials, obtaining tire treads as a result. Every time a process needs to be set up (due to the usage of new recipes or because the process was stopped for some reason), it is necessary to bring the production line back to an optimal production performance situation for which some adjustments are required, which is known as the *set-up process*. Until this production-ready point is reached, the tread that is being produced tends to be of low quality and therefore not useful. With this, the first model is aimed to identify the end of the set-up process and/or the optimal point of production or readiness of the extrusion process, while the second is intended to estimate the optimal values of the parameters involved in supporting the adjustment.

The input information for these models includes different types of data, such as the extrusion machine signals or the recipe to perform the extrusion process, detailing the corresponding compounds and their percentages. For the second scenario, the model to be developed was intended to inform the operator about the optimal moment for a blade change considering different parameters such as the number of cuts done, the material and components to cut or signals coming from the machine monitoring. The estimations obtained by all three models are shown to the operators through different interfaces and formats (traffic lights, control panels, etc.).

Regardless of the scenario, all these estimations had to be accountable as it was an explicit requisite of the solutions they were part of. With a classic approach, the information describing the relationship between the ML-based models and the system to which they correspond (building units for the energy efficiency scenario and the machine for the manufacturing ones), in the best of cases, if it were to be collected, would be stored in an Excel or similar file. As for the estimations, since they are typically stored in relational databases, specific SQL queries would be executed for their retrieval. Likewise, ideally, some minor details of the ML-based models, such as its performance, would be stored and updated in an Excel file as well. Finally, other model information is rarely collected and, thus, different functions would need to be executed, if known, in the development framework (R, Python, etc.) over each of them to obtain this information.

Therefore, it is evident that achieving accountability in the described scenarios 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. For this reason, FIDES was used. The following sections describe the three phases of the FIDES approach followed for these three scenarios.

4.1. First phase: ML-based model development and deployment

In this first phase, different teams of data scientists were involved in the development of each ML-based model. For the energy efficiency scenario and the second manufacturing scenarios (blade changing), the R programming language was used, whereas for the extrusion process set up scenario the language was Python. The selected algorithms for the predictive models of the energy demand forecasts models were the *KNN* algorithm of the *caret*[18] package, the *Random Forest Classifier* of the Sci-kit learn package[19] for the extrusion process set up and, finally, for the estimation of the optimal time for blade replacement, a custom implementation of the *Constant Profile Usage* algorithm [53].

The developed ML-based models for both scenarios have been exported in the form of R Data Serialisation (RDS) format for R and pickle for Python files, and put into production in Docker[20] containers, including the corresponding Rserver and Python packages (RServer 3.2.5 for the first scenario, Python 3.6.13 for the second and Rserver 3.6.3 for the third). This development and deployment information has been collected in the CSV file described in Section 3.2.

The deployed models are automatically executed for the energy demand forecast scenario once a day, using periodical tasks executed by a *cron daemon* process. As for the manufacturing scenarios, the task is executed every second.

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

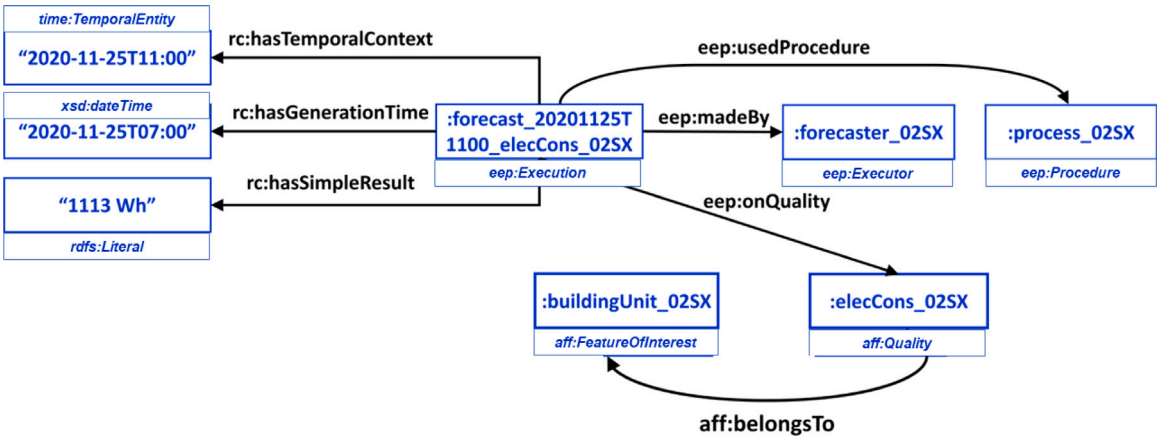

Fig. 5. Simplified graphic representation of the triples representing the 02SX building unit's electric consumption forecast.

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

?performanceMetric	?performanceValue
RMSE	242.03

4.2. Second phase: Semantic annotation and storage

Once all the ML-based models were developed and deployed in their corresponding Docker containers, their corresponding RDF triples were generated. This process was done automatically by taking as basis the CSV file generated in the previous phase by the data scientists, and the resulting RDFs were stored in the Virtuoso 8.3 repository, in different graphs, one for each scenario. Likewise, each time an estimation was generated, the corresponding CSV file was automatically filled, converted to RDF triples and stored in the same repository in the corresponding graph.

For the sake of demonstrating the semantic representation within FIDES, let us consider the following simplified use case, related to the energy efficiency scenario. 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 *KNN* algorithm implementation in R's *caret* package, with the hyperparameter k set to 7, and obtained an `RMSE` of `242.03` Wh.

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 been generated (*rc:hasGenerationTime*) and the time in which the forecast is valid (*rc:hasTemporalContext*). Additionally, the forecast has been related (via the *eep:onQuality* object property) with the electric consumption of the building unit `02SX`, represented as an individual of the *aff:Quality* class (`elecCons_02SX`). Finally, this quality belongs to (*aff:belongsTo*) the building unit `02SX`, represented as an individual of the *aff:FeatureOfInterest* class. The triples describing the electric consumption forecast made for `02SX` are represented in Fig. 5.

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, with a value *mls:hasValue* of `242.03` Wh. The predictive model and its features are characterised by the triples represented in Fig. 6. In addition, the RDF triples representing both the forecast and the predictive model's procedure can be found in Appendix A.

4.3. Third phase: Data exploitation

Once the estimations and details of the procedure used by the ML-based models to generate forecasts are semantically annotated and stored in the RDF Store, FIDES makes use of a GUI to let end users interact with this information by using REST API methods to access to the information stored in the corresponding Virtuoso graphs.

For a given graph, first of all, a list of all the systems with an associated model is presented. For instance, in the energy efficiency scenario, a list of all the participant building units is displayed, as shown in Fig. 7. This list is automatically obtained by calling a REST API method that executes a predefined SPARQL query.

To demonstrate the different data exploitation functionalities offered by FIDES, let us consider the energy efficiency scenario and one of the buttons in FIDES: *Model*. By using this button, as described in Section 3.2, end users are able to obtain information about the model and its parameter configuration.

Listings 1 and 2 show two examples of the parametric SPARQLs executed when accessing the *Model* section through their corresponding REST APIs. In the case of Listing 1, this SPARQL allows to obtain the performance obtained in the training of the model. Thus, when the *Model* button is clicked, the `$FORECAST_QUALITY` wild card is automatically replaced with the forecast quality's URI for the execution of the query. For instance (and following the example of previous sections), for the `02SX` building unit the `$FORECAST_QUALITY` wild card would be replaced with `elecCons_02SX`. The results of this query would be the ones displayed in Table 1, showing that the `RMSE` obtained when training the model is `242.03`.

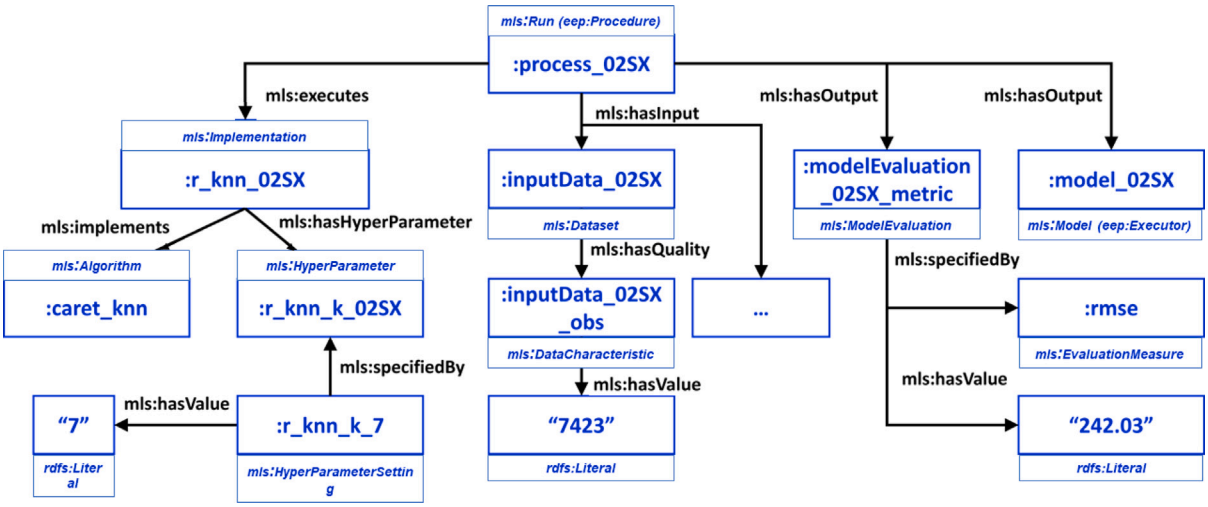

Fig. 6. Simplified graphic representation of the triples representing the example scenario's predictive model.

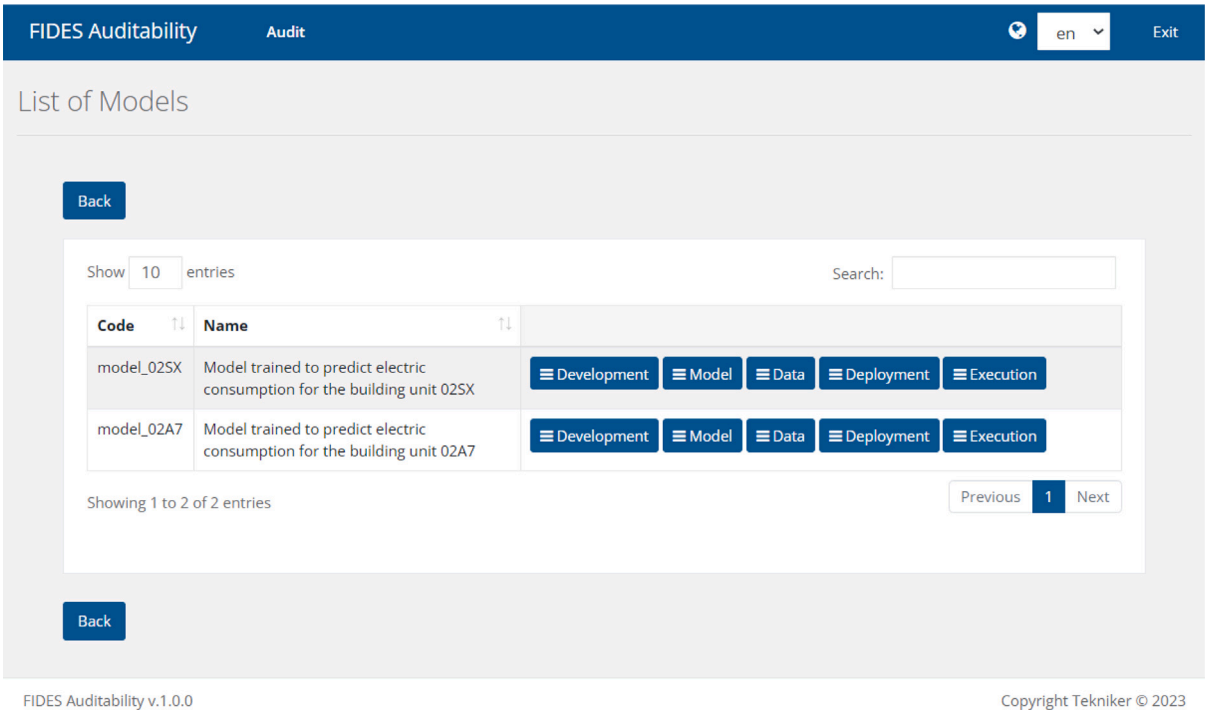

Fig. 7. FIDES GUI.

```
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#>
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.

As for Listing 2, this query allows to determine which algorithm was used for a certain forecast and its hyperparameters. Just like in Listing 1, the `$FORECAST_QUALITY` wild card would be automatically replaced with the corresponding value (which, again, for the 02SX building unit example, would be `:elecCons_02SX`). As

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

?algorithm	?hyperparam	?hyperparamValue
knn	k	7

for the `$FORECAST_QUALITY` wild card, its value corresponds to the forecast's timestamp. The results obtained from this query are shown in Table 2, which shows that, for the `forecast_2020-1125T1100_elecCons_02SX`, the model used the `KNN` algorithm with an hyperparameter `k` of 7.

```
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 ?hyperparam
       ?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
       ?hyperparam.
}
```

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

Similarly, the rest of the CQs can be answered through the different buttons of the GUI that, as it has been previously mentioned, call to the corresponding REST API methods by implementing a set of predefined parametric SPARQLs.

5. Evaluation

In order to assess the coverage of FIDES for supporting ML-based systems accountability, a user study has been carried using as basis the ML-based models for energy efficiency and manufacturing scenarios described in Section 4.

Although the semantic approach in FIDES includes 3 steps, as detailed in Section 3.2, in order to have a representative number of users evaluating the approach, the focus of this work has been put on FIDES' third step, the data exploitation phase, which is the one giving actual accountability support.

The aspects to evaluate in this user study were the system's usability (*usability*), the functions included (*functionality*) and the degree in which the system simplifies its intended task (*accessibility*) . These aspects were assessed by the users themselves through a questionnaire, as it will be detailed below.

The following lines will describe the experimentation in terms of its main characteristics, such as the type of participants or the tasks to be performed by each of them, along with the results obtained.

5.1. Experimental setup

In this user study, the number of participants recruited was 12, as it is considered an optimal number of participants to detect potential issues with the subject of study [54]. These subjects were considered potential users of the application that integrates FIDES, such as, data scientists, who were, in the end, professionals that dealt with Machine Learning systems on a daily basis. In fact, among these participants were the experts that helped define the CQs, since they may provide interesting insights on the possible limitations in the information modelled and stored in the ontology.

In this group of participants, there were 5 men, 5 women, and 2who did not disclose this information), with ages ranging from 22 to 52. Table 3 provides more information in this regard.

To perform the evaluation, each participant was presented a series of six situations caused by an unexpected ML model behaviour (e.g. errors or incorrect predictions). Examples (1) and (2) are two of those situations.

(1) The operator informs you that they are receiving a green code from the model and, according to them, the process cannot be started yet (Extrusion process scenario).
(2) The operator tells you that, after executing a series of recipes, the blade wear index is low, and they consider that, in fact, the blade needs to be changed (Blade changing scenario).

Given each situation, users were not required to solve the problem presented, but to access, through the different sections of the FIDES GUI, the relevant accountability information of the model causing the provided situation to be able to diagnose its origin, as FIDES is assumed to be able to provide all the necessary information to do so.

After finishing their designated tasks, each user was provided a questionnaire in a digital form. This questionnaire consisted of 4 question blocks. The first one was a series of demographic non-mandatory questions (i.e., age and gender). The second one included 11, 5-point-Likert scale-based questions, 10 of which correspond to the System Usability Scale (SUS) [55] (to evaluate *usability*) and 1 was generated for this experimentation to evaluate *accessibility*. The third block of the questionnaire included a yes/no question to determine whether users missed any functionalities when using the system. If users answered positively, an additional free-text question appeared so users could specify which functionalities considered necessary to implement. Finally, the fourth block was intended as a space for users to include any additional comments or remarks.

5.2. Results

As it has been pointed out previously, the results obtained from the user study will be reported according to the following three different perspectives: *usability*, *accessibility* and *functionality*.

5.2.1. Usability and accessibility

As noted earlier, the usability of FIDES has been measured with the SUS [55] questionnaire, and an additional question in the same format has been added for accessibility. The SUS questionnaire consists of 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. Furthermore, among its characteristics, it combines questions with positive (odd questions) and negative (even questions) connotations, as it can be observed in Examples (3) and (4):

Table 3
Demographic data for participants in the guide user study. (a) Gender information. (b) Age information.

Gender	
M	42%
F	42%
N/D	16%

Age	
22–34	42%
35–44	33%
45–52	16%
N/D	8%

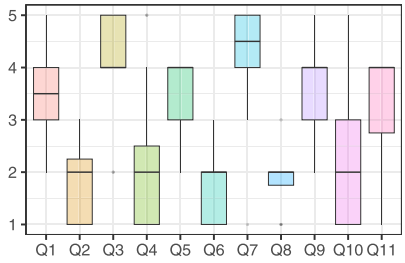

Fig. 8. Results obtained for the SUS questionnaire in the user study. Note that odd questions have a positive connotation (i.e. the higher the score the better) and even questions have a negative connotation (i.e. the lower the score the better).

(3) I think that I would like to use this system frequently. (Q1) → Positive connotation
(4) I found the system unnecessarily complex. (Q2) → Negative connotation

Regarding usability, the average score obtained in the user study has been 75[21] –that corresponds to a B grade [56]–, which indicates that in general the experience of using FIDES is good. To provide more detailed results, Fig. 8 includes the results obtained for each of the questions in the questionnaire. As for the most appreciated aspects, users consider the system easy to use (Q3, 4.17 points on average), that its usage can be learned quickly (Q7, 4.17 points on average), that it is consistent (Q6, 1.58 points on average) and that it is not cumbersome to use (Q8, 1.67 points on average).

As for the variability of the user responses, Q3, Q4 (regarding the need of a technical person to use FIDES), Q7 and Q10 (the need of learning a lot in order to use the system) have obtained the highest standard deviation (SD), with values of 1.11, 1.31, 1.19 and 1, respectively. This observation is also supported in Fig. 8, as Q3, Q4 and Q7 present outliers in their respective extreme points and Q10 presents responses in all the points in the scale. This behaviour may be justified by the duality of the questions, as these questions can be interpreted as referred to the system *per se* (that is, the usability of the application) and to the information provided by FIDES. This conclusion can be reached considering some of the comments from the users, that considered that FIDES is easy to use, but also consider that its content may be complex to understand to people who are not experts in the field or that are not tightly related to the use case.

Finally, in terms of accessibility, the provided question aimed to assess whether FIDES helps potential users to save time when performing their tasks. The average score for this question was 3.33, with a SD of 1. These results can be better interpreted by observing the box plot for Q11 in Fig. 8, in which it can be seen that the median of the results is 4 and that a 75% of the responses is also 4 (marked by the third quartile in the box plot), meaning that, although the average score may indicate that the overall impression is neutral regarding accessibility, the results in the box plot show that in fact most users are fairly satisfied with the system in this regard.

[21] Out of 100 maximum points. However, note that SUS scores are not percentages but more of percentile scores. Thus, the score in this user study remains at the 75th percentile.

5.2.2. Functionality

Besides SUS, the questionnaire provided to the participants of the user study included a question to determine if users missed any functionalities in FIDES. A 67% of the users considered that that was not the case, which hints that the overall impression in this matter is good. As for the remaining 33% of users, the main comments can be classified into three main categories: information, design and general comments.

The category that included more insights was the information one. In this category, users missed the following information:

- Values of each of the input variables of the model for each estimation.
- Means to access training data — as long as it is not confidential (e.g. a link to the training data location).
- Deployment status (i.e. whether the deployment of the model is running or not or even if it has encountered an error).
- Explanations for elements such as variables or metrics.

Further versions of FIDES will include this information, as well as the knowledge in the ontology for that matter.

As for design, users were mostly concerned about how information was arranged in the interface. One possible solution to explore is to provide users the possibility of arranging the columns that appear in FIDES GUI.

Finally, the general comments considered the model information in FIDES (*Model* tab) comprehensive, and FIDES as an easy-to-use solution that provides a fast access to relevant information for the accountability of ML models. As it can be seen, this comments are also in line with the positive results obtained in the SUS questionnaire.

6. Conclusions

Nowadays, 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 from users. In this context, there are some scenarios where certain legal, ethical and technological compliance requirements must be satisfied, and the potential causes that may lead to undesirable outcomes must also be identified.

To address these needs, to hold ML systems accountable and, in the end, to contribute to overcome adoption barriers related to AI systems, this article has presented FIDES. FIDES is an ontology-based framework for representing, structuring and setting formal relations among the models and the forecasts that conform a traditional, statistical ML-based 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 practices, FIDES reuses existing quality ontologies.

The validity of FIDES has been demonstrated in two real-world scenarios: an energy efficiency scenario and two manufacturing scenarios. From the results obtained on the user study carried out for its evaluation, it can be concluded that the overall usability of the system is good, that the current functionalities may satisfy most potential users' requirements, and that the access to the information needed for holding systems accountable is much more straightforward compared with a traditional approach. This user study has also helped identifying the limitations of FIDES, such as the information displayed or its design, which will be carefully analysed for their correction in future versions.

The potential of Semantic Technologies to fill existing gaps and address unsolved challenges towards trustworthy AI is high, even though it is an area that is not fully exploited yet. The contributions presented in this article aim 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 about the possibilities of Semantic Technologies in the different factors that may contribute to achieving trustworthy AI systems. Therefore, apart from accountability, the research in 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.

CRediT authorship contribution statement

Izaskun Fernandez: Conceptualization, Ontology development, Validation. **Cristina Aceta:** Validation, Evaluation. **Eduardo Gilabert:** Software, Validation. **Iker Esnaola-Gonzalez:** Conceptualization, Ontology development.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme by the project AI-PROFICIENT under grant agreement no. 957391, and also from the Basque Government, Spain (ELKARTEK 2022) by the project SIIRSE under grant agreement No KK-2022/00007. 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.

Appendix A. RDF examples

This appendix shows the RDF representation of the energy scenario 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 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 time: <http://www.w3.org/2006/time#> .
@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"^^time
        :TemporalEntity
        rc:hasSimpleResult "1113 Wh"^^rdfs:Literal.
```

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

```
@prefix : <http://example.com/> .
@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: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;

: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_k_7 rdf:type mls:HyperParameterSetting;
        mls:specifiedBy :r_knn_k_02SX.
        mls:hasValue "7"^^rdfs:Literal .

: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"^^rdfs:Literal .

: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"^^rdfs:Literal .

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

Listing 4: RDF representation of the energy scenario's predictive model process.

Appendix B. Competency questions list

This appendix shows the complete list of competency questions used for developing the ontology, according to their category. The *Development*, *Model* and *Data* categories correspond to the *model development procedures* topic, the *Data* category corresponds to the *deployment* topic and, finally, the *Execution* category corresponds to the *estimations* topic. These topics are defined in Section 3.1.1.

Development

- Which is the development framework?

- Which is the development framework's version?
- Which is the Operating System (OS) of the development environment?
- Who developed the model and which is their contact email?
- When was the model developed?
- Where is the source code stored? (Repository)
- Where is the Docker container that could reproduce the model stored and its associated commit? (if any)
- Where is the script that generated the model stored and its associated commit?
- Where is the model stored and its associated commit?

Model

- Which is the base algorithm of the ML-base model?
- Which is the package that implements the algorithm?
- Which is the version of the package?
- Which are the hyperparameter values of the ML-based model?
- Which are the optimal values for each parameter of the algorithm?
- Which is the validation method?
- Which is the validation metric used and its value?
- Which is the objective of the model?
- In case of classification problems, which are the values of the type class?

Data

- Where is the data stored?
- (If the data is tracked with version control) Which is the framework used and the commit for version controlling of a specific set of data?
- Which is the number of observations used for the training of a given model?
- When was the first data point collected within a given model's training data?
- When was the last data point collected within a given model's training data?
- Which are the names of the TD's variables?
- Which are the names of the TD's predictor variables?
- Which is the name of the TD's response variable?
- Which is the frequency and quality of a given model's TD?
- How is TD pre-processed?

Deployment

- Which is the OS of the deployment server?
- How is the model triggered, on event or under request?
- Where are the estimations (derived from the ML-based model executions) stored?
- Is the error between the real and predicted values being stored?

Execution

- When was the forecast made?
- For what point in time is the forecast valid?
- What is the forecast value?
- What is the forecast's error metric?
- What is the forecast's error value?

References

[1] M. Chui, S. Malhotra, AI adoption advances, but foundational barriers remain, mckinsey, 2018, See: https://www.mckinsey.com/featured-insights/artificial-intelligence/ai-adoption-advances-but-foundational-barriers-remain.

[2] S.S. ÓhÉigeartaigh, J. Whittlestone, Y. Liu, Y. Zeng, Z. Liu, Overcoming barriers to cross-cultural cooperation in AI ethics and governance, Philos. Technol. 33 (4) (2020) 571–593, http://dx.doi.org/10.1007/s13347-020-00402-x.

[3] M. Cubric, Drivers, barriers and social considerations for AI adoption in business and management: A tertiary study, Technol. Soc. 62 (2020) 101257, http://dx.doi.org/10.1016/j.techsoc.2020.101257.

[4] G. Baryannis, S. Validi, S. Dani, G. Antoniou, Supply chain risk management and artificial intelligence: State of the art and future research directions, Int. J. Prod. Res. 57 (7) (2019) 2179–2202, http://dx.doi.org/10.1080/00207543.2018.1530476.

[5] E. Broadbent, R. Stafford, B. MacDonald, Acceptance of healthcare robots for the older population: Review and future directions, Int. J. Soc. Robot. 1 (4) (2009) 319, http://dx.doi.org/10.1007/s12369-009-0030-6.

[6] 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, J. Am. Med. Inform. Assoc. 25 (9) (2018) 1248–1258, http://dx.doi.org/10.1093/jamia/ocy072.

[7] J.T.M. Ingibergsson, U.P. Schultz, 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, http://dx.doi.org/10.1007/978-3-319-26844-6_25.

[8] M. Raghavan, S. Barocas, J. Kleinberg, 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, http://dx.doi.org/10.1145/3351095.3372828.

[9] J. Sánchez-Monedero, L. Dencik, 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, in: FAT* '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 458–468, http://dx.doi.org/10.1145/3351095.3372849.

[10] S. Chuang, C.M. Graham, Embracing the sobering reality of technological influences on jobs, employment and human resource development, Eur. J. Training Dev. (2018) http://dx.doi.org/10.1108/EJTD-03-2018-0030.

[11] M. Madsen, S. Gregor, Measuring human-computer trust, in: 11th Australasian Conference on Information Systems, Vol. 53, ACIS, 2000, pp. 6–8.

[12] J.Y. Jian, A.M. Bisantz, C.G. Drury, Foundations for an empirically determined scale of trust in automated systems, Int. J. Cogn. Ergon. 4 (1) (2000) 53–71, http://dx.doi.org/10.1207/S15327566IJCE0401_04.

[13] B. Cahour, J.F. Forzy, Does projection into use improve trust and exploration? An example with a cruise control system, Saf. Sci. 47 (9) (2009) 1260–1270, http://dx.doi.org/10.1016/j.ssci.2009.03.015.

[14] D. Gunning, Explainable artificial intelligence (Xai), Def. Adv. Res. Projects Agency (DARPA), nd Web 2 (2) (2017).

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

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

[17] J. Fox, The uncertain relationship between transparency and accountability, Dev. Pract. 17 (4–5) (2007) 663–671, http://dx.doi.org/10.1080/09614520701469955.

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

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

[20] B. Nushi, E. Kamar, E. Horvitz, Towards accountable AI: Hybrid human-machine analyses for characterizing system failure, in: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, vol. 6, 2018, pp. 126–135, URL https://ojs.aaai.org/index.php/HCOMP/article/view/13337.

[21] D. Oberle, How ontologies benefit enterprise applications, Semantic Web 5 (6) (2014) 473–491, http://dx.doi.org/10.3233/SW-130114.

[22] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.

[23] R. Confalonieri, T. Weyde, T.R. Besold, F.M. del Prado Martın, TREPAN reloaded: A knowledge-driven approach to explaining black-box models, Front. Artif. Intell. Appl., 325 (2020) 2457–2464, http://dx.doi.org/10.3233/FAIA200378.

[24] S. Chari, O. Seneviratne, D.M. Gruen, M.A. Foreman, A.K. Das, 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, http://dx.doi.org/10.1007/978-3-030-62466-8_15.

[25] C. Panigutti, A. Perotti, 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, http://dx.doi.org/10.1145/3351095.3372855.

[26] F. Lécué, J. Chen, J.Z. Pan, H. Chen, Knowledge-based explanations for transfer learning, in: Studies on the Semantic Web, Volume 47: Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, IOS Press, 2020, pp. 180–195, http://dx.doi.org/10.3233/SSW200018.

[27] HLEG-AI, High Level Expert Group on Artificial Intelligence, Ethics guidelines for trustworthy AI, 2019, URL https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai. last visited on 2021-02-08.

[28] 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, Inf. Fusion 58 (2020) 82–115, http://dx.doi.org/10.1016/j.inffus.2019.12.012.

[29] A. Seeliger, M. Pfaff, 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, in: PROFILES-SEMEX 2019, vol. 2465, CEUR-WS, 2019, pp. 30–45, URL http://ceur-ws.org/Vol-2465/semex_paper1.pdf.

[30] S. Chari, D.M. Gruen, O. Seneviratne, D.L. McGuinness, Foundations of explainable knowledge-enabled systems, in: Studies on the Semantic Web, Volume 47: Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, IOS Press, 2020, pp. 23–48, http://dx.doi.org/10.3233/SSW200010.

[31] P.I. Nakagawa, L.F. Pires, J.L.R. Moreira, L.O. Bonino da Silva Santos, F. Bukhsh, Semantic description of explainable machine learning workflows for improving trust, Appl. Sci. 11 (22) (2021) 10804.

[32] I. Esnaola-Gonzalez, Semantic technologies towards accountable artificial intelligence: A poultry chain management use case, in: Artificial Intelligence XXXVII, Springer International Publishing, Cham, 2020, pp. 215–226, http://dx.doi.org/10.1007/978-3-030-63799-6_17.

[33] L. Waltersdorfer, Auditable semantic web machine learning systems, 2021.

[34] I. Naja, M. Markovic, P. Edwards, C. Cottrill, A semantic framework to support AI system accountability and audit, in: The Semantic Web: 18th International Conference, in: ESWC 2021, Springer International Publishing, 2021, pp. 160–176.

[35] F.J. Ekaputra, A. Ekelhart, R. Mayer, T. Miksa, T. Sarčević, S. Tsepelakis, L. Waltersdorfer, Semantic-enabled architecture for auditable privacy-preserving data analysis, Semantic Web Pre-press (Pre-press) (2021) 1–34, http://dx.doi.org/10.3233/SW-212883.

[36] E. Simperl, Reusing ontologies on the semantic web: A feasibility study, Data Knowl. Eng. 68 (10) (2009) 905–925, http://dx.doi.org/10.1016/j.datak.2009.02.002.

[37] M. Fernández-López, M.C. Suárez-Figueroa, A. Gómez-Pérez, Ontology development by reuse, in: M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, A. Gangemi (Eds.), Ontology Engineering in a Networked World, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 147–170, http://dx.doi.org/10.1007/978-3-642-24794-1_7.

[38] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez, A. Arnaiz, Ontologies for observations and actuations in buildings: A survey, Semantic Web 11 (4) (2020) 593–621, http://dx.doi.org/10.3233/SW-200378.

[39] A. Haller, K. Janowicz, S. Cox, M. Lefrançois, K. Taylor, D.L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, 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, http://dx.doi.org/10.3233/SW-180320.

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

[41] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez, A. Arnaiz, Two ODPs towards the notion of influenceable features of interest, in: Advances in Pattern-Based Ontology Engineering, in: Studies on the Semantic Web, vol. 51, IOS Press, 2021, pp. 89–106, http://dx.doi.org/10.3233/SSW210008.

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

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

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

[45] P. Panov, L. Soldatova, S. Džeroski, Ontology of core data mining entities, Data Min. Knowl. Discov. 28 (5–6) (2014) 1222–1265, http://dx.doi.org/10.1007/s10618-014-0363-0.

[46] C.M. Keet, A. Ławrynowicz, C. d'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, M. Hilario, The data mining optimization ontology, J. Web Semant. 32 (2015) 43–53, http://dx.doi.org/10.1016/j.websem.2015.01.001.

[47] G.C. Publio, D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, H. Zafar, ML-Schema: Exposing the semantics of machine learning with schemas and ontologies, 2018, arXiv:1807.05351.

[48] D. Garijo, M. Poveda-Villalón, Best practices for implementing FAIR vocabularies and ontologies on the web, in: G. Cota, M. Daquino, G.L. Pozzato (Eds.), Applications and Practices in Ontology Design, Extraction, and Reasoning, in: Studies on the Semantic Web, vol. 49, IOS Press, 2020, pp. 39–54, http://dx.doi.org/10.3233/SSW200034.

[49] M. Poveda-Villalón, A. Gómez-Pérez, M.C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, Int. J. Semant. Web Inf. Syst. (IJSWIS) 10 (2) (2014) 7–34.

[50] D. Garijo, O. Corcho, M. Poveda-Villalón, FOOPS!: An ontology pitfall scanner for the FAIR principles, in: International Semantic Web Conference (ISWC) 2021: Posters, Demos, and Industry Tracks, in: CEUR Workshop Proceedings, vol. 2980 (2021) URL http://ceur-ws.org/Vol-2980/paper321.pdf.

[51] D. Garijo, WIDOCO: A wizard for documenting ontologies, in: International Semantic Web Conference, Springer, Cham, 2017, pp. 94–102.

[52] M. Kuhn, J. Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, R.C. Team, et al., Package 'caret', R J. 223 (7) (2020).

[53] K. López de Calle-Etxabe, E. Garate-Pérez, A. Arnaiz, Towards a circular rotating blade wear assessment digital twin for manufacturing lines, IFAC-PapersOnLine 55 (2) (2022) 561–566, http://dx.doi.org/10.1016/j.ifacol.2022.04.253, 14th IFAC Workshop on Intelligent Manufacturing Systems IMS 2022. URL https://www.sciencedirect.com/science/article/pii/S2405896322002543.

[54] J. Nielsen, T.K. Landauer, A mathematical model of the finding of usability problems, in: Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, CHI '93, Association for Computing Machinery, New York, NY, USA, 1993, pp. 206–213, http://dx.doi.org/10.1145/169059.169166.

[55] J. Brooke, et al., SUS-A quick and dirty usability scale, in: Usability Evaluation in Industry, vol. 189, (no. 194) CRC Press, 1996, pp. 4–7.

[56] J. Sauro, 5 ways to interpret a SUS score, 2018, https://measuringu.com/interpret-sus-score/. (Accessed on 03/06/2023).