SML: Semantic Machine Learning Model Ontology

Lara Kallab, Elio Mansour, Richard Chbeir

SML: Semantic Machine Learning Model Ontology

Lara Kallab[1], Elio Mansour[2], and Richard Chbeir[3]

[1] Open Group, Levallois Perret, 92300, France
lara.kallab@open-groupe.com
[2] Scient Analytics, Paris, France
elio.mansour@scient.io
[3] Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000, Anglet, France
richard.chbeir@univ-pau.fr

Abstract. Artificial Intelligence is a set of technologies that simulate human-like cognition, using computer software and systems, to perform tasks associated with intelligent beings. One method of doing so, is Machine Learning (ML), which enhances system efficiency based on learning algorithms that create models from data and its underlying patterns. Nowadays, many ML models are being generated with different characteristics (e.g., type of the algorithm used, data set used to train it, resulting model performance), thus making the selection of a suitable model for a given use case a complex task, especially for non-expert users (with no or limited knowledge in ML). In this paper, we propose SML, an ontology-based model for Semantic Machine Learning description. SML allows, mainly, to describe and store ML models' characteristics with their operational specifications, related data features, contextual usage, and evaluation metrics/scores to facilitate and improve ML model selection. The conducted experiments show promising results on both the efficiency and the performance levels.

Keywords: Machine Learning Model · Supervised Learning · Ontology · Data Set · Context Description · Model Evaluation

1 Introduction

Today, Artificial Intelligence (AI) has become a major player in a wide range of fields (e.g., social, commercial, industrial), such as speech recognition, medical diagnosis, autonomous vehicles and building automation [5]. It is basically a computer system designed to mimic human intelligence by accessing data from numerous sources and systems, allowing to make decisions and learn from the outcomes. Machine Learning (ML) is an implementation of the AI that enables computers to learn from data without specific programming [12]. It is focused on building models that can learn from historical data, to identify meaningful data relationships and patterns [6], and make logical decisions with little or no human intervention. ML automates the construction of analytical models using data that encompasses various forms of numerical information including numbers, words, images, etc.

In ML, there are a plethora of models that a user can adopt and reuse (without the need to create new models every time), all having their own specifications and uses. Each model has different characteristics, such as the type of algorithm the model relies on (e.g., Linear Regression and Bayes Classifier [8]), the data set used to train it, the application domain (e.g., finance, travel and transportation), its performance, etc. All this makes the task of selecting a suitable model, for a given use case, a complex one especially for non-expert users (having limited or no ML knowledge). Choosing the right model for a specific use case is essential. In fact, the better the machine learning model fits a given case, the more accurately it can find features and patterns in the data. This means better decision-making, with more accurate analysis and forecasts. For example, using a regression model trained on a winter season data, to predict data related to a summer season, will more likely cause poor quality in the predicted results, since the learning of the model is applied on a different data set pattern (in terms of season). Therefore, it is necessary to describe ML models and represent their main characteristics, in a semantic form, to know how and where each model ca be better used/adopted, and understand their operating context. This allows to compare, evaluate and propose the most appropriate model(s) for a specific application scenario.

In the literature, there are several models, approaches, and reviews that describe machine learning models' characteristics, applicability, and performance. However, these works have several limitations. For instance, most of them, [1,2,7,10,14], do not describe well the data sets used to train and test the models. In addition, the majority of the works, [7,10,11,14], does not take into account model application domain and model operational performance. Also, none of them describes well the model usability or its context (e.g., temporal context, spatial context), and neglects considering model metadata on several levels (e.g., ML model metadata, algorithm metadata, data set metadata). The aforementioned criteria are important to consider in order to facilitate the usability and the selection of the ML models. Given the limitations of the existing solutions in representing machine learning models, which is essential for understanding their functioning, their applications, evaluating them, and comparing them, we propose, in this paper, **SML**: an ontology-based model for **S**emantic **M**achine **L**earning description. SML describes machine learning models through a human and a machine understandable vocabulary, to ease the comprehension, the evaluation and the selection of the convenient ML model to be used in a given context. As it is based on an ontology model [9], SML gives the same meaning to the specified and exchanged ML models characteristics and operating specifications. This makes it easier for systems, from various organizations and platforms, to store, integrate, and share ML knowledge, enabling both syntactic and semantic interoperability, and allowing for future extensibility and adaptation.

The rest of the paper is organized as follows. Section 2 presents a scenario to motivate the usability and applicability of our work. Section 3 reviews the related work and highlights the added value of our model. Section 4 details the specifications of our proposed semantic machine learning model ontology. Section 5 evaluates the efficiency and the performance of the solution. Finally, Section 6 summarizes the work and discusses future research directions.

2 Motivating Scenario

In order to show the motivation behind our proposal, consider the following Smart City scenario illustrated in Figure 1. The environment is densely covered by an extensive Wireless Sensor Network (WSN) that collects a wide variety of data from the city (e.g., CO_2 emissions, lighting conditions, noise levels, energy consumption, temperature). The city has appointed a team of experts to help monitor, analyze, and forecast elements from within the city, in an effort to make it a smart, proactive, safe, and healthy environment for its occupants. The team members have different expertise and are interested in forecasting and analyzing data related to their respective fields. Figure 1 illustrates some examples: (i) environmental experts are interested in predicting noise, air, and water pollution levels, to make the city a more healthy space; (ii) road safety experts are interested in predicting traffic congestion, risky traffic hours/conditions, and road deterioration to avoid deadly accidents; (iii) weather experts are interested in predicting rising temperatures and extreme weather conditions, in order to disseminate important information to the city occupants, in a proactive manner; and (iv) energy experts are interested in analyzing and predicting energy consumption, and production levels/patterns in the city, to help make it a greener more eco-friendly place.

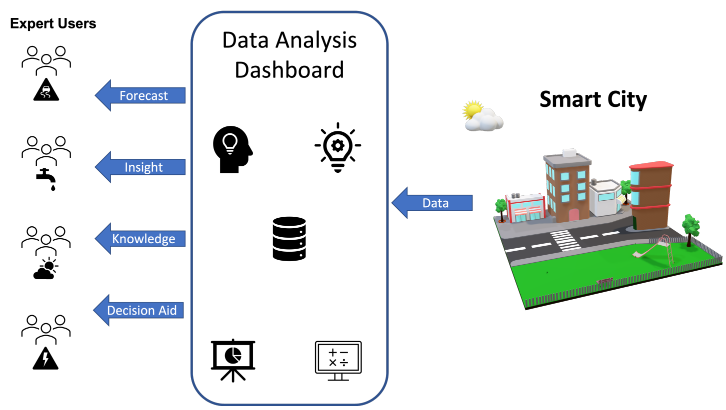

Fig. 1: Smart City Scenario

The aforementioned team members often need to collaborate with each other on cross-field projects. More importantly, they all need to generate, train, test, and deploy prediction models that ingest the collected data on one end, and provide the required forecasts on the other. In such a dynamic and collaborative environment, it is easy to quickly end up with a huge number of machine learning models each: (i) covering a specific need; (ii) relying on a specific algorithm; (iii) using a specific data set for training; (iv) having a specific level accuracy; and (v) deployed in a specific application domain/field of interest.

In order to sustain this collaborative workspace, prevent isolated analysis, and provide a decision-making process based on collective intelligence and shared insights, the team members require a system capable of storing, and recommending ML models for each new application use case. This will enable model re-usage instead of creating slightly different models every time one needs to make a prediction, and will ensure reproducibility of experiments in the context of open science. The system would suggest and retrieve a model that best fits the user's need if such model already exists. Otherwise, the user can always generate a new model from scratch. This will significantly prove useful when considering the growing number of ML models that will be generated over time. Moreover, such a recommendation system would increase users' understanding of the existing models, improve result explain-ability, and allow the team to collaborate in a more productive manner. In order to deliver this ML model recommendation engine, one needs to address two main sets of challenges related to: **(i) Model Representation**: this entails the challenges related to the description of the models, as well as their metadata, technical aspects (i.e., algorithmic specifications), used data sets (i.e., training, testing features/data specifications), the application domains in which the models are eventually deployed, and the evaluation metrics/scores; and **(ii) Model Recommendation:** this entails a whole different set of challenges related to model similarity measures, model recommendation processes, and model recommendation optimization. In this work, we solely focus on the first set of challenges related to the model representation. We will consider model recommendation in a future dedicated work. As a result, we specifically focus here on the following challenges:

Challenge 1. How to extensively represent the machine learning models and their various descriptive metadata? This helps users to easily search, version, and retrieve the existing ML models (e.g., describing when, where, and by whom a model was generated and for which purpose/application domain).

Challenge 2. How to cover technical aspects and map models to the algorithms that generated them? How to categorize technical specifications to allow easy search and retrieval of ML models? This helps clustering and filtering models based on their underlying technical features (e.g., separating classification from regression, linear from nonlinear, statistical from deep learning).

Challenge 3. How to cover the different intricacies of the training and testing data sets in the modeling? How to capture the context (i.e., spatial, temporal, or feature-based) of the data sets that help build and evaluate the ML models in the representation? This allows comparing model similarity from a data perspective, as well as discovering the usage context of the models based on their training/testing data set features (e.g., differentiating an indoor from an outdoor temperature prediction model since their contexts differ).

Challenge 4. How to include the application domains where the models are eventually deployed in the description? This allows a higher level clustering/-categorization of the ML models based on their field of application, which will consequently improve model suggestion to users based on their expertise (e.g., to provide team members from various fields with the useful models that are applicable in their application domain).

Challenge 5. How to include model evaluation metrics and scores in the representation? The evaluation is crucial for ranking and presenting adequate models for user needs (e.g., to provide energy experts with the most accurate model for energy prediction in a smart building).

Existing works focus mainly on data set similarity or some performance metrics when trying to suggest an adequate model for a specific task. In this work, we aim to extend existing solutions by considering a more complete set of concepts (e.g., application domains, usage scenarios/contexts, technical algorithmic aspects) that could impact a ML model recommendation. However, before detailing our proposal, we review, next, related works about Machine Learning model representation and evaluate the state of the art based on the challenges/requirements of our motivating scenario.

3 State of the Art

In this section, we study several ML description models, approaches, and reviews, that are defined to mainly, give knowledge about ML techniques and algorithms (e.g., categories, advantages, to mention a few), and to describe their performance and applicability. We compare these solutions according to the following different criteria, grouped into two categories:

1. ***ML representation criteria*:** which include the criteria used to represent ML models, their generation/building process, their behavior, their performance, and useful metadata descriptors:
   - **Criterion 1.A. Algorithm representation:** denoting the ability to describe and link the models to the corresponding ML algorithms that generated them, thus allowing the inference of their usability, and technical limitations.
   - **Criterion 1.B. Data representation:** denoting the representation of the data sets used to train and test the ML models including their characteristics (e.g., the features, their values, and statistical descriptors).
   - **Criterion 1.C. Performance representation:** denoting the ability to include accuracy and performance metrics/descriptions for each ML model, to give insights on the quality of the obtained results and allow for ML models comparisons.
   - **Criterion 1.D. Metadata representation:** denoting the ability to include meta descriptors that enrich the ML modeling and include various high-level features/information (e.g., data set metadata, algorithm metadata, model metadata).
2. ***ML usability and compatibility criteria*:** which hold the criteria used to describe the application domain and the context of each ML model:
   - **Criterion 2.A. Application-domain representation:** denoting the ability to cover a keyword-based representation of various application domains and link ML models to these domains (e.g., linking a temperature prediction model to environmental monitoring application domain).

- **Criterion 2.B. Usability representation:** denoting the ability to specify several ML models contexts (i.e., defining the environment constraints) in each application domain, to know where each ML model can be more convenient to be applied for more accurate results (e.g., when using a regression prediction model trained on a winter season data in a summer season, the quality of the results will be negatively affected).

3.1 Ontology-based ML Description models

MLOnto [2], Machine Learning Ontology, is an ontology that represents knowledge about the Machine Learning discipline. It consists of 7 main classes: Algorithms, Applications, Dependencies, Dictionary, Frameworks, Involved, and MLTypes. Despite representing different ML types (i.e., AutoML, Reinforcement Learning, Semi-supervised Machine Learning, Supervised Learning, and Unsupervised Learning), the representation of the model is very high level and limited, as it neglects several criteria, e.g., Data sets representation (training sets and testing sets), model performance, and usability.

In [4] an ontology-based approach is proposed for making Machine Learning systems accountable. The approach is based on three phases: (1) the creation of the predictive models and their deployment for availability, (2) the annotation of pertinent information derived from the predictive models and forecasts by using ontological-based terms, and (3) the storage of the annotated data while providing means to exploit them for accountability. The second phase is based on two areas. In the first one, the forecasts produced by the predictive model are represented, by using three ontologies models: the AffectedBy ontology (`https://iesnaola.github.io/AffectedBy`), the Execution-Executor-Procedure (EEP) (`https://iesnaola.github.io/EEP`), and the Result Context (RC) (`https://iesnaola.github.io/RC`). In the second one, the predictive procedures used for achieving the forecasts are modeled via the ML-Schema ontology [13]. Despite that these ontologies' models cover many aspects of Machine Learning models, including model performance and training data sets representation, they lack in considering several criteria, such as the model context (other than the temporal and the spatial ones) with its constraints (whenever it is necessary) and the model application domain.

Authors in [1] define an ontology-based IML (Interpretable Machine Learning), OnML, for generating semantic explanations, by using interpretable models, ontologies, and information extraction techniques, in order to generate semantic explanations. This is done by identifying and including ontology-based tuples into a sampling strategy, where the semantic relationships between terms, words, and ideas, are sampled and captured in training the interpretable model, rather of using each of them separately. To reduce the search space for semantic explanations, an anchor learning method is also proposed. The work mainly focuses on using ontologies models for semantic explanation of the predicted ML results, without representing or describing the ML data sets, their behavior, context, etc. However, by relying on the ontologies' models, the approach gives some hints regarding the application domain of the used ML, as well as their usability.

3.2 Context-aware ML Description Approaches

The work in [11] describes an approach that uses contextual information to train ML models. It mainly consists of training ML models to maximize a specific scoring function for each operating context. In the experiments, the context-aware approach results, obtained from specialized models that were trained for each particular context, were compared against the use of a general model that was trained using all contexts. The results demonstrate that the suggested approach lessens bias toward a strategy that employs a special general model, however, the error difference is considered to be low. Therefore, an evaluation is needed to identify which strategy fits more application needs. Nonetheless, the context-aware approach should be taken into consideration, depending on how crucial the application resources needs are (such as connectivity and memory). Comparing the proposed approach to our work, the contexts of the ML models are manually defined and used, without being represented (nor other aspects, e.g., data sets, application domains, etc.) via a machine understandable form, which allows for the correct and the automatic usage of ML models in the right contexts.

3.3 Reviews and Surveys-based ML Description

In [14], a review is given to provide definitions and a foundational understanding of the ML categories (i.e., Supervised, Unsupervised, and Reinforcement Learning). It discusses methods for the design of supervised ML studies, and introduces the bias-variance trade-off, as a key theoretical underpinning for supervised Machine Learning. The work provides an overview and a description of common supervised ML algorithms (Linear Regression, Logistic Regression, Naive Bayes, etc.), however, it does not represent them (i.e., data sets, applications domains, etc.) through a comprehensive model to machine, allowing for their correct usage in the required cases.

Table 1: Evaluation of existing ML description models and approaches w.r.t. the identified criteria

	1. ML Representation Criteria				2. ML Usability & Compatibility	
	Algorithm	**Data**	**Performance**	**Metadata**	**Application domain**	**Usability**
MLOnto, 2020 [2]	+	-	-	**Limited**	+	-
ML-Schema, 2021 [4]	+	+	+	**Limited**	-	**Limited**
OnML Approach, 2022 [1]	-	-	-	-	**Limited**	**Limited**
Context-aware ML-based Approach, 2018 [11]	-	-	-	-	-	**Limited**
Review, 2019 [14]	**Limited***	-	**Limited***	-	**Limited***	**Limited**
Survey, 2015 [10]	**Limited***	-	**Limited***	-	**Limited***	**Limited**
Study, 2020 [7]	**Limited***	-	**Limited***	-	**Limited***	**Limited**

In [10], a survey is given to discuss the strength and the weakness of different ML algorithms: Logic basic algorithms (e.g., Decision Trees and Learning Set of Rules), Statistical learning algorithms (e.g., Bayesian Networks), Instance-based Learning, Support Vector Machines, and Deep Learning. Despite describing the scope of the usage of each ML algorithm, such a description is dedicated to users that should have some expertise to know how and where these MLs' are better used. Our work goes further beyond by describing ML models, in terms of the data sets used, the corresponding application domain, etc., through an understandable machine form that facilitates ML use, based on different contexts.

The study in [7] gives an overview of ML classifications (i.e., Unsupervised, Semi-supervised, and Reinforcement), and presents three different ways (i.e., Clustering, Classification, and Prediction) that ML is used in enterprises. The work also includes a process model for choosing ML algorithms based on the type of data, intended interpretability, and desired accuracy. Although the study helps in comprehending the state of ML techniques, and their applicability in enterprise applications, along with the trade-off between their interpretability and their accuracy, it misses several aspects that are important to consider, to know what ML model is better to use in particular cases, e.g., describing ML data sets, their corresponding contexts, etc. This is apart from neglecting ML models description using a comprehensible machine format, reducing, thus, users expertise and knowledge.

In Table 1, we show the comparative study of the ML description models, approaches and reviews previously described, based on the criteria outlined at the beginning of this section. We utilized the "+" symbol to indicate a criterion's positive coverage, the "-" symbol to indicate a criterion's lack of coverage, "Limited" to denote partial coverage of a criterion, and "Limited*" to indicate partial coverage of a criterion with a lack of implementation/proposed model.

4 Semantic Machine Learning Model Ontology

To describe and store the characteristics and the operational specifications of ML models, which are necessary to facilitate ML models comprehension behavior, and ease their selection in the right contexts, we present, in this section, an ontology-based model entitled SML, for Semantic Machine Learning description. SML, which is represented in Figure 2 via entities an relationships between these entities, is based on a vocabulary that can be used by different environments and/or platforms to describe ML models in a normalized manner. We note that attributes of each entity are not shown in Figure 2 for the sake of clarity.

4.1 SML Ontology Features

SML Model' Representation and Application. As shown in Figure 3, a semantic Machine Learning Model, is a model that is designed to recognize patterns or behaviors in some collected data, based on previous or historical data, referred to as training data set (*SML:TrainingDataSet*). A training data set is a data set (*SML:DataSet*) that is used during the learning process to fit (train) a model for prediction or classification of values that are known in the training set, but unknown in other (future) data. Each SML model has some metadata (*SML:MetaData*), e.g., Creation Date, Model Developer, etc., and is applicable in specific application domains (*SML:ApplicationDomain*), such as smart buildings, healthcare, transportation, etc.

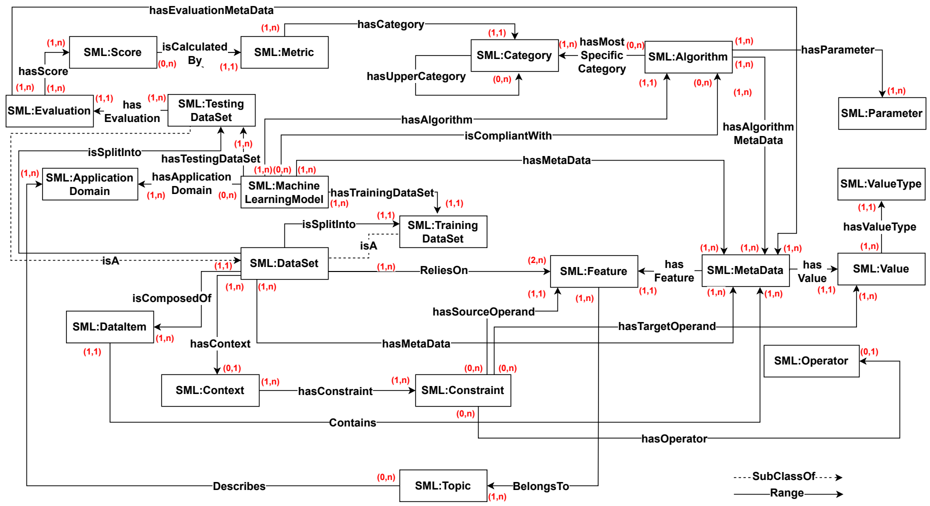

Fig. 2: Overview of the proposed Semantic Machine Learning (SML) model ontology

SML models are based on specific algorithms (*SML:Algorithm*), e.g., Naïve Bayes, Support Vector Machines, Decision Trees, etc., having, each, some metadata (e.g., Creation Date, Description), and some parameters (i.e., key-value pairs). An algorithm belongs to a specific category (*SML:Category*), such as Classification and Regression, which might have, if necessary, subcategories through the relation "hasUpperCategory". Some SML models that are based on particular algorithms (e.g., Linear Regression) can be compliant, through the relation "isCompliantWith", with other algorithms (e.g., Lasso Regression). This can be known by applying several calculations and studies on the training data set used for each SML model, along with their contexts (see subsection below).

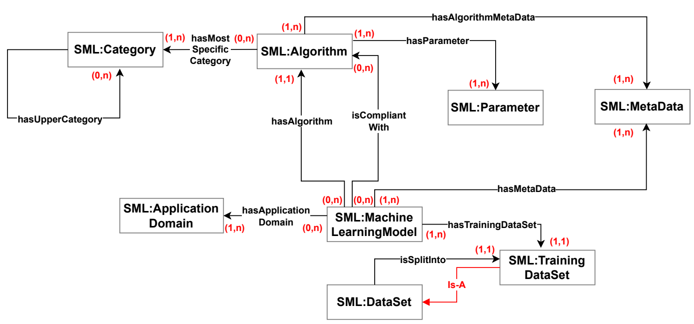

Fig. 3: SML model' representation and application

SML Data Set Modeling and Context. As represented in Figure 4, a data set, *SML:DataSet*, can be splitted into either: (1) a training data set

(*SML:TrainingDataSet)* that is used to train the SML model, or (2) a testing data set (*SML:TestingDataSet*), which is used to test and evaluate the SML model after being trained (see Figure 5). It is composed of data items (*SML:DataItem*). A data item has some metadata (*SML:MetaData*). Each metadata has a feature, *SML:Feature* (e.g., Creation Date, Description, Temperature, Location), and a value, *SML:Value*, linked to a value type, *SML:ValueType*. We defined concepts for each of the features, values and type of values, to be able to apply on some features values, specific constraints (see below), which can be in many cases necessary to describe the context of ML models.

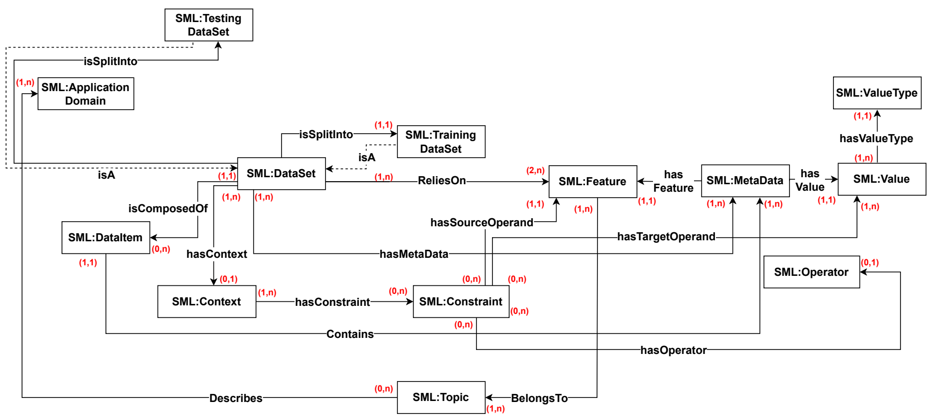

Fig. 4: SML data set' modeling and context

A data set, which has also some metadata, relies on, at least, two features, having, each, some attributes, such as a Name, a Type (e.g., categorical, textual, numerical), a Range, and a Boolean value (to check whether it is an independent feature or not). The independent feature is the cause. Its value is independent of other variables in the study. The dependent feature is the effect. Its value depends on changes in the independent feature. Features belong to topics (*SML:Topic*) that are used to describe the application domain (*SML:ApplicationDomain*) of a SML model. A data set has a context (*SML:Context*) represented by constraints (*SML:Constraint*), having, each, a source operand, i.e., *SML:Feature*, a target operand, i.e., *SML:Value*, and an operator (*SML:Operator*). For example, we can have a spatio-temporal context defined by the "Season" feature, with the value "Winter", and by a "Location" feature having "Paris" as a value. Such context allows to know that, for instance, the training data set of a specific ML model, is related to Paris in Winter. Contexts also enable to use some data sets for other ML models, depending on the matching or how closely their contexts are.

SML Model Evaluation. Once the machine learning model is built using the training data set, it needs to be tested using data, referred to as testing data set (*SML:TestingDataSet*). A testing data set is used to evaluate the performance and progress of the SML model training, which might need to be adjusted or optimized for improved results. As shown in Figure 5, a testing data set has an evaluation (*SML:Evaluation*). Each evaluation has some metadata (*SML:hasEvaluationMetaData*), and a score (*SML:Score*) that is computed using some metrics (*SML:Metric*), e.g., MAPE (Mean Absolute Percentage Error) and MSE (Mean Squared Error) [3]. The metrics are grouped into categories (*SML:Category*), depending on the algorithm used to build the SML model.

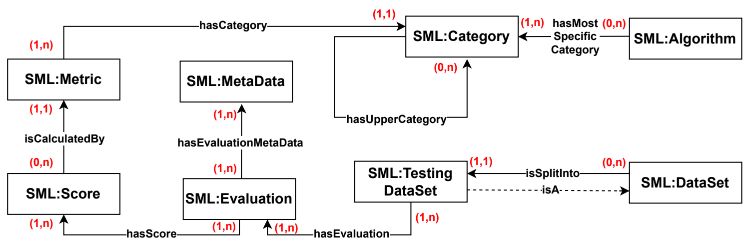

Fig. 5: SML model' evaluation

Next, we present the evaluation of the SML ontology model on two aspects: the efficiency and the performance, after being implemented.

5 Experimental Evaluation

In this section, we outline the experimental procedure that we used to evaluate SML ontology. It is founded on two types of evaluation:

1. *Evaluation of its efficiency:* This involves determining if the concepts and properties (objects and data properties) established in the SML ontology, are capable of overcoming the various challenges described in Section 2, and of meeting the criteria listed in Section 3.
2. *Evaluation of its performance:* This entails studying the response time of the SML ontology, by applying various simple and complex queries on different simulated SML model instances. Such instances are based on several configurations (e.g., increasing the number of SML models, their data-items, the used features in their training data sets, etc.).

5.1 Efficiency Evaluation

In this part, we define the most useful queries (see Table 2), which can be applied to the SML ontology, to help in facing the challenges identified and explained in Section 2. Also, we analyze the ability of these queries in meeting the criteria presented in Section 3. The following link: `https://tinyurl.com/bdrudrtw`, shows the list of queries that we used, expressed in SPARQL[4].

As shown in Table 2, different queries can be used to interrogate the SML ontology model. For instance, the Query *Q1*, where we require the algorithm of a given ML model, meets the challenge 2, which includes mapping models to the

[4] A standard query language that is able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.

algorithms that generated them. The representation of a ML model algorithm is a criterion that is part of the ML representation, hence the relationship between the Query *Q1* and the ML representation criteria. Another example of a query is Query *Q7* that requires the description of the training data set context of a given ML model. The Query *Q7* takes up challenge 3, which covers the context of the data sets (training or testing data sets) features to help build and evaluate the ML models, and meets ML usability and compatibility criteria, as it specifies the ML context.

Table 2: List of useful queries overcoming the required challenges and meeting the needed criteria

	ML Representation Criteria	ML Usability and Compatibility Criteria
Challenge 1	Q5- Retrieve the metadata of a given ML model, with those related to its algorithm, its training data set, its testing data set, and to the evaluations applied to the testing data	
Challenge 2	Q1 - Retrieve the algorithm of a given ML model	
Challenge 3	Q2 - Describe the training data set of a given ML model Q3 - Describe the testing data set of a given ML model	Q7 - Describe the training data context of a given ML model
Challenge 4		Q6 - Find the application domain of each ML model, and give a clear description of this domain
Challenge 5	Q4 - Retrieve the performance of a given ML model (i.e., the scores and the metrics used to calculate the evaluation applied to the testing data)	

5.2 Performance Evaluation

In this part, we took into consideration five different scenarios, to evaluate the performance of the SML ontology model, in terms of the response time, while applying several queries on the SML model. The scenarios were made by simulating different SML models using "Protégé" tool `https://protege.stanford.edu/`, through which all information about data sets for instance were filled, and varying their criteria: (1) their number (the number of SML model instances), (2) the number of the data items used in their training data set, (3) the number of the used features in their training data set, (4) the number of metrics used to compute the score of their testing data set, and (5) the number of their metadata. We display the query response time (ms) in the experiments based on an average of 10 sequential executions for each query. The tests have been carried out using "Stardog" (`https://www.stardog.com/`), a platform for enterprise knowledge graph, run on a Windows 10 Professional machine having an Intel i7-8665U CPU @ 1.90GHz 2.11GHz processor and 1 GB RAM.

Impact of ML Models Instances and their Metadata. In the first scenario (see Figure 6-(a)), we studied the impact of varying the number of ML models instances, when requiring the set of models having a given algorithm (i.e., Linear Regression). In the scenario, we fixed the number of algorithms to 50, linked every ML model (between 100 and 10000 models) to a single algorithm (as defined in the SML ontology), and measured the corresponding query response time. As per the resulted graph curve, the query run time increases quasi-linearly with the increased number of ML models instances. The time evolution is more important between the first two tests, as the number of ML models was increased from 100 to 1000 (with a difference of 900 models), contrary to the rest of the other tests, where the increased number of ML models was more constant (with a difference of 2000/3000 models).

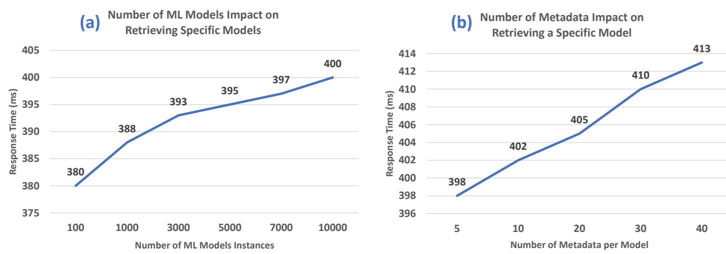

Fig. 6: Number of ML models instances vs ML models metadata impact

In the second scenario (see Figure 6-(b)), we studied the impact of varying the number of metadata related to each ML model instance, when requesting the set of the metadata related to a specific model. In the scenario, we fixed the number of ML models to 500, varied the number of their data items (from 5 to 40), and measured the corresponding query response time. The resulted curve shows that the query execution time evolves linearly with the increased number of metadata defined for each ML model.

Impact of Data Items and Features used in ML Training Data Sets. In Figure 7-(a), we looked at the effect of varying the number of data items included within the training data sets of ML models, while demanding the set of the data items of a specific model. We limited the number of ML models in the tests to 100, varied the numbers of data items used in the training data sets of the ML models (from 100 to 1000), and then, calculated the query response time. The resulted graph demonstrates that as more metadata are defined for each ML model training data set, the query run time increases linearly.

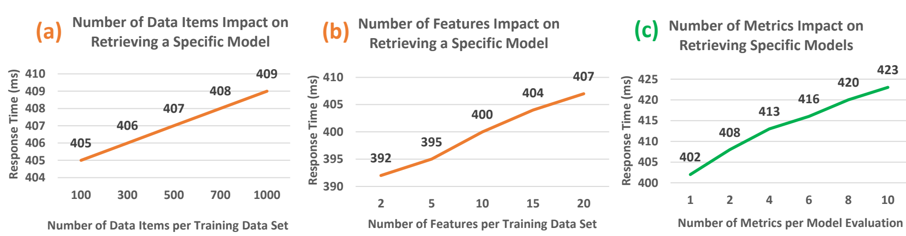

Fig. 7: Training data set data items vs Training data set features vs ML evaluation metrics impact

In Figure 7-(b), we investigated the impact of increasing the number of features used in the training data sets of the ML models, while requesting the set of the features used for a particular model. We set a cap of 1000 ML models for each test, distributed different number of features (from 2 to 20) to each ML model training data set, and then retrieved the corresponding query response time.

The resulted graph shows that the run time evolves linearly with the increased number of features used for each ML model training data set.

Impact of ML Evaluation Metrics. In the last scenario, we looked into the impact of adding metrics in the evaluation score related to the testing data set of ML models (see Figure 7-(c)). In the tests, where we requested the top 3 ML models having the best evaluation score, we fixed the number of ML models to 1000 and varied the number of score metrics from 1 to 10 (e.g., MAPE and MSE [3]), and got the corresponding query response time. From the resulted graph, we can see that as the number of metrics used in the score (to evaluate ML models testing data sets) increases, the run time evolves linearly.

Discussion. In the experimental scenarios, the resulted graphs reveal promising and positive linear curves, indicating that the query execution response time increases linearly with the growing number of ML models instances, their metadata, their data items and features used in their training data, as well as the number of the metrics used to compute the score of their testing data set. This shows a proportionate relation, with almost a constant growth between the various variables employed and the query execution time. The findings also highlight that the curves of the graphs, in which we increased the number of ML models instances (see Figure 6-(a)), where the time jump is equal to 20 ms, and the number of the score metrics of the ML models evaluation (see Figure 7-(c)), where the time jump is equal to 21 ms, are more important than the others, as the time jumps are equal to 15 ms, 4 ms and 15 ms, respectively, in Figure 6-(b), Figure 7-(a) and Figure 7-(b). In Figure 6-(a), the time jump is explained by the huge number of ML models instances we used (10000), and in Figure 7-(c), the time jump can be dedicated to the fact that there were more concepts to reach in the query (*SML:MachineLearningModel*, *SML:TestingDataSet*, *SML:Evaluation*, *SML:Score*, and *SML:Metric*). Moreover, we can see that the increased number of metadata, and the number of features used in ML models training data sets, have the same resulted curve with a time jump equal to 15 ms, despite of varying different variables: from 5 to 40 for metadata number, comparing to 2 to 20 for features number. This can be justified by the very close response times (405 and 407 ms) when the two variables are equal to 20. As for the increased number of data items in the training data sets of ML models, it has the lowest impact on the query response time, with a time jump equal to 4 ms.

6 Conclusion

In this paper, we propose a Semantic Machine Learning Model ontology (SML) that describes and stores ML models' characteristics and operational specifications (e.g., their used algorithms, their metadata, their training and testing data sets, their evaluation, etc.). SML allows to share ML knowledge across different platforms and environments, enabling to ease the comprehension of ML models, as well as their selection in various use cases. After implementing SML, we have evaluated its efficiency and performance in different scenarios, where we varied the number of ML instances models, their metadata, the number of data items and features used in their training data sets, and the number of metrics used to compute their testing data evaluation scores. Our experimental results are promising and encouraging.

As part of our ongoing evaluation of the ontology, we aim to check its consistency, to see if the defined concepts and properties cause any inconsistencies in the ontology's structure. This can be done by running different reasoners. We also seek to evaluate its clarity, to check whether the names or labels of the concepts and properties are clear to users (experts and/or non-experts), and see how easy it is for users to use/understand the ontology. Finally, we aim to use SML ontology in real environments or projects to evaluate further how the ontology can be exploited in practice, and work on the recommendation engine, which will use SML ontology to suggest the most suitable ML model(s) that can be applied in specific contexts and different scenarios.

References

1. Ayranci, P., Lai, P., Phan, N., Hu, H., Kolinowski, A., Newman, D., Dou, D.: Onml: an ontology-based approach for interpretable machine learning. Journal of Combinatorial Optimization pp. 1–24 (2022)
2. Braga, J., Dias, J.L., Regateiro, F.: A machine learning ontology (2020)
3. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. PeerJ Computer Science **7**, e623 (2021)
4. Esnaola-Gonzalez, I.: An ontology-based approach for making machine learning systems accountable. Semantic Web journal (2021)
5. Hassani, H., Silva, E.S., Unger, S., TajMazinani, M., Mac Feely, S.: Artificial intelligence (ai) or intelligence augmentation (ia): what is the future? Ai **1**(2), 8 (2020)
6. Janiesch, C., Zschech, P., Heinrich, K.: Machine learning and deep learning. Electronic Markets **31**(3), 685–695 (2021)
7. Lee, I., Shin, Y.J.: Machine learning for enterprises: Applications, algorithm selection, and challenges. Business Horizons **63**(2), 157–170 (2020)
8. Lindholm, A., Wahlström, N., Lindsten, F., Schön, T.B.: Supervised machine learning. Department of Information Technology, Uppsala University: Uppsala, Sweden p. 112 (2019)
9. Mishra, S., Jain, S.: Ontologies as a semantic model in iot. IJCA **42**(3), 233–243 (2020)
10. Muhammad, I., Yan, Z.: Supervised machine learning approaches: A survey. ICTACT Journal on Soft Computing **5**(3) (2015)
11. Nascimento, N., Alencar, P., Lucena, C., Cowan, D.: A context-aware machine learning-based approach. In: Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering. pp. 40–47 (2018)
12. Ngiam, K.Y., Khor, W.: Big data and machine learning algorithms for health-care delivery. The Lancet Oncology **20**(5), e262–e273 (2019)
13. Publio, G.C., Esteves, D., Ławrynowicz, A., Panov, P., Soldatova, L., Soru, T., Vanschoren, J., Zafar, H.: Ml-schema: exposing the semantics of machine learning with schemas and ontologies. arXiv preprint arXiv:1807.05351 (2018)
14. Rashidi, H.H., Tran, N.K., Betts, E.V., Howell, L.P., Green, R.: Artificial intelligence and machine learning in pathology: the present landscape of supervised methods. Academic pathology **6**, 2374289519873088 (2019)