

Domain-Oriented Multilevel Ontology for Adaptive Data Processing

Man Tianxing[1(✉)], Elena Stankova[4], Alexander Vodyaho[3], Nataly Zhukova[1,2], and Yulia Shichkina[3]

[1] ITMO University, St. Petersburg, Russia
mantx626@gmail.com, nazhukova@mail.ru
[2] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia
[3] St. Petersburg State Electrotechnical University "LETI", St. Petersburg, Russia
aivodyaho@mail.ru, strange.y@mail.ru
[4] St. Petersburg State University, St. Petersburg, Russia
e.stankova@spbu.ru

Abstract. In the data mining domain, the diversity of algorithms and the clutter of data make the knowledge discovery process very unfriendly to many non-computer professional researchers. Meta-learning helps users to modify some aspects of this process to improve the performance of the resulting model. Semantic meta mining is the process of mining metadata about data mining algorithms based on expertise extracted from the knowledge base. The knowledge base is usually represented in the form of ontology. This article proposes a domain-oriented multi-level ontology (DoMO) through merging and improving existing data mining ontologies. It provides the restrictions of the dataset characteristics to help the domain experts describe data set in the form of ontology entities. According to the entities of the data characteristics in DoMO, the users can query the ontology to obtain the optimized data processing process. In this paper, we take the time series classification problem as an example to present the effectiveness of the proposed ontology.

Keywords: Meta-learning · Data mining · Semantic meta mining · Ontology

1 Introduction

In the era of big data, data analysis is everywhere. The number of needed data science specialists is increasing permanently. Also, many IT specialists are ready to use the knowledge discovery process in various domains such as economic, meteorology, etc. [21, 22]. But the diversity of algorithms and the clutter of data make the knowledge discovery process very unfriendly to many non-computer professional researchers. Even for the data researchers, it is still difficult to find the best solutions for specific tasks quickly [23]. An intuitive and easy-to-understand intelligent assistant is needed.

Today meta-learning is very popular since it uses machine learning (ML) algorithms to learn from ML experiments for obtaining the best algorithms and parameters.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12249, pp. 634–649, 2020.
https://doi.org/10.1007/978-3-030-58799-4_46

Melanie Hilario proposed a new optimization solution: Semantic meta mining [2]. It relies on extensive background knowledge concerning data mining (DM) itself.

In the field of semantic meta mining, it is necessary to have a suitable description framework to make clear the complex relationships between tasks, data, and algorithms at different stages in the data mining process. Ontology is a computer-understandable description language. Naturally, it has become the choice of many DM intelligent assistants in various application scenarios.

The existing DM ontologies are usually dedicated to expressing one or several stages of the DM process in detail. This concentration on parts makes them lose the integrity of the description of the DM process. In addition, the different constraints of data set characteristics in different domains to make it difficult to propose a general and applicable description ontology.

This paper proposes a domain-oriented multi-level ontology (DoMO) for the DM process. It integrates existing DM ontologies so that it can describe each stage of DM. We also added restrictions on the description of the dataset at the upper level. We built a sub-ontology, which described the characteristics of dataset and task requirements and named it as "INPUT" ontology because it is the input part of the user's query. With the assistance of experts, it can be applied in specific fields.

As an intelligent assistant, the main purpose of DoMO ontology is to help users:

- Describe the data set in the form of ontology entities.
- Choose the suitable solutions based on the data characteristics and task requirements.
- Obtain the data processing processes of the selected solutions.

The structure of this paper is as follows: Sect. 2 introduces background knowledge and related work. Section 3 presents the architecture of the proposed ontology DoMO. Section 4 describes the ontology content. Section 5 presents the workflow of DoMO and a case study on time series classification. In Sect. 6, the authors present the conclusion and future work.

2 Background

2.1 Meta-learning and Semantic Meta Mining

Meta-learning [1] is defined as the application of ML techniques to past ML experiments, and its purpose is to modify certain aspects of the learning process to improve the performance of the results. Traditional meta-learning treats the learning algorithm as a black box, correlating the observed performance of the output model with the characteristics of the input data. However, the internal characteristics of algorithms with the same input/output type may vary.

Semantic meta mining [2] mines DM metadata through querying DM expertise in the knowledge base. It is different from the general meta-learning:

- Meta-learning methods are data-driven. And semantic meta mining is based on the related expertise and internal relations. So, developers usually represent knowledge in the form of ontology.

- Meta-learning for algorithm or model selection mainly involves mapping the dataset attributes to the observed performance of the algorithm as a black box. The parameters are updated based on experimental results, and the internal mechanisms of the algorithms are not the determining factor. In contrast, semantic meta mining complements the data set description by in-depth analysis and characterization of the algorithm: the basic hypothesis of the algorithm, the optimization goals and strategies, and the structure and complexity of the generated models and patterns.
- Meta-learning focuses on the learning phase of data mining, that is, the performance of the generated model. But semantic meta mining is oriented towards the entire data mining process. Based on the characteristics of the data to be processed and the task requirements, it provides users with entire corresponding solutions.

According to the above analysis, the role of classical meta-learning and semantic meta mining are not conflicting. The learning goals of meta-learning are more detailed (such as the parameters of the algorithms). And semantic meta mining provides the appropriate algorithm selection and formulates the execution process. These suggestions are more general. Such semantic meta mining can usually also solve the cold start problem of meta-learning to ensure that the learning process is in the correct direction.

2.2 Data Mining Process

To avoid meaningless operations in data analysis, it is necessary to have a structured framework to implement data mining effectively and correctly. A suitable DM process model is the basis for building DM ontologies. Today, there exist three common frameworks CRISP-DM [4], SEMMA [5], and KDD [3] to format the DM process. Table 1 shows the structures and corresponding relations of these frameworks.

The KDD model is the process of extracting the hidden knowledge according to databases. KDD requires relevant prior knowledge and a brief understanding of the application domain and goals. An improved model of KDD and its application to the construction of optimal routes in a dynamic network are shown in [6].

CRISP-DM provides a uniform framework and guidelines for data miners. It consists of six phases or stages which are well structured and defined for ontology building.

The SEMMA (Sample, Explore, Modify, Model, and Access) is the data mining method developed by the SAS institute. It offers and allows understanding, organization, development, and maintenance of data mining projects. It helps in providing solutions for business problems and goals.

The KDD process model is iterative and interactive, so that it is too complex as the framework of ontology building. SEMMA is linked to SAS enterprise miner as a logical organization of the functional tools. It has a cycle of five stages or steps. But it doesn't describe the steps "Task understanding" and "Deployment," which we are going to describe in the ontology. Based on the characteristics of several frameworks, the simplicity and completeness of CRISP-DM make it suitable for DM ontology building.

Table 1. The comparison and corresponding processes of CRISP-DM, SEMMA, and KDD.

	KDD	CRISP-DM	SEMMA
Processes	Developing and Understanding of the Application	Business Understanding	
	Creating a Target Data Set	Data Understanding	Sample
	Data Cleaning and Pre-processing		Explore
	Data Transformation	Data Preparation	Modify
	Choosing the suitable Data Mining Task	Modeling	Model
	Choosing a suitable Data Mining Algorithm		
	Employing Data Mining Algorithm		
	Interpreting Mined Patterns	Evaluation	Assessment
	Using Discovered Knowledge	Deployment	

2.3 Existing Ontologies for Data Mining

Recently, many intelligent assistants have been developed to optimize the DM process. Comparative studies are discussed in [7, 8]. Many DM ontologies have also been developed to help users build DM processes.

Panov et al. [9, 10] proposed a data mining ontology OntoDM, which includes formal definitions of basic DM entities, such as DM tasks, DM algorithms, and DM implementations. The definition is based on the general data mining framework presented by Džeroski [8]. This ontology is one of the first depth and heavyweight ontologies used for data mining. But it is just used for the description of DM knowledge, so the algorithm characteristics are not covered.

To allow the representation of structured mining data, Panov et al. developed a separate ontology module, named OntoDT, for representing the knowledge about data types [11]. OntoDT defines basic entities, such as datatype, properties of datatypes, specifications, characterizing operations, and a datatype taxonomy. But the problem in the application of ontoDT is that the basic data information is not enough to help users choose the appropriate algorithm. The OntoDT in this article is intended to use it as an upper-level ontology to help domain experts describe the characteristics of the dataset.

Hilario et al. [12] present the data mining optimization ontology (DMOP), which provides a unified conceptual framework for analyzing data mining tasks, algorithms, models, datasets, workflows, and performance metrics, as well as their relationships. As the authors of the concept of semantic meta mining, they use a large set of customized special-purpose relations in DMOP. But DMOP only covers 3 phases of CRISP-DM. And the structure of the ontology is very complicated and thus unfriendly to non-professional users.

In the existing ontologies, the CRISP-DM process, which is composed of the 6 phases, is the basic framework. As Fig. 1 shows, most ontologies only focus on specific phases (DMOP covers three phases that can be best automated: from data preparation to evaluation; OntoDM the last four phases; OntoDT only provides a general description of the first phase about the data type).

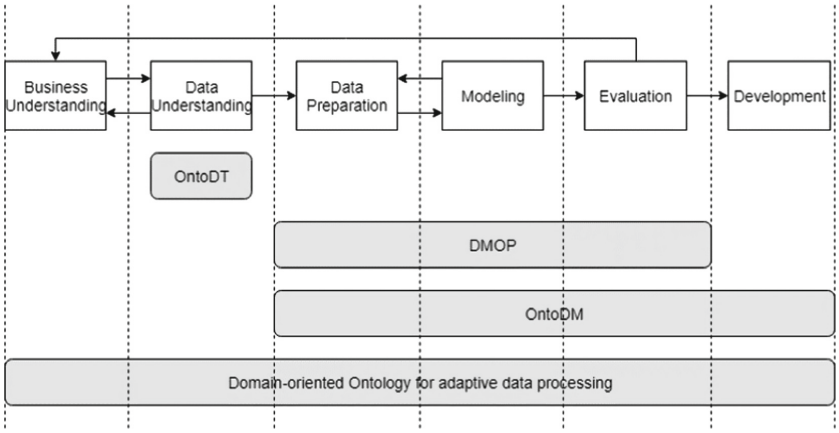

Fig. 1. The comparison of CRISP-DM Phases of Ontology Description (The highlighted grid means covering this phase).

There are several other data mining ontologies currently existing, such as the Knowledge Discovery (KD) Ontology [13], the KDDONTO Ontology [14], the Data Mining Workflow (DMWF) Ontology [15], which are also based on similar ideas.

3 DoMO Architecture

The multi-level architecture of DoMO ontology presents in Fig. 2. In the ontology, the role of each level is as follow:

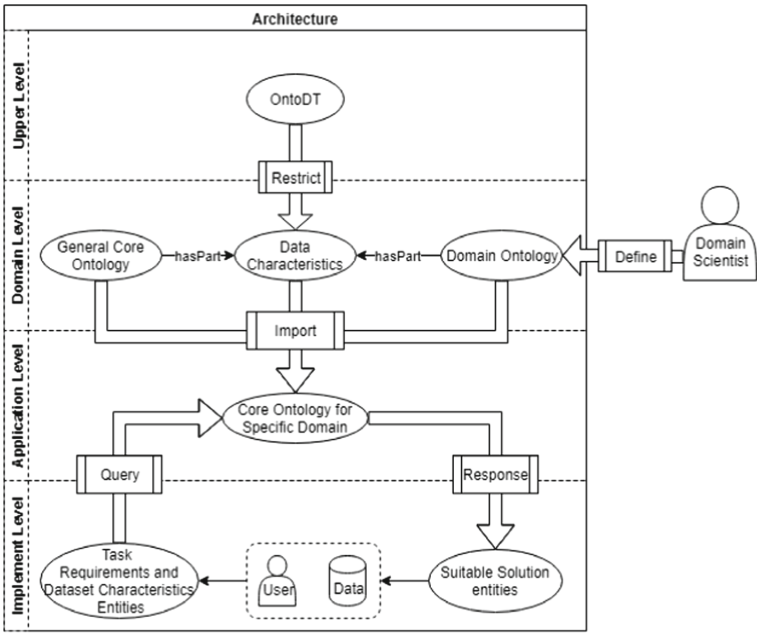

Fig. 2. DoMO Architecture

- Upper Level: The characteristics and task requirements of the data set are the basis for algorithm selection. Data in different fields have different standards for defining characteristics. But the basic properties of the dataset, such as datatype, extended datatype, datatype properties, characterizing datatype operations, datatype value space, are the same. These entities are included in OntoDT. We define it at the upper level as a common data property set
- Domain level: One advantage of DoMO is to use OntoDT's restriction rules to help experts create intelligent assistants suitable for specific fields. This creation process occurs at the domain level. In general core ontology, we name the characteristics of the data in advance. Experts define these characteristics with their knowledge based on upper-level restrictions.
- Application Level: Experts define the data characteristics of the field and import them into general core ontology. It means a core ontology for a specific domain is generated. Its internal structure will be discussed in Sect. 4. Users can query directly on the ontology to get the DM process for specific tasks.
- Implementation level: The generation of user queries and DM processes occurs at the implementation level. According to the characteristics of the data to be processed and task requirements, users obtain suitable solutions. Since the solutions have pre-processers and post-processers, complete DM processes are generated.

The key point of this architecture is to provide restrictions for the description of the domain ontology at the upper level. In the previous work, there is no suitable method to describe the data set in the form of ontology entities. In the general data type ontology OntoDT, the basic properties of the data set are defined. However, these properties cannot directly influence the DM generation process. The selection of the DM algorithm is based on the characteristics of the data set and the requirements of the task. However, the definitions of these characteristics are different in different fields. To make DoMO adaptively process data sets in various fields, we use the OntoDT classes as parameters to specify the definition (value or range) of data characteristics in general core ontology. Domain experts describe domain knowledge or existing domain ontology in general core ontology, making it suitable for data analysis tasks in this domain. An example of the definition in the domain of time series classification (TSC) is shown in Fig. 3.

Then users can query the generated core ontology to obtain a suitable DM process for a specific domain. The workflow is discussed in Sect. 5.1.

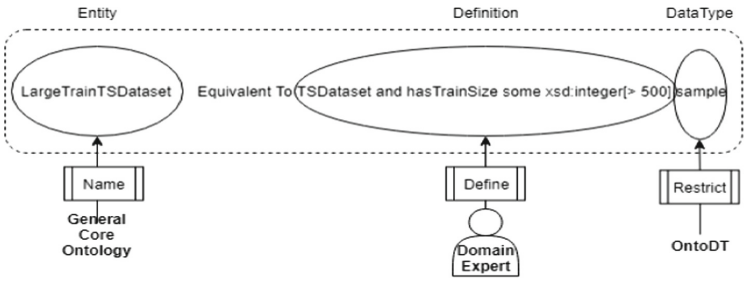

Fig. 3. An example of the definition of data characteristic "LargeTrainTSDataset."

4 DoMO Content

DoMO ontology is composed of two main parts: domain ontology and core ontology. During the initialization phase, core ontology is a general ontology, including an "INPUT" ontology and some other existing DM ontologies (DMOP, OntoDT, OntoDM, and DMWF). Domain ontology is built through defining the existing entities in general core ontology based on the restrictions of OntoDT. When experts import domain knowledge in the form of domain ontology, we obtain a core ontology for a specific domain (See Fig. 4.).

4.1 INPUT Ontology

We create "INPUT" ontology as the input interface for the user query. Its main goals are:

- Define data characteristic entities corresponding to algorithm characteristics.
- Describe the requirements of the DM task, that is, the output of the DM algorithm.
- Supplement the missing algorithm characteristics and measure characteristics in the existing DM ontologies.

INPUT ontology is the part directly associated with the user's queries. It makes the use of ontology more explicit. Users do not need to try to understand the other internal structures of the ontology.

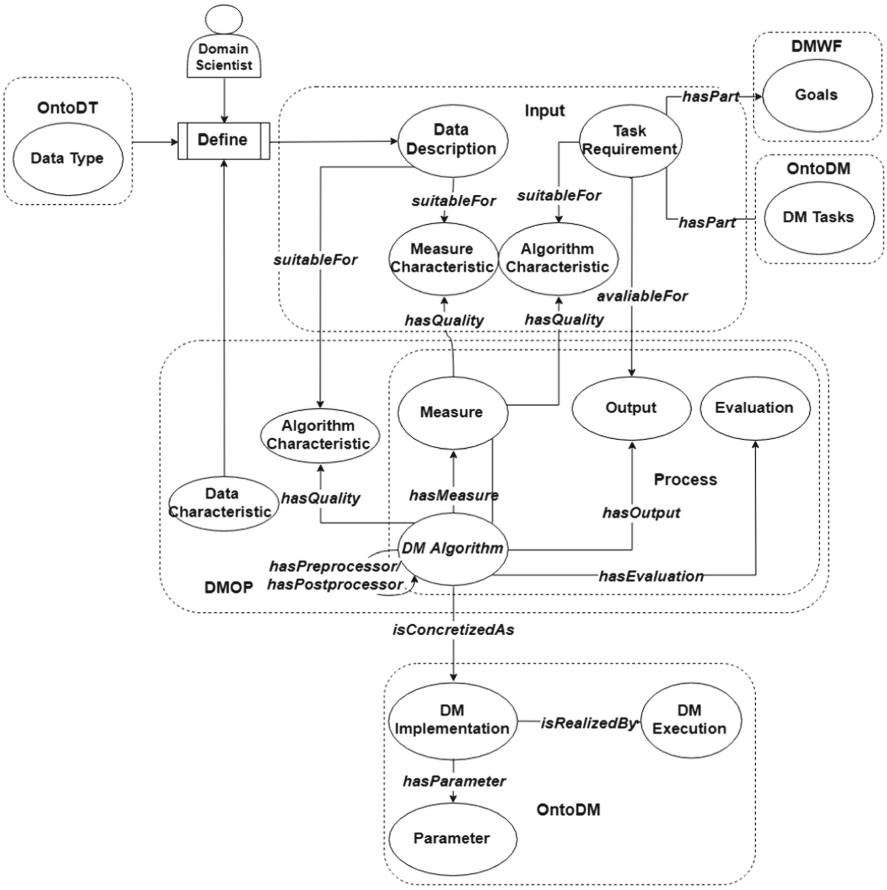

Fig. 4. The general structure of the core ontology

4.2 Existing DM Ontologies for DoMO

Besides, INPUT ontology is also the core part of integrating existing DM ontologies. The integration operation is based on the purpose of generating suitable solutions and processes.

In the process of integration, to reduce the complexity of the ontology, we discarded contents that were useless for this purpose and reconstructed the structures. The main classes in DoMO are shown in Table 2.

The reconstruction processes are as follows:

- OntoDT is fully retained as an upper-level restriction that defines the characteristics of the data.
- The class "Goals" in DMWF and class "DM-Task" in OntoDM are extracted for the description of task requirements.

- Although DMOP provides more than a hundred DM algorithms and their characteristics, we have reconstructed its structure. As components of the DM algorithms, the classes "Measure," "Output," "Evaluation," and "DM Algorithm" itself are included in a new class "Process" so that it is more understandable for the users.
- OntoDM describes the last CRISP- DM phase "Development". The classes "DM Implementation" and "Parameter" in OntoDM are integrated for the possible

Table 2. The main classes in DoMO

Class	Source	Annotation
Data description	INPUT	Describes the dataset characteristics in the form of ontology entities. Domain experts define their value and ranges
Task requirement	INPUT	Describes the task requirements in the form of ontology entities
Measure characteristic	INPUT	Existing DM ontologies do not describe the performance of measures (i.e., distance functions). In the INPUT ontology, we describe and name it "MeasureCharacteristic"
Algorithm characteristic	INPUT/DMOP	Describes the performance of DM algorithms including tolerating some data set defects (such as Missing value, Noise value), suitable for some task requirements (such as two-class, multi-class)
Data type	OntoDT	Provide basic data types that describe the characteristics of the data set (such as sample, label)
Goals	DMWF	Provide a description of the task requirements. It mainly focuses on the generalization of the types of output results
DM-Tasks	OntoDM	Provide a description of the task requirements. It mainly focuses on the description of specific details of the task
Data characteristic	DMOP	Provided by DMOP, the names of the characteristic of the dataset
DM algorithm	DMOP	Describe all DM algorithms that have been designed to perform any of the DM tasks, such as feature selection, missing value imputation, or modeling (or induction)
Measure	DMOP	Describes the distance functions and similarity functions, which usually directly affect the performance of DM algorithms
Output	DMOP	Describe the output models of the DM algorithms (such as decision tree structure, probability distribution structure)
Evaluation	DMOP	Describe the evaluation functions of the DM algorithms (such as external validity model function for clustering algorithms)
DM implementation	OntoDM	Provide DM algorithm implementation scheme and parameter settings
DM execution	OntoDM	Provide executable solutions for DM algorithms (such as R, python package, Weka)
Parameter	OntoDM	Provide parameters for DM algorithms (such as distance threshold, number of clusters and variance threshold for K-means algorithm)

Table 3. The relevant properties in DoMO.

Property	Domains	Ranges	Answering the competency questions
availableFor	INPUT	Characteristics	Given data characteristics or task requirements, which characteristics should the DM algorithms have so that they are suitable for?
suitableFor	INPUT	Characteristics	Given data characteristics or task requirements, which characteristics should the DM algorithms have so that they are available?
hasQuality	Process	Characteristics	Which characteristics do the given process have?
hasPreprocessor hasPostprocessor hasOutput hasMeasure hasEvaluation	DM algorithm	Process	Which processes do the DM algorithm have?
isConcretizedAs	DM Algorithm	DM Implementation	How can we implement the DM algorithm?
hasParameter	DM Implementation	Parameter	Which parameters should we set when we implement the DM algorithm?
isRealizedBy	DM Implementation	DM Execution	Where and how can we execute the DM algorithm?

parameter setting. And "DM Execution" presents where and how to execute the selected algorithms.

In order to build the logical structure of DoMO, the relevant properties are defined in Table 3.

4.3 Statistical Metrics for DoMO

In this paper, we use the statistical ontology metrics from the Protégé software [19] and Bioportal [20]. This includes metrics such as the number of classes and individuals, maximum depth, average number of siblings, maximum number of siblings, sub-class axioms count, disjoint classes axioms count, and annotation assertion axioms count. The values of these statistical ontology metrics for DoMO are presented in Table 4.

Table 4. Statistical metrics for DoMO

Axiom	1610
Logical axiom count	974
Declaration axioms count	636
Class count	626
Object property count	11
Data property count	4
Individual count	0
Maximum depth	12
Maximum number of children	37
Average number of children	3
Classes with a single child	28
Classes with more than 25 children	2
Classes with no definition	626

5 Usage

As long as the structure of the ontologies is reasonable, they can be operated on the corresponding editing software, for instance: protégé. Based on the relations presented in Table 3, users can query for suitable solutions with the following workflow.

5.1 General Workflow

The workflow of DoMO for data analysis in a specific domain is as follow:

1. Based on the restrictions of OntoDT, domain experts define the characteristics of domain data in the form of ontology.
2. Merge the domain ontology and the general core ontology to obtain the core ontology for the specific domain.
3. Manually obtain task requirements and data sets and describe them in the form of ontology entities as the inputs.
4. Execute the selection process on this core ontology for a specific domain.
   a. Input the entities of input-data description and task requirements. Based on the relation "suitableFor", obtain the characteristics which the solutions should have.
   b. According to the relation "hasQuality," obtain the algorithms or measures which have suitable characteristics. If the results are measures, obtain the algorithms according to the relation "hasMeasure."
   c. Choose the most suitable algorithms which meet as many characteristics as possible. They are the selected solutions.
   d. According to the relation "hasPre/Postprocessor," obtain the entire DM process.
   e. According to the relation "hasPart," obtain the process of the selected solutions.
   f. According to the relation "isConcretizedAs," obtain the implementations and parameter variants.
   g. According to the relation "isRealizedBy," obtain the available executions.

5.2 Practical Application in Time Series Classification

DoMO can be flexibly applied to the data analysis process in different fields. As an application example, we construct an ontology oriented to the time series classification (TSC) field. The entities of TSC data characteristics have been named in "INPUT" ontology. For describing the TS datasets in the form of these entities, explicit definitions are needed.

Expert knowledge of the definition of characteristics of TSC data comes from [16]. We define them as Table 5 shows. Then users can represent the TS datasets in DoMO.

Table 5. The definition of TSC data characteristics

Category	Range of class value	Ontology class
TSDataset ByTrainSize	hasTrainSize some xsd:integer [<100] sample	SmallTrainTSDataset
	hasTrainSize some xsd:integer [> =100, <=500] sample	MediumTrainTSDataset
	hasTrainSize some xsd:integer [>500] sample	LargeTrainTSDataset
TSDataset ByTestSize	hasTestSize some xsd:integer [<300] sample	SmallTestTSDataset
	hasTestSize some xsd:integer [>=300, <=1000] sample	MediumTestTSDataset
	hasTestSize some xsd:integer [>1000] sample	LargeTestTSDataset
TSDataset ByLength	hasLengthSize some xsd:integer [<300] timestamp	ShortTSDataset
	hasLengthSize some xsd:integer [>= 300, <=700] timestamp	MediumTSDataset
	hasLengthSize some xsd:integer [>700] timestamp	LongTSDataset
TSDataset ByNoOfClass	hasClassSize some xsd:integer [<10] label	FewClassTSDataset
	hasClassSize some xsd:integer [>=10, <=30] label	MediumClassTSDataset
	hasClassSize some xsd:integer [>30] label	ManyClassTSDataset
TSDataset ByArea	——	DeviceTSDataset
		ECGTSDataset
		ImageTSDataset
		MotionTSDataset
		SensorTSDataset
		SimulatedTSDataset
		SpectroTSDataset

In order to verify the effectiveness of DoMO to help users choose the right solutions, we selected an experimental data 'CinCECGtorso' for verification. For more examples, please refer to [17].

The data set 'CinCECGtorso' is derived from one of the Computers in Cardiology challenges, an annual competition that runs with the conference series of the same name and is hosted on physionet. Data is taken from ECG data for multiple torso-surface sites. There are four classes corresponding to 4 different groups of people [18].

The interaction between the users and DoMO takes place on "INPUT" ontology. Users can describe the dataset and query the corresponding entities of data characteristics in the following form:

"TSDataset and hasTrainSize exactly 40 sample"

Then users can receive the corresponding entity "SmallTrainDataset". Other corresponding entities are received, as Table 6 shows.

Table 6. the corresponding entities of the data set 'CinCECGtorso in INPUT ontology.

Category	Description of 'CinCECGtorso' in DoMO	Corresponding entities
Train size	hasTrainSize exactly 40 sample	SmallTrainTSDataset
Test size	hasTestSize exactly 1380 sample	LargeTestTSDataset
Length	hasLenghSize exactly 1639 timestamp	LongTSDataset
No. of classes	hasClassSize exactly 4 label	FewClassTSDataset
Data area	appliedOn some ECG	ECGTSDataset

The entities "SmallTrainTSDataset", "LargeTestTSDataset", "LongTSDataset" and "ECGTSDataset" are characteristics of the data set and the entities "FewClassTSDataset" means the task requirement is a few classes classification tasks. INPUT ontology allows formulating the tasks in the common form. Then the query for the suitable solutions is:

"*Algorithm*
and suitableFor some SmallTrainTSDataset
and suitableFor some LargeTestTSDataset
and suitableFor some LongTSDataset
and suitableFor some FewClassTSDataset
and suitableFor some ECGTSDataset"

In this experiment, BOSS (Bag of SFA Symbols), COTE (Collection of Transformation E), EE (Elastic Ensemble), MSM_1NN (Move-Split-Merge) and ST (Shapelet Transform) are selected for data set 'CinCECGtorso' by DoMO, since these algorithms are suitable for all the conditions.

We applied all the available TSC algorithms on this data set and gave a rank in Fig. 5. All chosen algorithms are obviously in the upper half and have good performance.

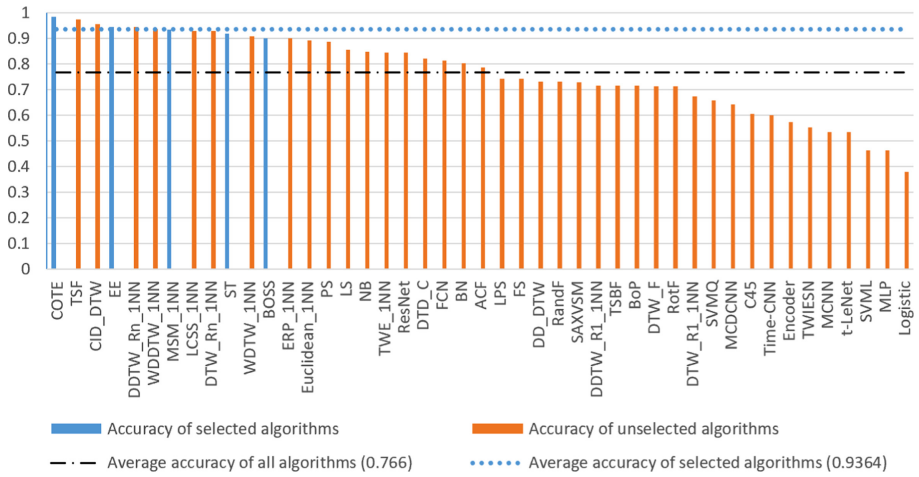

Fig. 5. The accuracy comparison of all TSC algorithms on dataset 'CinCECGtorso'

6 Conclusion

This paper proposes a domain-oriented multi-level ontology (DoMO) for data-adaptive processing. The multi-level structure of the ontology includes the upper level (restrictions described by data characteristics), domain level (definition of domain data characteristics), application level (core ontology for a specific domain), and implementation level (Users' queries and the generation of DM process).

The construction of DoMO includes creating an "INPUT" ontology that describes the characteristics of the data and task requirements and reconstructing and integrating existing DM ontologies. In comparison with existing approaches, DoMO describes the entire data mining process, and the "INPUT" ontology presents datasets in the form of metadata. Due to the intelligibility and portability of the ontology, DoMO can be applied to the data analysis process in different fields with the assistance of domain experts. The application of DoMO in the field of time series classification proved its effectiveness.

Although the ontology is focusing on building the foundation of data mining, it can be used by practitioners in real-world applications to optimize knowledge discovery processes by sequentially querying the suitable solutions based on specific task requirements and data characteristics. Meanwhile, DoMO is intended to be extensible and will continue to be updated to reflect future advancements in using it for building high-quality data-analytical processes rapidly.

Acknowledgments. The paper was prepared in Saint- Petersburg Electrotechnical University (LETI), and is supported by the Agreement? 075-11-2019-053 dated 20.11.2019 (Ministry of Science and Higher Education of the Russian Federation, in accordance with the Decree of the Government of the Russian Federation of April 9, 2010 No. 218), project «Creation of a domestic high-tech production of vehicle security systems based on a control mechanism and intelligent sensors, including millimeter radars in the 76-77 GHz range».

References

1. Jankowski, N., Duch, W., Grąbczewski, K. (eds.) Meta-learning in Computational Intelligence, vol. 358. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20980-2
2. Hilario, M., et al.: A data mining ontology for algorithm selection and meta-mining. In: Proceedings of the ECML/PKDD09 Workshop on 3rd Generation Data Mining (SoKD-09) (2009)
3. Brachman, R.J., Anand, T.: The process of knowledge discovery in databases. In: Advances in Knowledge Discovery and Data Mining, pp. 37–57 (1996)
4. Chapman, P., et al.: CRISP-DM 1.0: Step-by-step data mining guide. SPSS Inc. 9, p. 13 (2000)
5. SAS Enterprise Miner – SEMMA. SAS Institute, September 2014. http://www.sas.com/technologies/analytics/datamining/miner/semma.html
6. Shichkina, Y., Koblov, A.: Reducing the amount of data for creating routes in a dynamic DTN via Wi-Fi on the basis of static data, J. Comput. Netw. Commun. 9128785 (2017). https://doi.org/10.1155/2017/9128785. ISSN: 20907141
7. Serban, F., et al.: A survey of intelligent assistants for data analysis. ACM Comput. Surv. (CSUR) **45**(3), 1–35 (2013)
8. Ristoski, P., Paulheim, H.: Semantic Web in data mining and knowledge discovery: a comprehensive survey. J. Web Semant. **36**, 1–22 (2016)
9. Panov, P., Džeroski, S., Soldatova, L.: OntoDM: an ontology of data mining. In: 2008 IEEE International Conference on Data Mining Workshops, pp. 752–760. IEEE, December 2008
10. Panov, P., Džeroski, S., Soldatova, L.N.: Representing entities in the OntoDM data mining ontology. In: Džeroski, S., Goethals, B., Panov, P. (eds.) Inductive Databases and Constraint-Based Data Mining. Springer, New York *(2010)*. https://doi.org/10.1007/978-1-4419-7738-0_2
11. Panov, P., Soldatova, L.N., Džeroski, S.: Generic ontology of datatypes. Inf. Sci. **329**, 900–920 (2016)
12. Hilario, M., et al.: A data mining ontology for algorithm selection and meta-mining. In: Proceedings of the ECML/PKDD09 Workshop on 3rd Generation Data Mining (SoKD-09) (2009)
13. Záková, M., et al.: Automating knowledge discovery workflow composition through ontology-based planning. IEEE Trans. Autom. Sci. Eng. **8**(2), 253–264 (2010)
14. Diamantini, C., Potena, D., Storti, E.: KDDONTO: an ontology for discovery and composition of KDD algorithms. In: Third Generation Data Mining: Towards Service-Oriented Knowledge Discovery (SoKD 2009), pp. 13–24 (2009)
15. Kietz, J.-U., et al.: Towards cooperative planning of data mining workflows (2009)
16. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. **31**(3), 606–660 (2016). https://doi.org/10.1007/s10618-016-0483-9
17. Tianxing, M., Zhukova, N., Mustafin, N.: A knowledge-based recommendation system for time series classification. In: Proceedings of the 24th Conference of Open Innovations Association FRUCT. FRUCT Oy (2019)
18. PhysioNet/Computing in Cardiology Challenges. https://physionet.org/challenge/
19. Musen, M.A.: The protégé project: a look back and a look forward. AI Matt. **1**(4), 4–12 (2015)
20. Martínez-Romero, M., et al.: NCBO ontology recommender 2.0: an enhanced approach for biomedical ontology recommendation. J. Biomed. Semant. **8**(1), 21 (2017)

21. Stankova, E.N., et al.: OLAP technology and machine learning as the tools for validation of the numerical models of convective clouds. Int. J. Bus. Intell. Data Min. **14**(1–2), 254–266 (2019)
22. Stankova, E.N., Khvatkov, E.V.: Using boosted k-nearest neighbour algorithm for numerical forecasting of dangerous convective phenomena. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11622, pp. 802–811. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24305-0_61
23. Stankova, E.N., Ismailova, E.T., Grechko, I.A.: Algorithm for processing the results of cloud convection simulation using the methods of machine learning. In: Gervasi, O., et al. (eds.) ICCSA 2018. LNCS, vol. 10963, pp. 149–159. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95171-3_13