





A Well-founded Ontology to Support the Preparation of Training and Test Datasets

Lucimar de A. Lial Moura¹^a, Marcus Albert A. da Silva²^b, Kelli de Faria Cordeiro^{1,3}^c
and Maria Cláudia Cavalcanti^{1,2}^d

¹*Departamento de Sistemas e Computação, Instituto Militar de Engenharia (IME), Rio de Janeiro, RJ, Brazil*

²*Departamento de Engenharia de Defesa, Instituto Militar de Engenharia (IME), Rio de Janeiro, RJ, Brazil*

³*Centro de Análise de Sistemas Navais (CASNAV), Rio de Janeiro, RJ, Brazil*

Keywords: Data Preprocessing, Training and Test Datasets, Ontology, UFO, Provenance.

Abstract: In the knowledge discovery process, a set of activities guide the data preprocessing phase, one of them is the data transformation from raw data to training and test data. This complex and multidisciplinary phase involves concepts and structured knowledge in distinct and particular ways in the literatures and specialized tools, demanding data scientists with suitable expertise. In this work, we present PPO-O, a reference ontology of the data preprocessing operators, to identify and represent the semantics of the concepts related to the data preprocessing phase. Moreover, the ontology highlights data preprocessing operators to the preparation of the training and test datasets. Based on PPO-O, Assistant-PP tool was developed, which made it capable to capture the retrospective data provenance during the execution of data preprocessing operators, facilitating the reproducibility and explainability of the dataset created. This approach might be helpful to non-experts users in data preprocessing.

1 INTRODUCTION


The research and application of technologies related to Artificial Intelligence (AI) has been motivating the modern world. According to Gartner¹ group researches, AI is one of the five emerging technologies required in 2020. With large amounts of data available, organizations in almost all sectors of society are focused on exploiting it for the purpose of discovering and gaining knowledge. One of the reasons for the growth of AI comes from the development of powerful algorithms capable of connecting and processing datasets, allowing much broader and deeper analyses.


However, AI came with a diversity of technical terms such as Data Mining (DM), Big Data (BD), Data Science (DS), Machine Learning (ML). This variety of terms might cause difficulties in understanding and sharing knowledge. In this context, (Fayyad


et al., 1996) proposed the Knowledge Discovery and Data Mining Process (KDD), inspired by Knowledge Discovery in Database. (Chapman et al., 2000) proposed Cross-Industry Standard Process of Data Mining (CRISP-DM). Both KDD and CRISP-DM were conceived with the goal of structuring and guiding the discovery of knowledge based on data.


The KDD process describes the data preprocessing phase as a data-centric step, which aims to improve data quality for later consumption by ML algorithms. However, as highlighted in the report (CrowdFlower, 2016) data scientists spend over 80 percent of their time preparing the data. As the area of AI presents many concepts from different perspectives; similarly, the data preprocessing phase also deals with a diversity of terms discussed in a distinct and particular way in the literature (Han et al., 2011) (Faceli et al., 2015) (Goldschmidt et al., 2015) (García et al., 2015). For example, there are different terms to describe the correction operation, which aims to balance the distribution of records during a classification task: correction of prevalence, data balancing or data sampling.

In this sense, when different terms have the same meaning, there is a natural difficulty to understand

^a  <https://orcid.org/0000-0003-1575-860X>

^b  <https://orcid.org/0000-0002-1259-1763>

^c  <https://orcid.org/0000-0001-5161-8810>

^d  <https://orcid.org/0000-0003-4965-9941>

¹<https://www.gartner.com/smarterwithgartner/5-trends-drive-the-gartner-hype-cycle-for-emerging-technologies-2020/>

and choose the data preprocessing operator to execute the transformation of a raw dataset into training and test datasets. Thus, the use of ontologies might be helpful to deal with this terminological problem in the data preprocessing phase.

First of all, ontologies might structure, represent and store knowledge according to a conceptual model, allowing a consensual and uniform view of the data preprocessing scenario. An ontology is rich in semantic expressiveness, and remove or reduce ambiguities, facilitating human and machine understanding. Additionally, such model might guide the data preprocessing operator's execution and support non-expert users learning. Furthermore, it may assist critical requirements in the KDD process, such as reproducibility and explainability (Souza et al., 2020), capturing the transformation of raw data into training and test data.

Currently, ontologies have been adopted to support the KDD process in order to structure and represent the concepts related to the various entities in this domain (Vanschoren and Soldatova, 2010) (Panov et al., 2013) (Keet et al., 2015) (Esteves et al., 2015) (Publio et al., 2018) (Celebi et al., 2020) (Souza et al., 2020). However, these ontologies approaches do not cover most of the details within the data preprocessing phase. Moreover, most of them do not use well-founded conceptual modeling to improve semantic expressiveness and minimize ambiguities.

This article presents the PreProcessing Operators Ontology (PPO-O), which was built based on the Unified Foundational Ontology (UFO) (Guizzardi, 2005). The PPO-O is an ontology applied to the data preprocessing phase of the KDD process; it identifies and represents the semantics of the concepts related to this phase. Moreover, PPO-O evidences the preprocessing operators that transform the raw dataset into training and test datasets. Besides, it simultaneously enables the capture and retrieval of the executed operators through provenance queries. The ontology was developed following the Systematic Approach to Build Ontologies (SABiO) methodology (Falbo, 2014) and modeled using the OntoUML ontology language (Guizzardi, 2005).

This paper is organized according to the following structure. Section 2 provides an overview of some ontologies in the KDD area that are related to this work. Section 3 discusses the main concepts used to develop PPO-O. Section 4 presents PPO-O in detail, and finally, Section 5 makes the conclusion and points out future work.

2 ONTOLOGIES TO SUPPORT THE KNOWLEDGE DISCOVERY PROCESS

The study of ontologies, as a way of expressing knowledge about a domain, has been largely adopted. And, as defined by (Gruber, 1995), ontologies are formal and explicit specifications of the concepts and relationships that can exist in a given domain. Already (Falbo et al., 2002) points out that ontologies are used to describe a uniform and unambiguous domain model of entities and their relationships. While (Nigro, 2007) highlights that ontologies can be used for the DM process, in order to represent the description of the process, and also, to describe its execution, i.e., provenance metadata on the transformation of a given dataset.

The Ontology for Data Mining Experiments (Exposé) (Vanschoren and Soldatova, 2010) was developed with the aim of sharing ML experiment metadata. Exposé highlights the various entities related to the specification of *Dataset* for the *Supervised Classification Task*. While the Ontology for Representing the Knowledge Discovery Process (OntoDM-KDD) prepared by (Panov et al., 2013) uses the CRISP-DM model to represent the main entities in the area of DM, in the context of KDD. OntoDM-KDD represents the taxonomy of entities, which are essential for data preparation.

Data Mining Optimization Ontology (DMOP) developed by (Keet et al., 2015) supports decision making and the meta-learning of the complete DM process. It is a unified conceptual structure and, among the represented concepts, it shows that the execution of the *DM-Process* occurs by a *DM-Workflow*. The *DM-Process* specializes *DM-Operation* executed by *DM-Operator*, which is an algorithm for to execute transformations in *DM-Data*.

MEX Vocabulary presented by (Esteves et al., 2015), has as main purpose to describe terms used in ML experiments and share provenance information, captured with the PROV-O provenance ontology (Lebo et al., 2013). However, MEX Vocabulary not capture information regarding the data preparation process.

ML-Schema proposed by (Publio et al., 2018) is a simple shared schema that provides a set of classes, properties and restrictions that can be used to represent and interchange ML information. Regarding the preprocessing context, it includes the representation of Data (*ML Schema::Data*) and also its specializations, Dataset (*ML Schema::Dataset*) and Feature (*ML Schema::Feature*).

On the other hand, the Ontological Representation

of Relational Databases (RDBS-O) (de Aguiar et al., 2018) is a well-founded reference ontology that represents the structure of relational database systems and, although it is not an ontology in the context of DM, it includes the representation of entities that are important to describe the preprocessing domain, such as *Table*, *Line*, *Line Type* and *Column*.

OpenPREDICT (Celebi et al., 2020) is a unified semantic model of several existing ontologies, among them W3C PROV (Groth and Moreau, 2013), for the prospective, retrospective and workflow evolution provenance of ML scientific workflows. The conceptual modeling was supported by the UFO foundation ontology and its ontological language OntoUML. It captures operations performed by workflow plan steps.

PROV-ML developed by (Souza et al., 2020) is a data representation, also compatible with W3C PROV, from retrospective provenance workflows to support the lifecycle of scientific ML. It represents that a workflow is a composition of data transformations executed by an ML task.

Table 1 shows a comparative model that classifies these related works, according to the following criteria:

- (C1). Operational Ontology: denotes whether the ontology was developed with the Web Ontology Language (OWL)(Horrocks et al., 2004), a World Wide Web Consortium (W3C) standard;
- (C2). Provenance Ontology: informs if the ontology was developed using any of the W3C PROV document models;
- (C3). Foundational Ontology: indicates whether the ontology was developed based on the concepts proposed by UFO; and
- (C4). Details of the Preprocessing Phase: shows whether the ontology considers entities and relationships present in the preprocessing phase, according to the detailing criteria: Partial (P) or Total (T).

Table 1: Summary of Related Works.

Ontology	C1	C2	C3	C4
Exposé	X	-	-	P
OntoDM-KDD	X	-	-	P
DMOP	X	-	-	P
MEX Vocabulary	-	X	-	P
ML-Schema	X	-	-	P
RDBS-O	-	-	X	P
OpenPREDICT	-	-	X	P
PROV-ML	-	X	-	P

Given the above, it was identified that, among the related works, and as far as it was possible to investi-

gate, RDBS-O and OpenPREDICT use the UFO approach, but different from OpenPREDICT, RDBS-O was not built for the KDD context. On the other hand, Exposé, OntoDM-KDD, DMOP, MEX Vocabulary, ML-Schema and PROV-ML, developed in the context of KDD, aim to support all or most of the process, with an emphasis on the DM phase, while the preprocessing phase is given partial or no attention. And, additionally, they are ontologies built without taking into account the precepts of a foundation ontology. In order to fill this gap, this article proposes the PPO-O. To understand this proposal, in the next section, we present a brief discussion on the preprocessing phase, highlighting its specificities, and on the peculiarities of UFO.

3 BACKGROUND

3.1 Data Preprocessing Phase

The preprocessing phase covers all the activities necessary to build the training and test datasets, data that will be inserted in the modeling tool, based on the raw data (Chapman et al., 2000). The data preparation techniques used in this phase aim to improve the quality of raw data, by eliminating or minimizing various problems in the data. For example, the values of the attributes can be numeric or categorical; they may be clean or may contain outliers, incorrect, inconsistent, duplicate or missing values; attributes can be independent or related; datasets can have few or many objects, which in turn can have a small or high number of attributes (Faceli et al., 2015).

There is not a consensus in the literature (Han et al., 2011) (Faceli et al., 2015) (Goldschmidt et al., 2015) (García et al., 2015) about the classification and conceptualization of preprocessing activities.

On the other hand, they all agree on the meaning of an operation in the KDD domain. It is the execution of an operator, a program that implements an algorithm, which specifies a procedure addressed to a KDD activity or task (Keet et al., 2015). Operators are executed on data items² made up of data examples and, in terms of granularity levels, can be a dataset (derived from one or more tables) or just one attribute (a column within a table), or just one instance (a row in a table).

An important concept that must be discussed here is the table concept, because the most DM works use a single fixed format table (Provost and Fawcett, 2016). In the RDBMS domain, a table is represented

²<http://ml-schema.github.io/documentation/>

as a logical structure, that is, an abstraction of the way the data is physically stored, with explicit values in column positions, organized in table lines (Date, 2004). While in the DM domain, a table corresponds to the data itself, i.e., data about certain entities in a given domain, such as customer data, product data, purchase data, etc. Therefore, a dataset can be observed under the following aspects: i) intentional, when it refers to its scheme, which in this context are: columns, features or attributes; and ii) extensional, when it refers to facts, examples, instances or records (Elmasri and Navathe, 2011). As pointed out by (Goldschmidt et al., 2015) the KDD process assumes that the data is organized in a single two-dimensional tabular structure containing facts (organized in rows) and attributes (organized in columns) of the problem to be analyzed.

Another important concept is the supervised learning task. It refers to learning the examples labeled in the training dataset (Han et al., 2011), whose goal is to find a function, model or hypothesis, so that the label of a new example can be predicted. If the label data type is categorical (Cotton, 1999), i.e., its domain is a finite set of unordered values (Faceli et al., 2015), this task is usually known as classification (Faceli et al., 2015).

ML tasks such as the classification task, may need some transformations to be applied to some column values in the dataset. These transformations, also named operations, correspond to the application of an operator on data items. It can be understood, in this way, that the set of enchainned executions of preprocessing operators results in a workflow execution.

On the other hand, as defined in (Celebi et al., 2020) a workflow is a collective of instructions, since its parts have the same functional role in the whole. The execution of a workflow is a description of the process, that is, a set of step-by-step instructions, where each instruction describes an action taken.

In fact, those workflow execution data correspond to provenance data for the generated training and test datasets. It contains structured and linked records of the data derivation paths, referring to the transformation activities those data went through (Souza et al., 2020). This kind of data provenance is known as retrospective provenance. It facilitates the reproducibility and explainability of the training and test datasets.

3.2 Unified Foundational Ontology

In the last decades, ontological analysis has brought significant advances providing a sound foundation for conceptual model development to reach better representations of computational artifacts, especially con-

ceptual schemas (Guizzardi, 2012).

The ontological analysis is based on the use of foundational ontologies (also called top-level ontologies), which provide a set of principles and basic categories (Guarino, 1998). Foundation ontologies apply formal theories to represent aspects of reality and describe, as accurately as possible, the real-world knowledge regardless of the domain, language, or state of affairs.

UFO, initially presented by (Guizzardi, 2005), is a descriptive ontology that represents universals (types) and particulars (substantial or individual), endurants and perdurants. Through continually updating, it has been incorporating ideas from other ontologies such as GFO (Herre et al., 2006) and DOLCE (Gangemi et al., 2002), as well as from the OntoClean methodology (Guarino and Welty, 2004). UFO has three main fragments: UFO-A (Ontology of Endurants), UFO-B (Ontology of Perdurants), and UFO-C (Ontology of Social and Intentional Entities).

Over the years, UFO has been applied to the development of core and domain ontologies in different areas (Guizzardi et al., 2015). For instance, it has been successfully used to provide conceptual clarification in complex domains such as Legal (Ghosh et al., 2017), Brazilian Higher Education (Silva and Belo, 2018), Information Security Incidents (Faria et al., 2019) and Critical Communications (Tesolin et al., 2020).

Figure 1 represents some UFO-A constructs and their relations used in PPO-O conceptual model. This fragment of the UFO deals with the structural aspects of conceptual modeling and referring to objects and entities from the real-world. It also represents types (Universal) and Individuals of these types (Individuals).

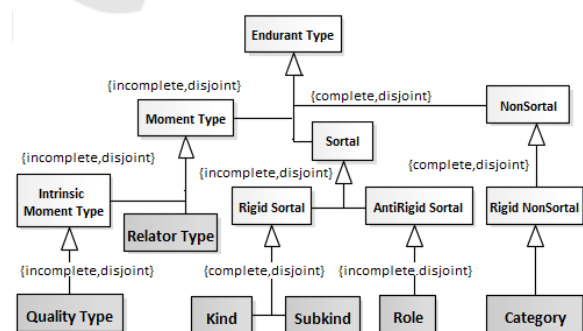


Figure 1: UFO-A fragment, based on (Guizzardi et al., 2018).

Furthermore, UFO-A categorizes types such as *Sortals* carrying identity principle and *NonSortal* aggregating properties in common from different *Sortals*. Thus, *Sortals* describe the real-world objects using

concepts with strong or rigid identity principle such as *Kinds* and *Subkinds* or anti-rigid concepts such as *Roles*, which classify rigid elements under transitory conditions. On the other hand, *NonSortal* constructs generalize different identity principles, based on common characteristics, such as *Category*, which abstracts two or more rigid elements.

Besides, *Endurants* constructs, such as *Sortals*, are existentially independent; on the other hand, *Moment Types*, also known as *Tropes*, are existentially dependent. Thus, a *Quality Type* is an intrinsic property of an Individual, and *Relator Type* plays the role of connecting, relating, or mediating, at least two individuals who share the same foundation (Guizzardi and Wagner, 2008). Hence, a *Quality (Moment)* as "color" cannot exist without a *Sortal* like a car (*kind*), or a *Relator (Moment)* as a marriage cannot exist without two objects (*Sortals*) like a man (*Subkind*) and woman (*Subkind*).

UFO-B deals with objects that persist based on temporal features, representing *Events* acting on *Situations*, *Dispositions*, *Time Points*, as well as the connections between *Endurants* and *Perdurants* (Guizzardi et al., 2013) and (Almeida et al., 2019).

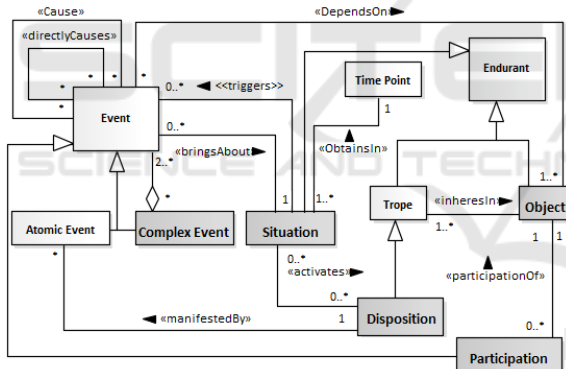


Figure 2: UFO-B fragment, based on (Guizzardi et al., 2013).

As shown in Figure 2, *Situation* is a snapshot of the real-world reality obtained at a particular point in time, modified or created by an *Event* that has mereological features and can be classified as atomic or complex.

An *Atomic Event* has no proper parts and depends on a unique *Object*. On the other hand, *Complex Events* are aggregations of at least two disjoint *subEvents*. *Participation* is an example of *subEvent* that materializes object participation in an *Event*. Furthermore, *Events* can be caused by other *Events*, directly or indirectly. All of these event types are described using axioms in (Guizzardi et al., 2013).

A *Disposition* is a trope existentially dependent on

an object, representing a specific propensity, capacity or feature of an object that can or not be manifested through an Atomic Event. For instance, the event of a heart pumping is the manifestation of the heart's capacity to pump (disposition); the event of a metal being attracted by the magnet is the manifestation of the magnet's disposition to attract metallic material (Guizzardi et al., 2016).

3.3 Methodologies for Building Ontologies

A large number of ontologies have been developed by different groups, under different approaches, and using different methods and techniques (Fernández-López et al., 1997). This way, some relevant methodological approaches in ontology engineering, such as Methontology (Fernández-López et al., 1997), Neon (Suárez-Figueroa et al., 2015) and SABiO (Falbo, 2014), have been proposed as a best practice to build domain ontologies grounded in meta-ontologies or foundational ontologies. These practices make ontological analysis able to explain concepts and relations in the light of such top ontologies, which provides a sound basis for better reality representation applying conceptual modeling.

Moreover, SABiO recommends UFO as foundational ontology and distinguishes between reference and operational ontologies, providing activities that apply to the development of both domain ontologies. A *reference ontology* is a special type of conceptual model because it makes a clear and precise description of the domain entities and might improve communication, learning, and problem-solving. On the other hand, an *operational ontology* is a machine-readable implementation version of the reference ontology.

In this Section, we present some approaches that have been adopted as guidelines on the elaboration of an ontology. Thus, SABiO was adopted as the PPO-O ontology building approach, for its clear distinction between reference and operational ontologies. Moreover, UFO was chosen as the foundational ontology for its conceptual coverage and its large number of case studies found in the literature.

4 PPO-O ONTOLOGY

This research presents the PPO-O ontology, a reference ontology for mastering the preprocessing phase of the KDD process. The purpose of this ontology is to identify and represent the concepts related to the preparation of "cured (Souza et al., 2020)" raw data,

that is, data significantly selected, more organized, easier to analyze and understand. The idea is to facilitate the generation of training and test data to be consumed by ML algorithms.

The preparation of PPO-O was supported by the initial phases of the development process proposed by the SABiO approach, namely: Purpose Identification and Requirements Elicitation, leading to the definition of functional and non-functional requirements (FRs and NFRs, respectively); Ontology Capture and Formalization, giving rise to conceptual modeling of captured concepts; and Design, with the establishment of technological architecture and NFRs for the implementation of the reference ontology.

The following Competency Questions (CQs) are related to the FRs: **CQ1.** What are the types of data preprocessing operator? **CQ2.** Which data structure granularity are needed in the context of KDD? **CQ3.** How can a dataset be described? **CQ4.** What are the data types of the dataset columns? **CQ5.** How can we characterize a labeled dataset? **CQ6.** What is the data type of the target column of the dataset specified for the classification task? **CQ7.** How can a data preprocessing assistant be characterized? **CQ8.** How can a data preprocessing operator execution be registered? **CQ9.** What is the chain of operators that executed to generate a training and test datasets?

4.1 PPO-O Modeling

The ontologies are built to be reused or shared (Fernández-López et al., 1997). In this sense, the semantic models of the PPO-O reuse concepts already formalized by the DMOP, ML Schema and RDBS-O ontologies.

Figure 3 categorizes the taxonomy of the types of data preprocessing operators, according to their role in the data preparation process. The idea is to establish a hierarchy in order to resolve ambiguities. Among the subtypes of a *Data Preprocessing Operator*, we distinguish an operator used to improve data quality as a *Data Cleaning Preprocessing Operator*. On the other hand, to obtain more accurate data, subtypes of *Data Transformation Preprocessing Operator* represent operators that are used for feature engineering. For example, the *Data Reduction Preprocessing Operator* subtype represent those operators to obtain the most appropriate dimensionality for the dataset. In addition, for a dataset with an imbalance in the number of samples of the target attribute, a typical situation in the context of a supervised classification task, a *Data Sampling Correction Preprocessing Operator* is used. And finally, a *Data Partition Preprocessing Operator* is used for partitioning the dataset

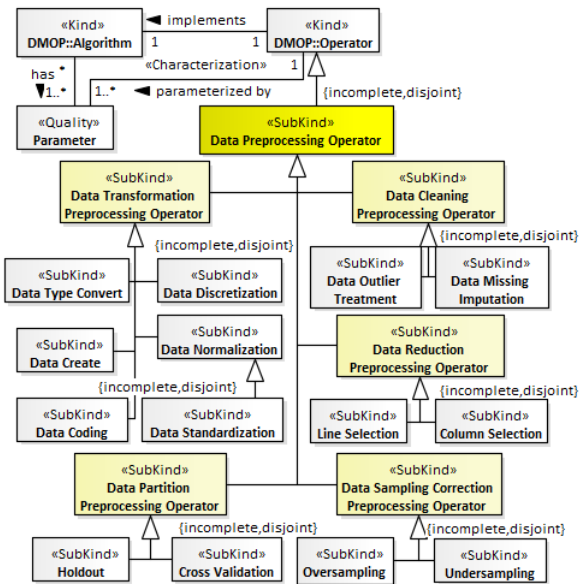


Figure 3: UFO-A based model that represents the categories of data preprocessing operators.

into training and test datasets.

In order to explain the elements related to data used in KDD processes, Figure 4 presents the conceptual model that characterizes a dataset. In relation to the extensional perspective, a *Dataset* is a specialization of *ML Schema::Data*, which may be specialized in different types of representation according to its granularity and identity principles, such as *RDBS-O::Line* and *Column Value*. A *Dataset* is a bidimensional tabular structure for representing data, which is composed of *RDBS-O::Line* instances. Each *RDBS-O::Line* instance is a true proposition (fact) of the problem to be analyzed, and it is instantiated according to a *RDBS-O::Line Type*. Thus, a *Dataset* is described by a *RDBS-O::Line Type*, which is said to be its schema. The *RDBS-O::Line Type* aggregates a set of *RDBS-O::Columns*, where each column represents an attribute that describes some *Column Values*. A set of *Column Values*, described by the different *RDBS-O::Columns* that compose a *RDBS-O::Line Type*, constitute a *RDBS-O::Line*, or a fact. Finally, each *RDBS-O::Column* is defined by a *RDBS-O::Data Type*, which is specialized in *Qualitative Data Type*, when the domain of values is categorical, or *Quantitative Data Type*, when it represents numerical values.

While preprocessing data in the context of a KDD process, a set of measured values can help in characterizing a dataset and its columns. A *Dataset Characteristic*, for example, may be its dimensionality (numbers of lines and columns), or the proportion of missing values. Also, descriptive statistics may character-

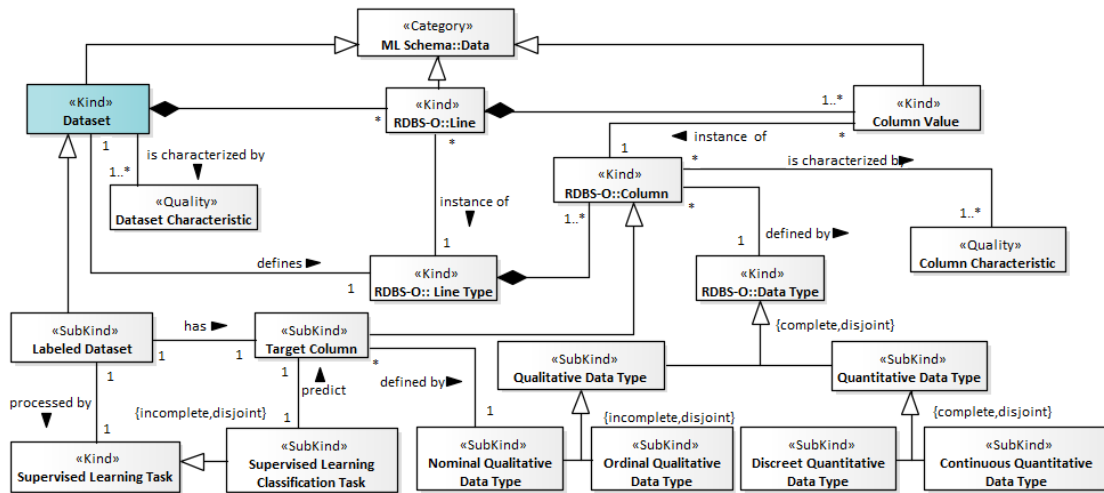


Figure 4: UFO-A based model for the dataset concept.

ize a column (*Column Characteristic*), such as mean and standard deviation. These measured values can be obtained from the *RDBS-O::Lines* or *Column Values* that constitute a dataset or a column, respectively. This characterization facilitates the understanding of the data under analysis, and the identification of the need to apply other preprocessing operators.

A *Labeled Dataset* is a type of *Dataset* specifically created to be processed by a *Supervised Learning Task*. When the goal is to generate a model to predict the value of a *Target Column*, defined by a *Nominal Qualitative Data Type*, then it means that the dataset is to be processed by a *Supervised Learning Classification Task*. Other specializations of *Supervised Learning Tasks* were not within the scope of the present work.

Figure 5 shows, the *Data Preprocessing Assistant* tool is a computational *Software* resource, whose purpose is to support the execution of a *Data Preprocessing Workflow Plan*. This tool uses the metadata of the dataset and its columns (*Dataset Characteristic* and *Column Characteristic*) to facilitate the choice of *Data Transformations*, systematically. And, in parallel, this tool captures the retrospective provenance from each *Operator Execution*, registering the transformation implementation (*Data Preprocessing Executable Operator*) that each *RDBS-O::Column* of a raw dataset is submitted to, in order to generate the corresponding training and test datasets. In this way, the execution of each data transformation is encapsulated by a data capture task, which occurs through a function call, a program execution. Note that the executions belong to a *Data Preprocessing Executable Workflow*, which implements a *Data Preprocessing Workflow Plan*, initially defined by the Assistant tool. Figures 3, 4 and 5 present models grounded on UFO-

A. *Kinds* and *subkinds* categorize the operator and the data and data type hierarchies. Additionally, note that a *relator* is used to bring to light the execution of an operator over a dataset column. This representation is specially important for the provenance capture, as it distinguishes concepts such as the executable code, the transformation it implements, and the relationship of the code when it runs on some dataset column.

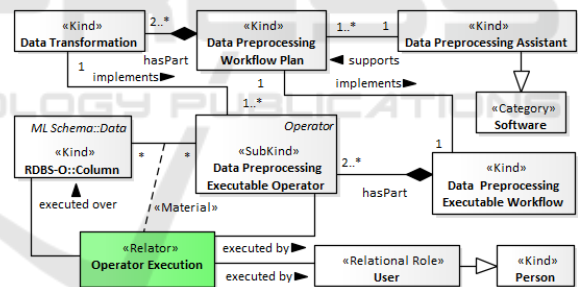


Figure 5: UFO-A based model to capture provenance information of the operator's execution.

The representation of the dynamic aspects of PPO are grounded on UFO-B fragment, which is summarized in the metamodel of Figure 2. The model in Figure 6 represents the events of the preprocessing phase that are necessary for the construction of a training and test datasets. It identifies the situations that triggers each event, and the participants involved. It shows that a *Data Preprocessing* is a complex event composed of two sub-events. One of them is the *Exploratory Data Analysis*, which might identify, among other problems, columns with outliers, null or blank data, denormalized data or unbalanced data. As a consequence of this identification, a *Situation* named (*Column Characteristic Identified*) represents these anomalies, which might activate *Dispo-*

sitions that are inherent capabilities or abilities (*inheresIn*) of *Data Preprocessing Operators*, such as outlier removers, data imputation operators, data normalization, or data balance operators.

In other words, since the *Exploratory Data Analysis* event identifies a suspicious column, it activates one of the *Data Preprocessing Operators* dispositions, which is manifested through another sub-event, named *Data Preprocessing Operator Execution*. As a final step, the *Situation Processed Column* represents the identified anomaly and duly resolved.

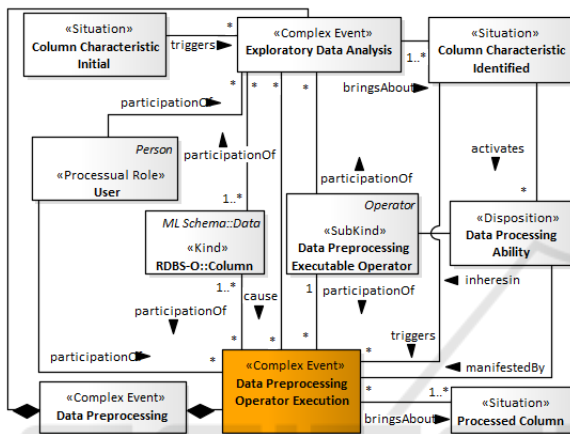


Figure 6: UFO-A and UFO-B based model of the data preprocessing event.

4.2 PPO-O Evaluation and Application

The PPO-O evaluation was carried out through the verification activity, according to the evaluation support process of the SABiO methodology. This activity involves the identification of the answers to the competence questions presented previously in this Section, using the concepts and relationships that constitute the ontology, as detailed below:

- CQ1. Data Preprocessing Operator *specializes* Data Cleaning Preprocessing Operator, Data Reduction Preprocessing Operator, Data Transformation Preprocessing Operator, Data Sampling Correction Preprocessing Operator and Data Partition Preprocessing Operator;
- CQ2. ML Schema::Data *specializes* Dataset, RDBS-O::Line and Column Value;
- CQ3. Dataset *has* RDBS-O::Line, which is an *instance of* RDBS-O::Line Type, which is *defined by* Dataset; The RDBS-O::Line *has* Column Value(s), which are *instance(s) of* RDBS-O::Column; The RDBS-O::Column *isPart of* RDBS-O::Line Type and is *defined by* RDBS-O::Data Type;

- CQ4. RDBS-O::Data Type *specializes* Qualitative Data Type which is *specialized in* Nominal Qualitative Data Type and Ordinal Qualitative Data Type; RDBS-O::Data Type *specializes* Quantitative Data Type which is *specialized in* Discreet Quantitative Data Type and Continuous Quantitative Data Type;
- CQ5. The Labeled Dataset is a *specialization of* Dataset, which is *processed by* a Supervised Learning Task and *has a* Target Column, which is a *specialization of* RDBS-O::Column;
- CQ6. The Supervised Learning Classification Task *predicts the* Target Column *defined by* Nominal Qualitative Data Type;
- CQ7. The Data Preprocessing Assistant *is a Kind of* Software that *supports a* Data Preprocessing Workflow Plan, which is a *collection of* Data Transformations; a Data Transformation *is implemented by* a Data Preprocessing Executable Operator, whose *execution over* each RDBS-O::Column is captured by the Relator Operator Execution, which is *executed by* a Person playing the User Relation Role;
- CQ8. The Data Preprocessing Operator Execution is a *Complex Event manifesting a* Data Processing Ability, which *inheres in* a Data Preprocessing Executable Operator; The Data Processing Ability is *activated by* a Column Characteristic Identified, which had been *broughtAbout by* the Exploratory Data Analysis *Complex Event*; The Data Preprocessing Operator Execution *is captured by the* Relator Operator Execution, with the *participationOf* a Person playing the User Processual Role; and
- CQ9. The Data Preprocessing Executable Workflow is a *collection of* Data Preprocessing Executable Operator, whose *execution over* each RDBS-O::Column is captured by the Relator Operator Execution during the Data Preprocessing Operator Execution *Complex Event*, that occurs with the *participationOf* a Person playing the User Processual Role when use Data Preprocessing Assistant.

The conceptualization of the data preprocessing phase was made explicit through conceptual modeling grounded on UFO-A and UFO-B. Based on these models (PPO-O) it was possible to conceive an architecture, shown in Figure 7, that was implemented as a PreProcessing Assistant (Assistant-PP)³ tool, using the Python Programming Language with the Streamlit

³<https://github.com/LucimarLial/AssistantPP>

Framework and PostgreSQL RDBMS. The main purpose of this tool is to guide a *non-expert user* in the selection of *data preprocessing operators* and, in parallel, to capture structured and rich provenance data, for each *data preprocessing operator execution*, which are stored in the (ProvOp) layer.

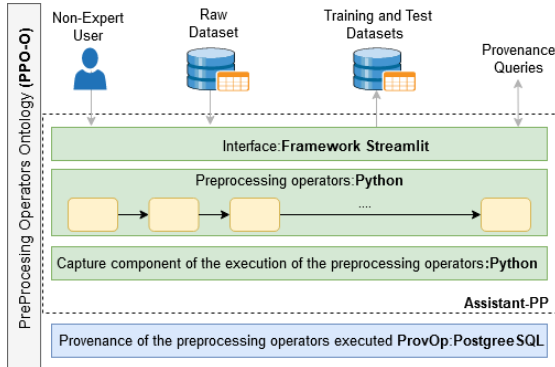


Figure 7: Assistant-PP architecture components.

The PPO-O validation activity occurred with the instantiation of the competence question (CQ9), during the execution of the Assistant-PP and with provenance queries from the ProvOp layer to verify if it was able to capture the workflow generated during creation of a training and test datasets.

As a test case, we used the Adult (Dua and Graff, 2017) dataset specified for the *Supervised Learning Classification Task*, whose goal is to predict whether an adult’s income exceeds \$50K per year, based on census data. Table 2 summarizes the number of *RDBS-O::Lines* and *RDBS-O::Columns* processed through the execution of a *Data Preprocessing Executable Workflow*. Note that the output datasets has a smaller numbers of columns and lines. This is due to the impact of operators such as *Data Reduction - Column Selection* and *Data Sampling Correction - Undersampling*. In addition, test dataset does not include the target column.

Table 2: Input and output *Datasets* processed by the Data Preprocessing Executable Workflow (CQ9).

Dataset		Line	Column
Input	Raw	48842	15
	Training	16250	14
Output	Test	14653	13

Getting into the preprocessing workflow execution details, Table 3 shows all the *Operator Executions* that took place in the workflow execution. It correlates each Adult dataset *RDBS-O::Column*, to the *Data Preprocessing Executable Operator* type, according to the categorization shown in Figure 3: Data

Cleaning (DC), Data Reduction (DR), Data Transformation (DT), Data Partition (DP) and Data Sampling Correction (DS). It is possible to see in Table 3 that it involved 42 operator executions, showing how hard it is to keep track of the preprocessing operations, which is why we need to keep a record of each one of them.

Analyzing Table 3, we can see that most of the Adult dataset columns were processed by DP and DS operators, which are the most general, i.e., independent of the data type of column, unlike DC and DT operators that take into account the data type of column. Moreover, with the analysis of the particular properties of the column values only a few columns should be cleaned and transformed to obtain more accurate data to ML algorithms. As example, standardization for columns with quantitative data type and coding for columns with qualitative data type, such as the sex column, which was coded by a *Data Coding Operator*, deriving two new columns, Sex_1 (1=male) and Sex_2 (0=female) and, in this case, DP and DS operators will be processed in Sex_1 and Sex_2 and no longer in Sex. Finally, the Assistant-PP may suggest the elimination (DR operator) of columns with low relevance after the analysis of the correlation between the predictive columns and the target column. This was the case of *Capital-loss* and *Fnlwgt*.

Table 3: Operator Execution by the Data Preprocessing Executable Workflow (CQ9).

Columns	DC	DR	DT	DP	DS
Age	X		X	X	X
Capital-gain	X		X	X	X
Capital-loss		X			
Country	X			X	X
Education				X	X
Education-num	X		X	X	X
Fnlwgt		X			
Hours-per-week	X		X	X	X
Marital-status				X	X
Occupation	X		X	X	X
Race				X	X
Relationship				X	X
Sex			X	X	X
Workclass	X			X	X
Target			X	X	X

All these operator executions are captured by the Assistant-PP tool and registered in our provenance database. Figure 8 shows an example of a query in such database, which lists the operators, and their corresponding categories, that were applied to columns *Age*, *Occupation*, *Fnlwgt*, *Sex*, *Sex_1*, *Sex_2* and *Target*. Note that Assistant-PP is able to provide a fine grain provenance record, where it is possible to keep

track of actions on each column of a dataset.

```
SELECT id, name_column, function_operator, name_operator, type_operator
FROM public.tb_log_operation WHERE number_workflow=1
AND name_column in ('Age','fnlwtg','Occupation','Sex','Sex_1','Sex_2','Target')
ORDER BY name_column, type_operator|
```

name_column	function_operator	name_operator	type_operator
Age	DropOutlier	Data Outlier Treatment	Data Cleaning
Age	TrainTestSplit	Houldout	Data Partition
Age	RandomUnderSampler	Undersampling	Data Sampling
Age	StandardScaler	Data Standardization	Data Transformation
Age	KBinsDiscretizer	Data Discretization	Data Transformation
fnlwtg	DropQuantitativeColumn	Column Selection	Data Reduction
Occupation	ImputationUnknown	Data Missing Imputation	Data Cleaning
Occupation	TrainTestSplit	Houldout	Data Partition
Occupation	RandomUnderSampler	Undersampling	Data Sampling
Occupation	OrdinalEncoder	Data Coding	Data Transformation
Sex	OneHotEncoder	Data Coding	Data Transformation
Sex_1	TrainTestSplit	Houldout	Data Partition
Sex_1	RandomUnderSampler	Undersampling	Data Sampling
Sex_2	TrainTestSplit	Houldout	Data Partition
Sex_2	RandomUnderSampler	Undersampling	Data Sampling
Target	TrainTestSplit	Houldout	Data Partition
Target	RandomUnderSampler	Undersampling	Data Sampling
Target	StandardScaler	Data Standardization	Data Transformation
Target	LabelEncoder	Data Type Convert	Data Transformation

Figure 8: ProvOp - provenance query of part of the Data Preprocessing Executable Workflow captured.

Imputation of qualitative data
['Workclass', 'Occupation', 'Country']

Choose an imputation option:

Input with unknown

Imputation of quantitative data
['Education_Num']

Choose an imputation option:

Select an option

- Input with -1
- Input with 0
- Input with mean
- Input with median
- Input with mode

Figure 9: Assistant-PP - Data Preprocessing Executable Operator by RDBS-O::Data Type.

It is worth to highlight that the Assistant-PP tool incorporates the knowledge raised by the models presented in Section 4.1, and guides a *non-expert user* to select the appropriate *Data Preprocessing Operator*, according to the *RDBS-O::Data Type* of the *RDBS-O::Column*. For example, the "unknown" imputation option is indicated only for columns of a *Qualitative Data Type*. Other imputation options, such as mode, median, mean, values 0 or -1, are indicated for columns of a *Quantitative Data Type*. Both examples are illustrated in Figure 9.

More examples of such expertise are shown in

Figure 10, where the *Data Discretization* operator is chosen for *Continuous Quantitative Data Type* columns (e.g. Age, Education_Num, etc.). Note that *Capital-loss* and *Fnlwtg* columns were excluded by DR operator (Table 3) previously during the process, and therefore, they do not appear in the list of available columns for *Data Discretization*. On the other hand, the *Data Standardization* operator is chosen for the *Discreet and Continuous Quantitative Data Type* columns. In this case, all available columns of this type are processed. Finally, the *Data Coding* operator, in Figure 11, is recommended only for *Qualitative Data Type* columns (e.g. Workclass, Sex, etc.).

Data discretization

Quantitative columns

Continuous

Discretization explanation

n_bins:

2 20

encode

onehot-dense

strategy

quantile

Inform the columns to apply the discretization:

Age X

- Education_Num
- Capital_Gain
- Hours_per_week

Figure 10: Assistant-PP - Data Preprocessing Executable Operator by RDBS-O::Data Type.

Data normalization and standardization

Discreet and Continuous

Normalization and Standardization explanation

Choose the method:

Standardization

Data coding

Qualitative columns

Nominal (OneHot Encoder)

OneHot Encoder explanation

Inform the columns to apply the encoding:

Sex X

- Workclass
- Education
- Martial_Status
- Occupation
- Relationship
- Race
- Sex
- Country

Figure 11: Assistant-PP - Data Preprocessing Executable Operator by RDBS-O::Data Type.

5 CONCLUSION AND FUTURE WORK

In this paper we present PPO-O, a domain reference ontology for the preprocessing phase of the KDD process, built using UFO ontological foundations. The idea is to support the non-expert user in data preprocessing, indicating the appropriate operators for the transformation of a cured raw dataset into a training and test datasets. It was developed following the guidelines of the SABiO ontology engineering approach. Its focus is on the supervised learning classification task, and it reused concepts from KDD and RDBMS ontologies, which incorporate already grounded concepts that are essential to clarify the semantics of the preprocessing phase.

The PPO-O evaluation was carried out by answering the competence questions previously defined, and showed the completeness of the represented concepts and relationships. In addition, a tool named Assistant-PP was built based on the PPO-O ontology, which made it capable of capturing the retrospective data provenance during the execution of preprocessing operators. Therefore, it was shown that it attends the reproducibility and explainability requirements for a preprocessing workflow executed.

As future work, we intend to extend the PPO-O to incorporate other data preprocessing operators, as well as other ML tasks, such as operators applied to the Supervised Regression Task. Also, we plan to develop a new version of the assistant tool, using an operational version of the PPO-O ontology.

REFERENCES

- Almeida, J. P. A., de Almeida Falbo, R., and Guizzardi, G. (2019). Events as entities in ontology-driven conceptual modeling. In Laender, A. H. F., Pernici, B., Lim, E., and de Oliveira, J. P. M., editors, *Conceptual Modeling - 38th International Conference, ER 2019, Salvador, Brazil, November 4-7, 2019, Proceedings*, volume 11788 of *Lecture Notes in Computer Science*, pages 469–483. Springer.
- Celebi, R., Moreira, J. R., Hassan, A. A., Ayyar, S., Ridder, L., Kuhn, T., and Dumontier, M. (2020). Towards fair protocols and workflows: the openpredict use case. *PeerJ Computer Science*, 6:e281.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R., et al. (2000). Crisp-dm 1.0: Step-by-step data mining guide. *SPSS inc*, 9:13.
- Cotton, P. (1999). Iso/iec fcd 13249-6: 1999 sql/mm saf-005: Information technology-database languages-sql multimedia and application packages-part 6: Data mining.
- CrowdFlower (2016). Cfds16.pdf. <http://www2.cs.uh.edu/~ceick/UDM/CFDS16.pdf>. (Accessed on 11/21/2020).
- Date, C. J. (2004). *Introdução a sistemas de bancos de dados*. Elsevier Brasil.
- de Aguiar, C. Z., de Almeida Falbo, R., and Souza, V. E. S. (2018). Ontological representation of relational databases. In *ONTOBRAS*, pages 140–151.
- Dua, D. and Graff, C. (2017). UCI machine learning repository.
- Elmasri, R. and Navathe, S. B. (2011). *Database systems*, volume 9. Pearson Education Boston, MA.
- Esteves, D., Moussallem, D., Neto, C. B., Soru, T., Usbeck, R., Ackermann, M., and Lehmann, J. (2015). Mex vocabulary: a lightweight interchange format for machine learning experiments. In *Proceedings of the 11th International Conference on Semantic Systems*, pages 169–176. ACM.
- Faceli, K.; Lorena, A., Gama, J., and Carvalho, A. (2015). *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina. Edição 1*. LTC Editora, 2015. 378 f.
- Falbo, R. d. A. (2014). Sabio: Systematic approach for building ontologies. In *ONTO. COM/ODISE@ FOIS*.
- Falbo, R. d. A., Guizzardi, G., and Duarte, K. C. (2002). An ontological approach to domain engineering. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 351–358. ACM.
- Faria, M. R., de Figueiredo, G. B., de Faria Cordeiro, K., Cavalcanti, M. C., and Campos, M. L. M. (2019). Applying multi-level theory to an information security incident domain ontology. In *ONTOBRAS*.
- Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., et al. (1996). *Advances in knowledge discovery and data mining*, volume 21. AAAI press Menlo Park.
- Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. *AAAI-97 Spring Symposium Series*.
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening ontologies with dolce. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181. Springer.
- García, S., Luengo, J., and Herrera, F. (2015). *Data preprocessing in data mining*. Springer.
- Ghosh, M. E., Abdulrab, H., Naja, H., and Khalil, M. (2017). Using the unified foundational ontology (ufo) for grounding legal domain ontologies. In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2: KEOD, (IC3K 2017)*, pages 219–225. INSTICC, SciTePress.
- Goldschmidt, R., Passos, E., and Bezerra, E. (2015). *Data Mining, Conceitos, Técnicas, algoritmos, orientações e aplicações. Edição 2*. Elsevier, 2015. 296 f.
- Groth, P. and Moreau, L. (2013). W3c prov: An overview of the prov family of documents.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International*

- Journal of Human-Computer Studies*, 43(5):907 – 928.
- Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98)*, June 6-8, Trento, Italy, volume 46. IOS press.
- Guarino, N. and Welty, C. A. (2004). An overview of ontoclean. In *Handbook on ontologies*, pages 151–171. Springer.
- Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. PhD thesis, University of Twente, The Netherlands.
- Guizzardi, G. (2012). Ontological meta-properties of derived object types. In *International Conference on Advanced Information Systems Engineering*, pages 318–333. Springer.
- Guizzardi, G., Fonseca, C. M., Benevides, A. B., Almeida, J. P. A., Porello, D., and Sales, T. P. (2018). Endurant types in ontology-driven conceptual modeling: Towards ontouml 2.0. In *International Conference on Conceptual Modeling*, pages 136–150. Springer.
- Guizzardi, G., Guarino, N., and Almeida, J. P. A. (2016). Ontological considerations about the representation of events and durants in business models. In *International Conference on Business Process Management*, pages 20–36. Springer.
- Guizzardi, G. and Wagner, G. (2008). What's in a relationship: an ontological analysis. In *International Conference on Conceptual Modeling*, pages 83–97. Springer.
- Guizzardi, G., Wagner, G., Almeida, J. P. A., and Guizzardi, R. S. (2015). Towards ontological foundations for conceptual modeling: The unified foundational ontology (ufo) story. *Applied ontology*, 10(3-4):259–271.
- Guizzardi, G., Wagner, G., de Almeida Falbo, R., Guizzardi, R. S., and Almeida, J. P. A. (2013). Towards ontological foundations for the conceptual modeling of events. In *International Conference on Conceptual Modeling*, pages 327–341. Springer.
- Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., and Michalek, H. (2006). General formal ontology (gfo)-a foundational ontology integrating objects and processes [version 1.0].
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., Dean, M., et al. (2004). Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79):1–31.
- Keet, C. M., Ławrynowicz, A., d'Amato, C., Kalousis, A., Nguyen, P., Palma, R., Stevens, R., and Hilario, M. (2015). The data mining optimization ontology. *Journal of web semantics*, 32:43–53.
- Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., and Zhao, J. (2013). Prov-o: The prov ontology. *W3C recommendation*, 30.
- Nigro, H. O. (2007). *Data Mining with Ontologies: Implementations, Findings, and Frameworks: Implementations, Findings, and Frameworks*. IGI Global.
- Panov, P., Soldatova, L., and Džeroski, S. (2013). Ontodm-kdd: ontology for representing the knowledge discovery process. In *International Conference on Discovery Science*, pages 126–140. Springer.
- Provost, F. and Fawcett, T. (2016). Data science para negócios. *Tradução de Marina Boscatto*.
- Publio, G. C., Esteves, D., Ławrynowicz, A., Panov, P., Soldatova, L., Soru, T., Vanschoren, J., and Zafar, H. (2018). MI-schema: Exposing the semantics of machine learning with schemas and ontologies. *arXiv preprint arXiv:1807.05351*.
- Silva, C. and Belo, O. (2018). A core ontology for brazilian higher education institutions. In *Proceedings of the 10th International Conference on Computer Supported Education - Volume 2: CSEDU*, pages 377–383. INSTICC, SciTePress.
- Souza, R., Azevedo, L. G., Lourenço, V., Soares, E., Thiago, R., Brandão, R., Civitarese, D., Brazil, E. V., Moreno, M., Valdúriez, P., et al. (2020). Workflow provenance in the lifecycle of scientific machine learning. *arXiv preprint arXiv:2010.00330*.
- Suárez-Figueroa, M. C., Gomez-Perez, A., and Fernández-López, M. (2015). The neon methodology framework: Ascenario-based methodology for ontology development. *Applied Ontology*, 10:107–145.
- Tesolin, J., Silva, M., Campos, M., Moura, D., and Cavalcanti, M. C. (2020). Critical communications scenarios description based on ontological analysis. In *ONTOBRAS*.
- Vanschoren, J. and Soldatova, L. (2010). Exposé: An ontology for data mining experiments. In *International workshop on third generation data mining: Towards service-oriented knowledge discovery (SoKD-2010)*, pages 31–46.