VIS4ML: An Ontology for Visual Analytics Assisted Machine Learning

Dominik Sacha, Matthias Kraus, Daniel A. Keim Member, IEEE, and Min Chen Member, IEEE

Abstract—While many VA workflows make use of machine-learned models to support analytical tasks, VA workflows have become increasingly important in understanding and improving Machine Learning (ML) processes. In this paper, we propose an ontology (VIS4ML) for a subarea of VA, namely “VA-assisted ML”. The purpose of VIS4ML is to describe and understand existing VA workflows used in ML as well as to detect gaps in ML processes and the potential of introducing advanced VA techniques to such processes. Ontologies have been widely used to map out the scope of a topic in biology, medicine, and many other disciplines. We adopt the scholarly methodologies for constructing VIS4ML, including the specification, conceptualization, formalization, implementation, and validation of ontologies. In particular, we reinterpret the traditional VA pipeline to encompass model-development workflows. We introduce necessary definitions, rules, syntaxes, and visual notations for formulating VIS4ML and make use of semantic web technologies for implementing it in the Web Ontology Language (OWL). VIS4ML captures the high-level knowledge about previous workflows where VA is used to assist in ML. It is consistent with the established VA concepts and will continue to evolve along with the future developments in VA and ML. While this ontology is an effort for building the theoretical foundation of VA, it can be used by practitioners in real-world applications to optimize model-development workflows by systematically examining the potential benefits that can be brought about by either machine or human capabilities. Meanwhile, VIS4ML is intended to be extensible and will continue to be updated to reflect future advancements in using VA for building high-quality data-analytical models or for building such models rapidly.

Index Terms—Visual Analytics, Visualization, Machine Learning, Human-Computer Interaction, Ontology, VIS4ML

1 INTRODUCTION

In computer science, an ontology is typically represented using a type of graph. The primary use of an ontology is for encoding the knowledge about common concepts, properties, and relations in a subject [24]. Given a key phrase “an ontology for”, Google Scholar indicates that there may be some 35,100 publications online. Its applications extend well-beyond the discipline of computer science. Ontology development is a major aspect of building a theoretical foundation of visualization [11]. However, such effort was only reported sparsely in the literature (e.g., [5, 14, 51]). While a number of conceptual workflows have been proposed for Visual Analytics (VA) (e.g., [10, 31, 57, 58]), there is not yet an ontology for describing common concepts and relations in VA. This work represents the first step towards a comprehensive ontology for VA by focusing on a subarea of VA, that is, VA-assisted Machine Learning (ML).

ML is an inspiring area of artificial intelligence. In data science in general and VA in particular, ML can play a significant role in developing machine-learned models that can be used to automate analytical tasks. In the past, the goal (A) to develop ML models is often intertwined with the goal (B) to develop such models automatically. In recent years, many started to separate these two goals in order to ensure the optimal achievement of goal (A) in ways including the use of human intelligence in the ML model development. For example, in IEEE VAST 2017, more than a dozen of papers presented VA solutions for aiding ML processes, covering a range of problems (e.g., clustering [56] or classification [42]) and ML solutions (e.g., deep learning [28] or decision trees [40]). A new subarea is emerging in VA. Partly because many are still used to the coupling of goals (A) and (B) and partly because the new VA solutions for assisting ML are drops of the ocean in comparison with automated solutions for developing automatic ML models, it is not always clear where an ML process can benefit from a VA solution.

In this work, we present an ontology VIS4ML as a “knowledge map” of this new landscape. We systematically extract goals and requirements of using visualizations within ML workflows as the raw facts for establishing this knowledge map. Our aim is not only to use the ontology to illustrate the “routes” that have been taken by recent VA solutions for ML, but also to provide VA practitioners with a means to “navigate” the complex landscape of ML in order to identify aspects that may benefit from introducing more machine or human capabilities. VIS4ML can help practitioners identify where VA has been used to assist in and improve ML workflows. The online version of VIS4ML (http://vis4ml.dbvis.de/) provides links to the previous works to aid further understanding as to why and how. We conceptualize the entities and relations in VIS4ML based on the related works in VA in general and VA assisted ML in particular (Section 2). We extract six major goals for using VA in ML from previous works (Section 3). We specify the definitions, rules, syntaxes, and visual notations for formulating this ontology (Section 4). We generalize the traditional VA pipeline to encompass model-development workflows (Section 5) and present a formalized ontology ‘VIS4ML’ for VA-assisted ML (Section 6) implemented in the Web Ontology Language (OWL). We validate VIS4ML using existing VA solutions for ML (Section 7), illustrating how VIS4ML can be used (Section 8). We offer our concluding remarks, envisaging the continuing development of VIS4ML (Section 9).

2 RELATED WORK

Our endeavor is related to Conceptual Research in VA/ML and Existing Ontologies in the Field of Visualization.

Conceptual Research in VA/ML: Conceptual workflows exist in Data Mining (DM) as well as in information visualization. Fayyad et al. [19] describe a pipeline of Knowledge Discovery in Databases (KDD) processes and Card et al. [7] describe a workflow for Information Visualization (InfoVis). VA aims at integrating both pipelines in order to combine the human and machine strengths by tightly coupling automated analysis with interactive visualization [32]. There are many other conceptual workflows proposed in the visualization literature (e.g., [10, 22, 67, 68]). Humans play a fundamental role in data intelligence processes: 1.) in InfoVis, analysts explore and manipulate visualizations to reveal patterns in the data, 2.) In DM, analysts select appropriate techniques to extract knowledge from data. 3.) In ML, model-developers determine the structures of models, select training methods, and conduct evaluation. VA research has shown that it is more
Table 1. Goals and requirements extracted from recent papers that make use of visualization to assist ML.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Kahng et al. [28]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Sacha et al. [56]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3. Tam et al. [66]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Wongsuphasawat et al. [71]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Mühlbacher et al. [42]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>6. Ren et al. [54]</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>7. Wang et al. [70]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>8. Liu et al. [39]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Rauber et al. [53]</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>10. Liu et al. [38]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>11. Ming et al. [41]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. Pezzotti et al. [49]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Kwon et al. [37]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. Ahslallah et al. [1]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Liu et al. [40]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>17. Krause et al. [35]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18. Strobelt et al. [64]</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>20. Oh et al. [44]</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>21. Strobelt et al. [63]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Frequency</td>
<td>8</td>
<td>16</td>
<td>10</td>
<td>10</td>
<td>16</td>
<td>10</td>
</tr>
</tbody>
</table>

Effective to involve human analysts in these processes in a “human-is-the-loop” approach (e.g., [15]), a sensemaking loop [50], or the human cognition model [23]). Chen and Golan [10] provide a theoretical analysis of human-machine workflows in VA using information theory.

In DM and ML, we observe a growing interest in increasing humans’ involvement. Fails and Olsen [17] describe an interactive ML approach to build and improve classifiers iteratively. Other classical scenarios are recommender systems [27] and active learning [61] approaches, where humans provide relevance information during training. Amershi et al. [2] review interactive ML systems with a tight user-coupling. They argue for the need of a common language across the diverse research areas. The ontology proposed in this work represents a progress towards such a common language.

Keim at al. [31] described the VA process as a baseline that was subsequently extended by Sacha et al. [58] to cover the knowledge generation processes. A focus at the intersection between ML and VA results in a human-centered ML process [57], which integrates interactive visualization with dimensionality reduction [59]. Complementarily, in a recent survey [16], Endert et al. reviewed a number of conceptual workflows for integrating ML into VA and called for a deeper integration of ML and VA research. Andrienko et al. [3] recently proposed to view VA as a workflow for building mental and formal models. They revisited the definitions of some common concepts, such as “data”, “analysis”, “tasks”, “structural/mental/formal models”, etc. along the VA workflow, which covers important stages of model evaluation and model development within shared human-machine workflows. Our endeavor builds on these definitions and complements their work with a more specialized focus on building and improving a type of formal models, i.e., ML models, with the help of visualizations. We achieve this through a formalized ontology of this “VIS4ML” landscape.

There are many implemented ML systems that feature VA capabilities. However, it is not clear how these workflows are related to each other. In order to build a holistic view about the role of VA in ML, it is necessary to develop a common ontology that would allow us to map out existing VA-assisted ML workflows and facilitate the identification of more opportunities and benefits in using VA in ML workflows. This is the main aim of this work. Therefore, as a starting point, we first analyze in detail the common requirements and goals of the recent approaches of using VA for ML in Section 3.

Effective to involve human analysts in these processes in a “human-is-the-loop” approach (e.g., [15]), a sensemaking loop [50], or the human cognition model [23]). Chen and Golan [10] provide a theoretical analysis of human-machine workflows in VA using information theory.

In DM and ML, we observe a growing interest in increasing humans’ involvement. Fails and Olsen [17] describe an interactive ML approach to build and improve classifiers iteratively. Other classical scenarios are recommender systems [27] and active learning [61] approaches, where humans provide relevance information during training. Amershi et al. [2] review interactive ML systems with a tight user-coupling. They argue for the need of a common language across the diverse research areas. The ontology proposed in this work represents a progress towards such a common language.

Keim at al. [31] described the VA process as a baseline that was subsequently extended by Sacha et al. [58] to cover the knowledge generation processes. A focus at the intersection between ML and VA results in a human-centered ML process [57], which integrates interactive visualization with dimensionality reduction [59]. Complementarily, in a recent survey [16], Endert et al. reviewed a number of conceptual workflows for integrating ML into VA and called for a deeper integration of ML and VA research. Andrienko et al. [3] recently proposed to view VA as a workflow for building mental and formal models. They revisited the definitions of some common concepts, such as “data”, “analysis”, “tasks”, “structural/mental/formal models”, etc. along the VA workflow, which covers important stages of model evaluation and model development within shared human-machine workflows. Our endeavor builds on these definitions and complements their work with a more specialized focus on building and improving a type of formal models, i.e., ML models, with the help of visualizations. We achieve this through a formalized ontology of this “VIS4ML” landscape.

There are many implemented ML systems that feature VA capabilities. However, it is not clear how these workflows are related to each other. In order to build a holistic view about the role of VA in ML, it is necessary to develop a common ontology that would allow us to map out existing VA-assisted ML workflows and facilitate the identification of more opportunities and benefits in using VA in ML workflows. This is the main aim of this work. Therefore, as a starting point, we first analyze in detail the common requirements and goals of the recent approaches of using VA for ML in Section 3.

Existing Ontologies in the Field of Visualization: A workshop at UK’s National e-Science Center in 2004 [5] represents the first effort to build a visualization ontology. The participants sketched a top-level visualization ontology consisting of Users, Data, Representations, and Techniques. Duke et al. [13] further discussed how a visualization ontology might be organized and realized using semantic-web technologies, and the need for turning existing conceptual terminologies and taxonomies into an ontology [14]. Shu et al. [62] provided a “prototypic” visualization for using the Web Ontology Language (OWL). Zeng et al. [65] pointed out that their ontology is still tentative and incomplete. Pérez et al. [48] modified a top-level visualization ontology [5] for representing processes and data models in visualization. Vogt and Polowski [51, 69] made an important advancement in specifying a visualization ontology, VISO, that unifies previous works, and in sharing their ontology for further refinement by the community.

In DM and ML, Cannataro and Comito [6] designed a DM ontology for grid programming (DAMON). Their ontology covers Tasks, Methods, Algorithms, and Software. Panov et al. designed an ontology (OntoDM) [45] that contains DM entities (such as data, tasks, generalizations, algorithms, components, and constraints). It has been continuously extended and aligned with other related ontologies, (e.g., in [46, 47]) and specialized (e.g., for network/graph analysis [34]). More recently, Sudathip and Sodanil [65] describe an ontology that focuses on ML concepts, such as the Learning paradigm (e.g., supervised, unsupervised, semi-supervised, reinforcement), the ML Techniques (e.g., Classification, Clustering, Regression), Evaluation (e.g., precision, accuracy, recall), and Applications (e.g., Forecasting, Diagnosis, or Screening). Our ontology is inspired by such existing ontologies and aims at filling the gap between VA and DM/ML.

Each ontology covers a domain of knowledge. Top- or high-level ontologies contain some general terms and meta-concepts that can be reused for more specific domain ontologies. For example, the OntoDM ontology by Panov et al. [47] integrates higher level ontologies, such as BOF1, BOF2, and BOF3. However, the notion of “high” and “low” is relative. There has not been widely accepted ontologies for VA and ML, which may be used to underpin this work. We thus focus on the domain VA-assisted ML, which is at a higher level than domains of individual ML problems, ML frameworks, and visualization techniques.

3 LEVERAGING VA TO ASSIST ML

We selected 21 VA-assisted ML workflows published recently in the VA literature as a starting point for a systematic requirements analysis of why and where in the ML workflow visualization is used. From these papers we identified requirements, goals, tasks, and questions that represent the motivations for using visualization (see an additional report in the supplemental materials for details). We found the improvement of the ML model as a primary motivation. A more detailed analysis revealed six major goals of using VA for ML:

G1: Examine/prepare data: 8 papers motivate the use of visualization for examining the input data, such as outlier analysis (e.g., [38, 66]), understanding datasets [64] and instance relations [40], or for partitioning the data [56]. Another motivation is the selection and validation of training data (e.g., [42]).

G2: Examine/understand ML model: A frequent (16 papers) goal is to examine the architecture/structure of ML models by providing overview visualizations (e.g., graphs) with details on demand interactions in order to recognize similarities and differences in the model structure (e.g., [53, 71]). Visualizations are specifically leveraged to analyze relationships between neurons or layers, neuron activations, filters, and...
or hidden states within complex neural networks (e.g., [28, 44, 49, 64]).

G3: Feature/parameter analysis: 10 papers provide visualizations of the feature or parameter spaces in order to understand and improve the ML model with respect to feature transformations/selections (e.g., [37, 66]) or parameter correlations (e.g., [70]).

G4: Learning process: 10 papers specifically visualize the training process to understand and assess the model learning process. An aim is, e.g., to early identify stable layers for deeper investigation (e.g., [49]) or to examine the debugging information (e.g., [39]). Some papers involve users to steer model building during the training iteration (e.g., [66]).

G5: Quality/result analysis: Many papers (16) leverage visualizations to evaluate ML results. Examples include the analysis of errors, performance, or accuracy (e.g., [35, 37, 54]). Further quantitative measures concern ML model characteristics, such as complexity or interpretability (e.g., [42]).

G6: Comparative analysis: A subset of papers particularly focus on the visual comparison of different ML model instances, including their structures, features/parameters or results (e.g., [41, 56]) or to analyze and compare sets of ML models (e.g., [36, 70]).

Table 1 summarizes these goals along a typical ML workflow. Most workflows focus on the Evaluate-Model step (G5), while there are VA supports to the Prepare-Learning steps (G2, G3) and a number of VA-enabled feedbacks within a step or across different steps iteratively. We can observe that not many papers reported about the Prepare-Data (G1) step and the visualization in the Model-Learning (G4) is typically for monitoring rather than for active-control.

Most of these goals were defined under consideration of specific ML frameworks, applications, and users. It is thus highly desirable to transform them to a set of general goals common to different ML frameworks and applications. Conceptually and methodologically, building an ontology can enable a mapping from the generalized goals to VA solutions in the context of ML. The aim of this paper is to establish an ontology that can be used to illustrate where exactly visualization is used in a specific ML workflow, allowing us to evaluate and compare existing VA approaches but also to identify novel or under-explored pathways. In the next section, we will introduce the definitions, notation elements and syntactic rules for designing such an ontology.

4 DEFINITIONS, VISUAL NOTATIONS, AND SYNTACTIC RULES

Our ontology is based on existing conceptual work on defining and characterizing VA workflows (e.g., the ones mentioned in Section 2). In this sequence of evolving definitions, this work echoes the recent work by Andrienko et al. [3], which generalizes the relation between VA workflows and structural/mental/formal models. The basic terms related to our endeavor are: entities (separated and distinguished things), data (recorded observations, instances and relationships), formal models (model in computer readable form, intended for performing calculations), knowledge (as the ultimate goal of any VA workflow, it has many facets and appears in different forms along the VA workflow), and processes transforming entities between the different stages (e.g., data processing, data mining, ML, visualization, or human cognitive processes). In the following, we will describe in detail how we build up our ontology based on these major concepts. Further descriptions of terms can be found in the glossary in the supplementary materials.

Our ontology is based on two main classes, Process and IO-Entity (Fig. 1(a) and (b)). We are interested in how each Process is dynamically related to IO-Entities such as, Data, Models, or human Knowledge. In a VA workflow, some processes may be fully automated and communicate only with other automatic processes (i.e., Automatic-Process), some may receive inputs from humans (i.e., K-Driven-Process), some may generate outputs intended for humans (i.e., K-Oriented-Process), and some may have two-way interactions with humans (i.e., K-Centred-Process). As shown in Fig. 1, the class Process is categorized into these four subclasses. For example, it is common for an action updating a neural network in each iteration of learning to be an automated process, activating an ML session be knowledge-driven, making an algorithmic prediction be knowledge-oriented, and visually exploring data be knowledge-centered.

The class IO-Entity consists of three sub-classes, Data, Model, and Knowledge. The sub-class Data encompasses any data to be analyzed, extracted features, or analytical results. It may be of a variety of data types (e.g., tabular, imagery, etc.) and includes visualization images, and commands in human-computer interaction and inter-process communication. The sub-class Models encompasses all machine-centric computational functions used or generated by VA workflows, such as processing software, machine-learned models, scientific simulation models, and decision-making algorithms. The sub-class Knowledge encompasses all human knowledge that may be available to a VA workflow as well as all human knowledge that may be gained from a VA workflow. Instances of Data and Models are stored on the computers, while instances of Knowledge are stored in humans’ mind.

The ontological relations (or properties as referred to in OWL) between Processes and IO-Entities are conceptually similar to “transform from/bi/into” defined in [62]. In our ontology, we explicitly represent these relations separately as connections has-Process-Successor from IO-Entities to Processes and connections has-IO-Entity-Successor from Processes to IO-Entities. In other words, these two types of directed connections explicitly define
The predecessor-successor and action-actor relationships within any workflow. In addition, there are hierarchical class relations, such as is-SubClass-Of linking sub-classes to their parent class. In ontology modeling, it is common to represent a section of the ontology as a macro-block. The relation is-BreakDown-Of links such a sub-ontology to the macro. The relation is-Decomposition-of is a generic relation encompassing all types of relations from details to a summary abstraction. Examples of using different types of relations will be shown in Sections 5-7 in conjunction with our discussions on the ontology for VA-assisted ML. It is necessary to define a set of syntactic rules for the aforementioned relations. These include:

1: A has-Process-Successor relation can only be used to connect from an IO-Entity to a Process. A has-IO-Entity-Successor relation can only connect from a Process to an IO-Entity.

2: In a full ontological representation, a Process can directly connect only with an IO-Entity or its sub-classes, and an IO-Entity can directly connect only with a Process or its sub-classes.

3: In a sub-ontology representing the decomposition of a Process, all incoming connections from outside the boundary can only be has-Process-Successor relations. All outgoing connections from inside the boundary can only be has-IO-Entity-Successor relations. Similarly, in a sub-ontology representing the decomposition of an IO-Entity, all incoming connections can only be has-IO-Entity-Successor relations. All outgoing connections can only be has-Process-Successor.

When one follows the above syntactic rules strictly, there will inevitably be many connections in the graph representation of any slightly complex workflow. For example, any Process other than Automated-Process would have connections to IO-Entities representing human knowledge. To reduce the visual clutter, we allow the omission of such connections by simply assuming that we can infer such connections for any K-*Process. In addition, we introduce several compact visual notations as illustrated in Fig. 1. An IO-Entity can be miniaturized, resulting in a connection between two Processes via an implicitly-defined IO-Entity. Similarly, a Process can be miniaturized with a connection between two IO-Entities via an implicitly-defined Process.

As part of the implementation, we have used a standard OWL editor, Protégé [43], to specify all ontological representations in this work and used OntoGraf [18] to visualize the recorded representations. A few examples are included in the supplementary materials. However, OntoGraf cannot visually accommodate new icons and compact notations as shown in Fig. 1. Therefore all visual illustrations in this paper were redrawn to enable the richer visual notations introduced above.

5 VISUAL ANALYTICS DIAMOND

As reviewed by Chen and Golan [10], in the field of visualization, many schematic representations of workflows have been proposed. We use the diamond-shaped workflow proposed by Keim et al. [31] as the baseline representation. We chose this workflow as a baseline because it is used as a foundation in many existing conceptual and methodological research papers and has proven to be generic (i.e., it can be used to embed and relate to other existing workflows). In this section, we first generalize this representation using the ontological notation given in the previous section. We then demonstrate its generality by transforming a number of visualization and VA workflows in the literature into diamond-shaped representations.

5.1 Ontological Representation and Generalization

Fig. 2(a) shows the workflow proposed by Keim et al. [31], and it features two interacting parallel components for DM models and visualization respectively. Using the ontological notation in Section 4, we can represent Data and Knowledge as IO-Entities and Visual Data Exploration (VDE) and Data Mining as Processes. As illustrated in Fig. 2(b), we have introduced extra components, which were implicitly assumed in [31], in order to adhere to the notational rule about connections between IO-Entities and Processes. Note that the self-loop associated with visualization in the original workflow is now represented by the path labeled as 8-9-3-4 and 5-6-7-9-3-4. Using the compact notation in Section 4, we can depict the ontology of Fig. 2(b) as Fig. 2(c), which is more or less the same as Fig. 2(a).

The essence of the original representation [31] is its emphasis on the need for both human- and machine-centric processes in VA and the need for various interactions that transform information between processes. Furthermore this essence, we purposely denote any data intelligence workflow or its processing components with two generic Processes, one for a human-centric process and one for a machine-centric process. This diamond-shaped layout, which is illustrated in Fig. 3(a), is a further generalization of Fig. 2(c). Here the conversion of drawing is to place any K-Centered process at the upper part of the diamond (or on the right if we depict the primary progression of a workflow from top to
Fig. 4. A high-level overview of the overall VA ontology, including a coarse sub-ontology for visual data exploration and a pointer to a detailed sub-ontology for VA-assisted ML (VIS4ML).

Figures 5 and 6

5.2 Overview of the VA Ontology

We can leverage the ontological elements, rules, and notions in Section 4 and the convention for ontological representation and generalization in Section 5.1 to construct a VA ontology. A comprehensive VA ontology will need to map out many aspects of VA (e.g., statistical inference, DM techniques, visualization, interaction, human cognition, and so on) as well as to cover a broad range of workflows in different applications (e.g., financial data, social media, and so on). While it may take some time and a collective effort of the VA community to create such an ontology, here we sketch out an overview of such an ontology in Fig. 4 and we will detail a sub-ontology VIS4ML in Section 6.

The typical visualization workflows (e.g., the visualization loop in [68] and the two workflows $W_1$ (dissemination) and $W_2$ (observation) in [10]) can be represented by the path 1-7 or the two feedback loops 1-10 and 7-9 in Fig. 2(b). The central process, VDE, can be decomposed into a sub-ontology as illustrated in the upper part of Fig. 4. Although VDE is considered as a human-centric process, we can observe that there are many machine-centric sub-processes within the sub-ontology. Most of these machine-centric sub-processes are K-Driven processes as they are controlled by the users but do not deliver output directly to the users except Display, which is a K-Oriented as a typical display device is almost always automated when it refreshes the screen. However, the involvement of the users in controlling visual mapping and other K-Driven processes, and more importantly, in navigation and direct manipulation warrants the whole process VDE as a K-Centered process.

5.3 VIS4ML – An Ontology for VA-assisted ML

In this section, we focus on one sub-ontology (VIS4ML) of the general VA ontology (Fig. 4). We expand the sub-class VA-assisted-ML on the center-left part of Fig. 4 and examine how steps in machine learn-
ing (ML) workflows can be assisted by VDE. VIS4ML drills-down hierarchically from the general VA ontology and directly relates to the general goals (G1-G6) of using VA to assist ML.

The primary objective of ML is to create an algorithmic model that can be used to perform an analytical task automatically. These tasks may include classification, clustering, and so on. There are also a large number of technical frameworks (e.g., neural networks and decision trees), each of which underpins a type of algorithmic model and determines how they are specified and trained. Despite the diversity among these frameworks and their many variations, a ML process is typically composed of four major processes (Fig. 5) that facilitate the pre-processing of the data, the preparation for the learning process, the model learning itself, and the evaluation of the learned model. In the following discussion, we sequentially walk through these major steps and provide more details about their decompositions. Note that we position machine-centred processes on the left and human-centred processes on the right. This duality illustrates that visualization can be used in any human-centred stage to assist ML at all the major steps.

**Prepare-Data**: Data preparation or pre-processing is a common step for many data intelligence workflows, such as data mining (DM), ML, and visualization. The Data IO-Entity is input and output of this process and is also used by all subsequent ML steps. Data has many sub-classes (we added some examples, such as Tabular, Network, and so on). In an ontological representation, normally, one would depict only one Data IO-Entity and draw all input and output connections with the four processes. In order to illustrate the ordering of the four steps clearly and avoid the clustering of the connections, we allow IO-Entities to be duplicated. We use a superscript (e.g., G2') to indicate that an IO-Entity is duplicated in the visual representation.

As shown in Fig. 5 (G1), the **Prepare-Data** process can be further decomposed to include typical machine-centric processes, such as statistical profiling, automatic detection of errors or missing values, or data transformation. Because the information and knowledge required in data cleaning and transformation is typically not in the data and automatic error detection often is not reliable, it is necessary for such machine-centric processes to be controlled and monitored by an analyst.

In practice, it is more common for the data to be visually inspected and manually cleaned and annotated by analysts, with or without using VA tools. For example, an analyst may observe the data in a tabular representation, correct errors, fill in missing data, and annotate individual data records for supervised learning (G1). Such manual processes are highly labor-intensive. A prominent visual-assisted data preparation example is the data wrangler system [29], which allows the analyst to explore the data with the aid of visualization in order to discover errors and missing values more efficiently and effectively while controlling and monitoring the applications of various operations for data cleaning and transformation. Note that the processes in this step are mostly
independent of any specific ML framework. The pre-processing step can also benefit other aspects of VA, such as DM and VDE.

**Prepare-Learning:** This process is primarily for making preparation for the learning process to be performed in the succeeding process. The main focus is to deliver an initialized model. In Fig. 5, the Model IO-Entity is also visually duplicated as indicated by *. It has also many sub-classes (e.g., Framework, Template, Parameter, etc.). This step, which is often described as “model building”, involves many human design decisions and therefore contains many human-centric processes as shown in Fig. 5 (top-left).

For many types of raw data (e.g., videos and documents), some ML techniques have difficulties to handle such raw data directly. One common approach is to extract feature data from the raw data, for example, in the study of Tam et al. [66], several hundreds of different candidate features were defined for imagery data and algorithms were implemented for extracting these features. This enables the corresponding ML processes to learn a model based fully or partly on the feature data, which is commonly referred to as “a bag of words or features”. At this stage, the model-developer does not know exactly which candidate features are useful and which are not. The selection of these features relies on the model-developer’s knowledge about possible specifi- cations of features, possible algorithms for extracting these features, and their availability. Note that some feature extraction algorithms may also be constructed using ML, but most feature extraction algorithms contain mathematical formulations or algorithmic structures that are not machine-learned. It is also common for ML workflows to partition data into Training-Data and Testing-Data. As this operation is usually specifically for ML, we include the Partition-Data process in the decomposition of Prepare-Learning.

One of the most important decisions is to select the ML Framework determining which principle approach (e.g., supervised or unsupervised), which structure and constructs are used to define a model (e.g., convolutional neural networks, decision trees, etc.), which learning technique (e.g., k-means, hierarchical clustering, Hunter’s algorithm, random forest, etc.), and so on. In practice, quite often, the ML Framework was determined before the step Prepare-Data or even before the raw data was captured. In principle, decisions on some details of a framework are usually evolved during a ML workflow.

Some ML techniques, such as many types of neural networks, require the specification of a Template model (i.e., an empty structure). Some ML techniques need to define one or more Metrics such as cost functions, stress or performance measures, and update rules. Many of these metrics are parameterized. For example, distance metrics, which are fundamental to many ML techniques, such as dimensionality reduction or clustering, may contain weights that can be also modified. Most ML techniques have Parameters for controlling the learning process, (e.g., the number of training iterations) or constraining a model (e.g., the depth of a decision tree). Many of these parameters cannot be chosen in a purely automatic fashion and have to be manually tuned. Some ML technique may require an Initialize-Model process to create an initialized Partial-Model that may steer the learning process towards a certain part of the model space. Different initialization strategies exists, e.g., a machine-centric randomized-initialization or more human-centric techniques (e.g., that allow an analyst to sketch initial neuron-prototypes for a self-organizing map [60]). Prepare-Learning is typically an iterative design process. Hence, there are a number of feedback loops that connect previous processes inside the decomposition block or the previous steps outside. At the end of this step, it generates two collections of IO-Entities, Data and Model, to be passed to the subsequent Model-Learning step. We can relate G2 (understand ML model) and G3 (examining features/parameterizations) to this Prepare-Learning step. Note, that we can also relate G1 (examining data) to the Partition-Data process.

**Model-Learning:** The initialized model is then trained by an algorithmic process, which is usually performed in a fully-automatic fashion. As illustrated in Fig. 5 (G4), the training is usually performed iteratively by two machine-centric processes, Compute-Fitness and Update-Model. The number of iterations vary from a few to millions. At this step, VA can be used to examine the learning process (G4). More commonly, visualization of intermediate results and learning process is provided to enable the model-developer to Monitor-Learning. Some semi-automatic approaches exist, such as active learning, which allow the model-developer to control some aspects of the training dynamically. In addition to the basic pausing/resuming the training process, some ML workflows allow for human-centric decisions for data selection, outlier removal, parameter change, and so on. Our ontology accommodates such approaches through the Active-Control process. The resulting full Model is passed to the Evaluate-Model step together with a collection of Data, which may include monitoring data.

**Evaluate-Model:** In this step, the trained Model performs the actual task (e.g., applying a classifier to unseen data), while the model-developer interprets and assesses the results (G5). As illustrated in Fig. 5 (G5), a machine-centric process Model-testing applies the Model to Testing-Data and computes quality metrics, such as precision-recall or accuracy. Similar to the Active-Control in the previous step, this testing process can also be controlled interactively. With complex ML models it is a major difficulty for model-developers to know what is going on. In recent years, more visualization processes have been introduced to help model-developers in understanding a model, its behaviors, and the learning process at different levels of detail. We accommodate these processes as part of Understand-Model, which can be used in conjunction with Active-Test-Control and Quality-Analysis to examine the model’s behaviors under certain conditions and visualize the results (e.g., a confusion matrix). Naturally following the Evaluate-Model process, the model-developer may wish to make some changes to what was set in the previous steps, such as Prepare-Data and Prepare-Learning. Hence, a number of feedback loops originate from the Evaluate-Model process. The resulting Model is the final output of the ML process and can be used in other VA processes as shown in Fig. 4.

Multiple iterations of the feedback loops in Fig. 5 result in different ML models and training provenance. A further step is needed to support analysis of different ML models, comparing their frameworks or templates, choices of features or parameters, performances of individual models and their ensemble. We therefore added G6 (comparative analysis) at the end of the ML workflow.

7 **EXAMPLE WORKFLOWS AS ONTOLOGY PATHWAYS**

Usually, there are many ways to represent a complex workflow schematically, especially if it has feedback loops, which most VA workflows would have. As our ontology is intended to encompass most, if not all, VA workflows for ML, it is inevitable that the ontology may have different schematic representations that are functionally equivalent. So we do anticipate that some colleagues in the communities of visualization and ML could have or prefer alternative ways to structure the ontology. We therefore place the emphasis of validation on its descriptive power, that is, can it describe all VA-assisted ML workflows reported in the literature. One objective of studying the papers listed in Table 1 is to validate the ontology proposed in Section 6 by ensuring that each workflow can be comfortably mapped onto a pathway in the ontology. As shown in Fig. 6, four example pathways are superimposed onto the ontology. During the process of validation, we identified a number of missing sections in some pathways, enabling us to revise and improve the ontology. Below we described four of such pathways briefly.

An Example of Deep Learning – ActiVis: Kahng et al. [28] present a system for assisting in learning large-scale deep learning models. It enables the visual exploration of neuron activations in different classification cases (subsets and instances). We extracted a common workflow from the use cases of ActiVis in the paper, such as understanding activation patterns (Evaluate-Model) and revising the CNN based on observations (Prepare-Learning). This workflow is described by the blue pathway No. 1 in Fig. 6: An initial model is prepared (Create-Template, Define-Parameter) and explored after the training has finished. Then, the analyst selects particular nodes within the architecture graph of the neural network to inspect its neuron activations and how the model performs on different test cases (subsets) and particu-
An Example of Clustering – SOMFlow: Sacha et al. [56] propose a workflow for the interactive cluster analysis of time series. Their system (SOMFlow) utilizes a self-organizing map algorithm to cluster time series and to project them in a compact Kohonen Map. Subsequently, a SOM template can be generated from the data. The analyst can go back to the Prepare-Data step and improve the Template or the training Parameters.

An Example of Decision Tree Construction: Tam et al. [66] juxtaposed a VA workflow for constructing decision trees with an automated ML workflow. Both of their workflows are shown as the green pathway in Fig 6. The main difference between the two workflows is the automated workflow at this stage, which is marked as 3a, is a series of automated iterations facilitated by either C4.5 [52] or CART [4]. The VA-assisted workflow marked as 3b, allows a model-developer to visualize the distribution of training data against each feature specified in the Prepare-Learning step. While being guided by the fitness values computed for all features in the same way as in the automated workflow, the model-developer can exercise his/her judgment (Active-Control in Model-Learning) about the most suitable feature for the current iteration using additional knowledge unavailable in the data. E.g., knowledge about the reliability of various feature extraction algorithms, the observed anomalies in the raw imagery data, and so on). In addition, the model-developer can determine
the best participation of a selected feature axis, for which the automated system still has some difficulties in making an optimal decision.

Another Example of Deep Learning – TensorFlow: Wongsuphasawat et al. [71] describe the TensorFlow Graph Visualizer as part of the TensorFlow ML platform for building neural network models. Their task analysis showed that this VA facility helped model-developers perform a variety of visualization tasks, such as gaining an overview of the high-level structure of a model in the steps Prepare-Learning and Evaluate-Model, observing similarities and differences between components in a model and identifying potential bugs, and so on. Their evaluation scenarios confirm that the TensorFlow Graph Visualizer enabled model-developers to improve their model templates in a more effective design process at the Prepare-Learning step, and was critical in assisting the model-developers to gain a good understanding of the learned model and the learning process at the Evaluate-Model step. Their pathway (No. 4) is drawn in purple in Fig. 6.

These four examples illustrate that VIS4ML is capable of illustrating particular ML workflows as pathways that emphasize why and where visualization can be used to assist ML steps.

8 USING VIS4ML

A Past Scenario of Practical Usage. A PhD student (Martin, not a co-author) designed a combined neural network (CNN + RNN) for multi-line offline handwriting recognition. However, his initial error rate was well under expectation and he therefore started to make use of visualizations. As a first step, we drew his workflow as a pathway in the VIS4ML ontology (it can be found in the supplemental materials). By following the pathway systematically, we were able to identify and discuss the critical aspects that might benefit from humans’ involvement and formulated various visualization solutions by learning from previous works linked to each of the bus stops on the pathway.

He then developed a novel workflow that leverages visualizations for his critical stops. For example, in the Monitor-Learning step, he captured model snapshots for different data samples (e.g., misclassified characters). As part of Model-Understanding, he visualized these snapshots using a purposely-designed heatmap in conjunction with computed quality measures. He was able to identify the problems and postulated solutions. With new insight, he went back to the Prepare-Data step (e.g., adjust the image scaling factor) and to the Prepare-Learning steps (e.g., revise the model template or re-tune parameters).

Encouraged by the usefulness of his “bus stops”, we started to explore other pathways to identify new critical points along his pathway and ideas for new bus stops. He planned to try neuron activation visualizations and add a steering functionality for continuing the Model-Learning dynamically after a stop for snapshot visualization and parameter modification. The VIS4ML ontology has guided the PhD student to develop a more effective ML workflow for creating better ML models for handwriting recognition.

A Broader Scenario of Practical Usage. The ontology outlined in this work is supported by a web-based platform (http://vis4ml.dbvis.de), which can help researchers and developers answer the question of where by identifying aspects of their ML workflows that may benefit from a visual analytics approach, and the questions of why and how by reading the previous works linked to the ontology. In this way, more people can benefit from the knowledge captured in VIS4ML.

A Scenario of Theoretical Research. The VIS4ML ontology is based on the established conceptual workflow for VA by Keim et al. [31] and supports and complements a number of new works on model development, e.g., Andrienko et al. [3], Endert et al. [16], Choo and Liu [12], and Hohman et al. [25]. As ontology development is a major component of the theoretical foundation of visualization [11], it will be highly valuable for researchers to extend the VIS4ML ontology, e.g., by detailing lower-level ontologies for individual ML problems (e.g., classification, prediction, etc.), individual ML frameworks (e.g., Bayesian network and genetic algorithm, etc.), individual visualization techniques (e.g., data flow graphs and confusion matrices), and so on. Similarly, there is a need for ontologies at higher levels (e.g., VA and VIS) and in neighboring domains (e.g., InfoVis and SciVis).

Meanwhile VIS4ML will continue to evolve in response to new advances in VA and ML. In the future, there will be opportunities to develop a detailed ontology for the process inside the Knowledge entity. We have made the source representation of VIS4ML available at https://gitlab.dbvis.de/sacha/VIS4ML to facilitate future ontological research effort by the community.

Scenarios of Technology Development. As a knowledge representation, ontologies can be used to support many technical developments [8, 21, 26, 30, 33]. For example, the terms and connections in an ontology can be used to support document and corpus analysis, such as text searching, labeling, and clustering. The relationships among pathways, critical points, and commonly-used VA techniques can be used to enable automated visualization generation or technique recommendation. The increasing complexity of ontologies and scenarios of their usages will demand for more VA techniques to support the visual and analytical exploration of ontologies.

9 DISCUSSION AND CONCLUSION

Based on our detailed study of existing ML workflows and our construction of VIS4ML, we are able to derive the following observations. The use of VA for the phase of Prepare-Data is least reported in general. The typical activities in this phase, such as initial data analysis, data cleansing and annotation can benefit from VA significantly. Hence, this is an area that VA can potentially have a huge role to play. The phase of Prepare-Learning is a highly human-centered process. Traditionally, this is the phase where researchers in ML exercise their knowledge about ML feature extraction algorithms, creativity in developing new algorithms, and intuitions in constructing model templates and setting parameters. Visualizations have now started to play a significant role in helping this phase as illustrated in our examples. We will witness more advances in the coming years. The phase Model-Learning is highly automated for some frameworks (e.g., CNN, RNN, etc.). These ML workflows can benefit from VA-assisted monitoring of the learning processes. Some models can be trained and tested rapidly, for which one can monitor and control the entire process. In contrast, more complex models require a longer and more expensive training step and often cannot be visualized dynamically. To deal with such challenges, novel VA approaches for steering ML are emerging [20]. The data and visualization captured in monitoring can support the effort to understand the learned model, learning processes, and the deficiencies identified. The phase Evaluate-Model usually features a huge amount of data, including the original data, ground truth annotation, feature data, monitoring data, testing results, multiple models for comparison, and so on. It is a natural playground of VA.

Although ML is commonly considered as an AI technique and is meant to be automated, VIS4ML allows us to view ML workflows holistically. Among the four main phases, three are largely human-centered processes. Some latest research discussed in the paper suggested that VA can also provide direct support to the step of Model Learning.

This paper describes the construction of an ontology for VA, focusing on the formal framework of the ontology (e.g., definition, rules, notations), the generalization and extension of the top-level VA ontology, the convention of diamond-shaped layout, and in particular the sub-ontology for VA-assisted ML. While there are many concepts and ontologies in VA, this ontology can be a starting point. Similar to the classification and clustering problems in VA and ML, it was not always trivial to assign a Process or IO-Entity to a particular step. For example, feature extraction or transformation could appear in either Prepare-Data and Prepare-Learning. In some cases, we duplicated IO-Entities such as Data and Model visually. In other cases, we assigned some processes to specific phases.

With the support by the VIS4ML web site (http://vis4ml.dbvis.de), and the open source of the VIS4ML ontology (https://gitlab.dbvis.de/sacha/VIS4ML), we call for active participation in the validation and extension of the ontology.

ACKNOWLEDGMENTS

We wish to thank Martin Schall for his feedback and also thank the dbvis support for hosting/deploying the websites.
REFERENCES


