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

- *Dominik Sacha, Matthias Kraus and Daniel Keim are with University of Konstanz. E-mail: lastname@dbvis.inf.uni-konstanz.de*
- *Min Chen is with University of Oxford. E-mail: min.chen@oerc.ox.ac.uk*

Manuscript received 31 Mar. 2018; accepted 1 Aug. 2018.
Date of publication 16 Aug. 2018; date of current version 21 Oct. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2864838

Table 1. Goals and requirements extracted from recent papers that make use of visualization to assist ML.

	G1	G2	G3	G4	G5	G6
1. Kahng et al. [28]		X			X	
2. Sacha et al. [56]	X	X	X	X	X	X
3. Tam et al. [66]		X	X	X		
4. Wongsuphasawat et al. [71]		X		X	X	
5. Mühlbacher et al. [42]	X	X	X		X	X
6. Ren et al. [54]	X				X	X
7. Wang et al. [70]			X		X	X
8. Liu et al. [39]		X	X	X		
9. Rauber et al. [53]		X		X	X	
10. Liu et al. [38]	X	X		X		
11. Ming et al. [41]		X			X	X
12. Pezzotti et al. [49]		X		X	X	
13. Kumpf et al. [36]					X	X
14. Kwon et al. [37]			X		X	X
15. Alsallakh et al. [1]	X	X		X	X	
16. Liu et al. [40]	X	X	X	X	X	X
17. Krause et al. [35]	X	X	X		X	
18. Strobelt et al. [64]	X	X			X	X
19. Cashman et al. [9]			X	X		
20. Olah et al. [44]		X	X			
21. Strobelt et al. [63]		X			X	X
Frequency	8	16	10	10	16	10
ML Stage	**Data**	**Prep.**	**Prep.**	**Learn**	**Eval.**	**Comp.**

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 process. 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 moels. 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 up 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 "prototype" ontology for visualization using the Web Ontology Language (OWL) and Protégé, pointing 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. Voigt and Polowinski [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 ontolgies, such as BFO[1], OBO[2], and OBI[3]. 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,

[1] `http://basic-formal-ontology.org/`, accessed 18.03.18

[2] `http://obofoundry.org/ontology/ro.html`, accessed 18.03.18

[3] `http://obi-ontology.org/`, accessed 18.03.18

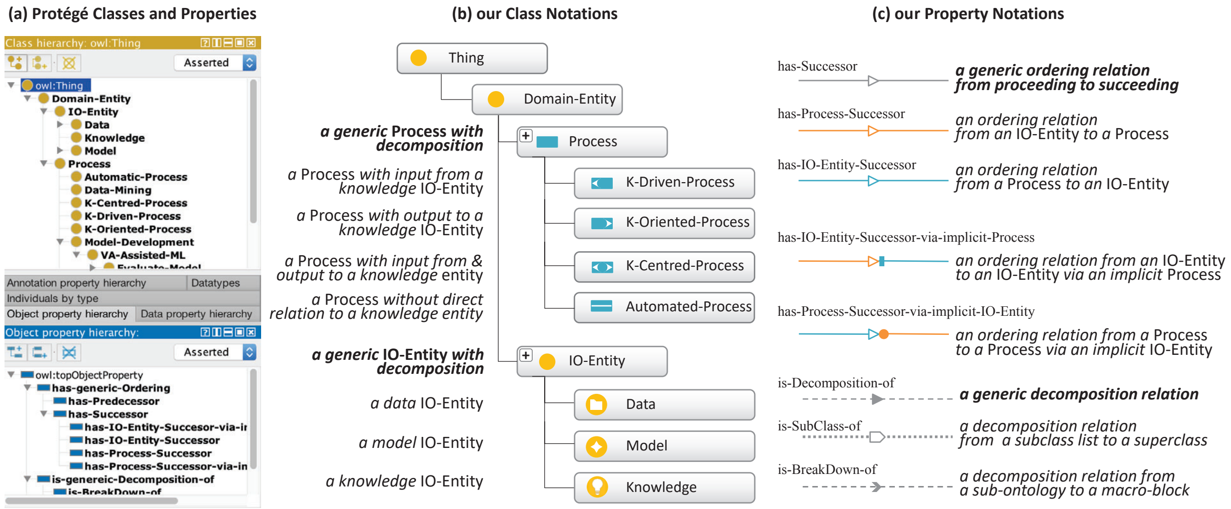
(a) Protégé Classes and Properties
Class hierarchy: owl:Thing
Asserted
owl:Thing
Domain-Entity
IO-Entity
Data
Knowledge
Model
Process
Automatic-Process
Data-Mining
K-Centred-Process
K-Driven-Process
K-Oriented-Process
Model-Development
VA-Assisted-ML
Evaluate-Model
Annotation property hierarchy
Datatypes
Individuals by type
Object property hierarchy
Data property hierarchy
Object property hierarchy:
Asserted
owl:topObjectProperty
has-generic-Ordering
has-Predecessor
has-Successor
has-IO-Entity-Succesor-via-ir
has-IO-Entity-Successor
has-Process-Successor
has-Process-Successor-via-in
is-genereic-Decomposition-of
is-BreakDown-of
(b) our Class Notations
Thing
Domain-Entity
a generic Process with decomposition
Process
a Process with input from a knowledge IO-Entity
K-Driven-Process
a Process with output to a knowledge IO-Entity
K-Oriented-Process
a Process with input from & output to a knowledge entity
K-Centred-Process
a Process without direct relation to a knowledge entity
Automated-Process
a generic IO-Entity with decomposition
IO-Entity
a data IO-Entity
Data
a model IO-Entity
Model
a knowledge IO-Entity
Knowledge
(c) our Property Notations
has-Successor
a generic ordering relation from proceeding to succeeding
has-Process-Successor
an ordering relation from an IO-Entity to a Process
has-IO-Entity-Successor
an ordering relation from a Process to an IO-Entity
has-IO-Entity-Successor-via-implicit-Process
an ordering relation from an IO-Entity to an IO-Entity via an implicit Process
has-Process-Successor-via-implicit-IO-Entity
an ordering relation from a Process to a Process via an implicit IO-Entity
is-Decomposition-of
a generic decomposition relation
is-SubClass-of
a decomposition relation from a subclass list to a superclass
is-BreakDown-of
a decomposition relation from a sub-ontology to a macro-block

Fig. 1. The main components of a VA ontology: The two major subclasses of the class hierarchy, namely Processes and IO-Entities, are shown in a Protégé view (a) and in our extended visual notations (b). Classes can be connected to each other using several types of relations (i.e., properties in OWL) shown in their visual notations (c).

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 interpretabilty (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 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 supplemental 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/by/into" defined in [62] (Fig. 1 (c)). 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

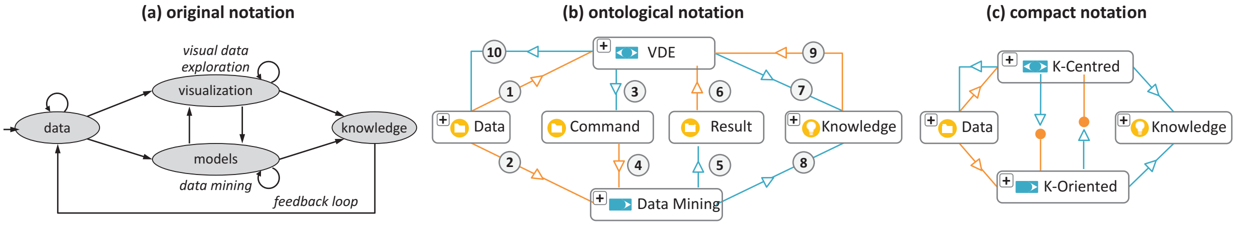

Fig. 2. The baseline schematic representation by Keim et al. [31] in (a) is redrawn in (b) using the ontological notation (we introduced IO-Entities and Processes according to our syntactic rules). The baseline diamond representation is further simplified in (c) using the compact notation.

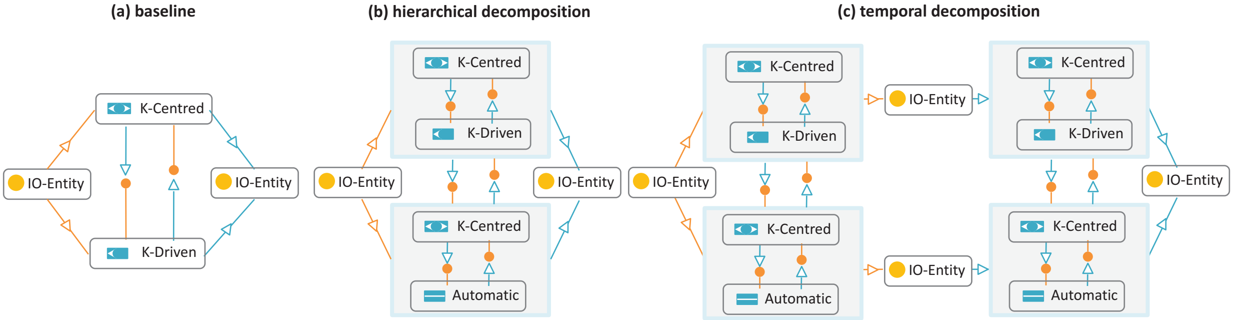

Fig. 3. A diamond-shaped baseline (a) layout for depicting any data intelligence workflow or its processing component. Its hierarchical decomposition (b) features a diamond-shaped substructure within each process, and its temporal decomposition (c) features concatenated diamond-shaped substructures. In (a), (b), (c), a K-Driven process can be replaced with a K-Oriented or Automatic process, and vice versa.

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 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 7-9, that with data is now represented by the path 1-10, and that associated with models is now represented by the paths 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. Furthering 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 convention 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

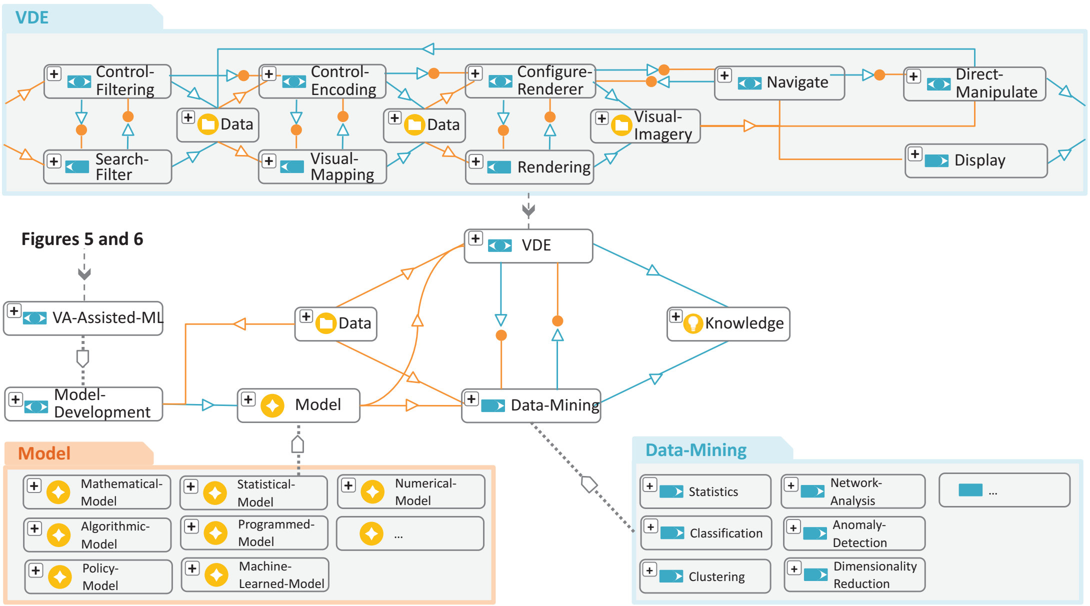

Fig. 4. A high-level overview of the overall VA ontology, including a coarse sub-ontology for visual data exploration and a pointer to a detailed sub-ontology for *VA assisted ML* (VIS4ML).

bottom), and any Automatic, K-Driven, or K-Oriented process at the lower part of the diamond (or on the left in a top-to-bottom workflow). The layout does not in any way imply that all workflows must have two parallel processes. This diamond-shaped layout convention serves two purposes. (i) They visually distinguish processes that involve more human decisions from those which involve more automated statistical or algorithmic decisions. (ii) They encourage the designers of VA workflows to think about the means to support human decisions with statistics and algorithms as well as the means to augment machine decisions with human intelligence.

In fact, neither placeholder is restricted to a process performed solely by humans or machines. A K-Centered process can be decomposed into a sub-ontology where humans play a more significant role than machines, e.g., a K-Centered process in parallel with a K-Driven. Similarly, a K-Oriented process can involve humans in its sub-processes. Fig. 3(b) shows the hierarchical decomposition of the workflow in Fig. 3(a), while Fig. 3(c) shows the temporal decomposition of Fig. 3(b). Because any feedback loop can be temporally sequentialized using a series of concatenated baseline representations, the feedback loop in Fig. 2 is a compact depiction of such sequentialized workflow, but not an essential component of the baseline representation in Fig. 3(a).

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 feedfack 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. We can observe that within the sub-ontology for VDE, there are several diamond-shaped configurations, suggesting that VDE can benefit from advanced machine-centric processes such as statistical analysis and processing and rendering algorithms while empowering users to have interactive controls, through which their knowledge about the contexts of the data and tasks can be part of the input of the process.

Although most readers of this paper appreciate that VDE can take place at any stage of a data intelligence workflow, there is still a widespread misconception that visualization is just for disseminating the results of automated DM. Nevertheless, such a workflow (i.e., W_3 in [10]) can be represented by the path 2-5-6-7 in Fig. 2(b). The workflow of Keim et al. [31] as illustrated in Fig. 2 highlights the necessity for integrating VDE with DM throughout a data intelligence workflow.

There are a variety of DM tasks and each may utilize different techniques. The lower-right part of Fig. 4 shows a list of sub-classes of DM. An initial attempt has been made to devise a sub-ontology for DM [55] and further investigation is required to map out this large collection of tasks, techniques and application workflows. The central to any DM process is one or more analytical Models as an input IO-Entity to DM. Such a model can be developed in many ways. The lower-left part of Fig. 4 shows a list of sub-classes of Model-Development. It is a vast area where VA has been deployed to assist the model developers (e.g., in software visualization) and can potentially have a more significant role to play. Chen and Golan outlined two workflows, W_5 and W_6, for VA-assisted Model-Development [10], suggesting that this process can be decomposed into two parallel but inter-related processes in a way similar to that VA is decomposed into VDE and DM.

One important method of Model-Development is *VA-assisted ML*, which is represented by the sub-class VA-assisted-ML. With the generalization in Fig. 3(a), we can maintain Data and VDE in the VA ontology, and replace Knowledge with Model and DM with machine-centric processes designed for model development (representing typical ML workflows). In the next section, we will show that the VA-assisted-ML sub-ontology does indeed contain many diamond-shaped stages.

6 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-

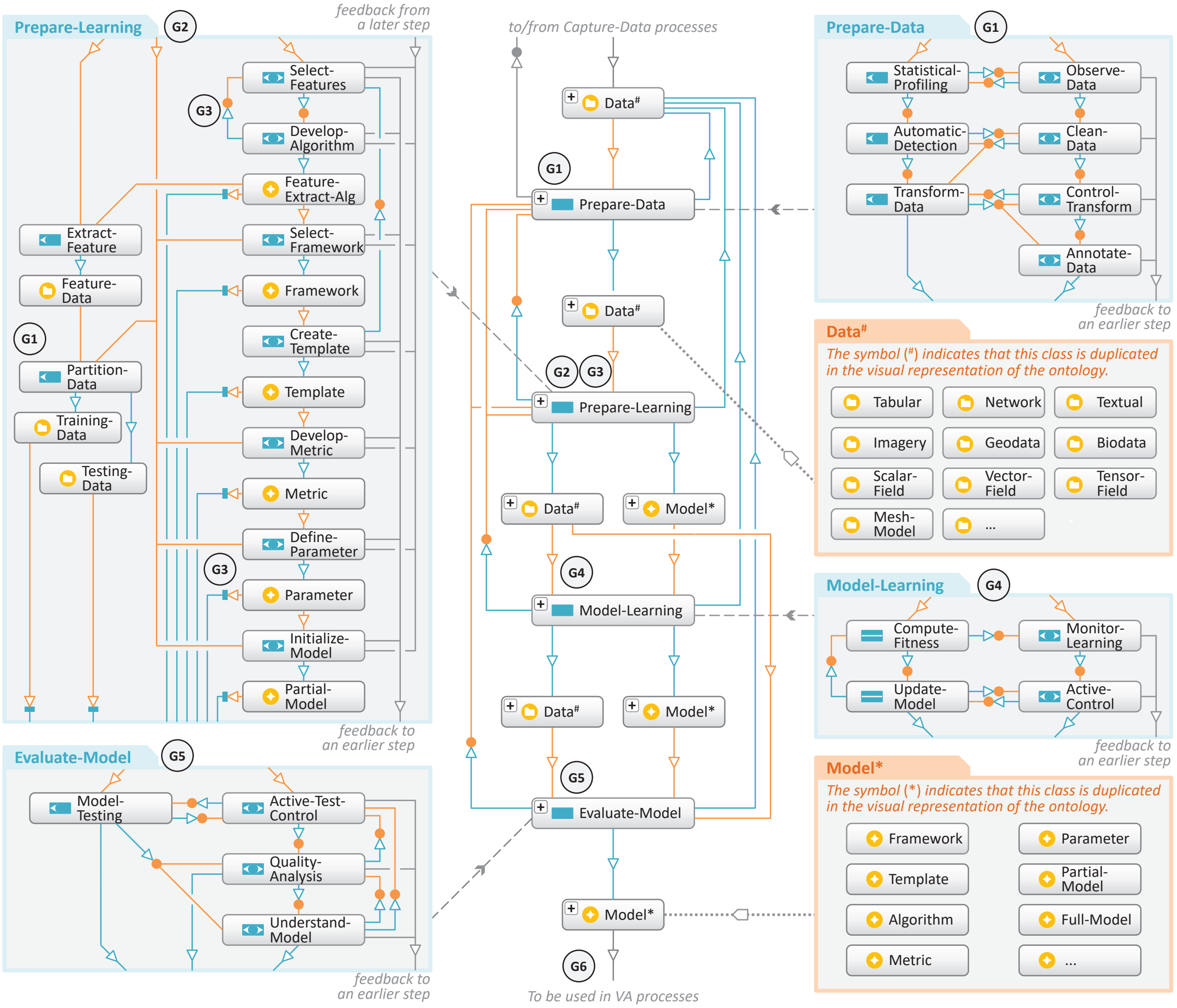

Fig. 5. VIS4ML – An ontology for *VA-assisted ML* as part of the overall VA ontology. The ML workflow is represented with four major steps of Prepare-Data, Prepare-Learning, Model-Learning, and Evaluate-Model. We also mapped six major goals for using VA to assist ML to these steps: G1 - Examine/prepare data, G2 - Examine/understand ML model, G3 - Feature/parameter analysis, G4 - Learning process, G5 - Quality/result analysis, G6 - Comparative analysis.

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 trough 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 cluttering of the connections, we allow IO-Entities to be duplicated. We use a superscript (e.g., $^{\#}$ and *) 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 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 specifications 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-

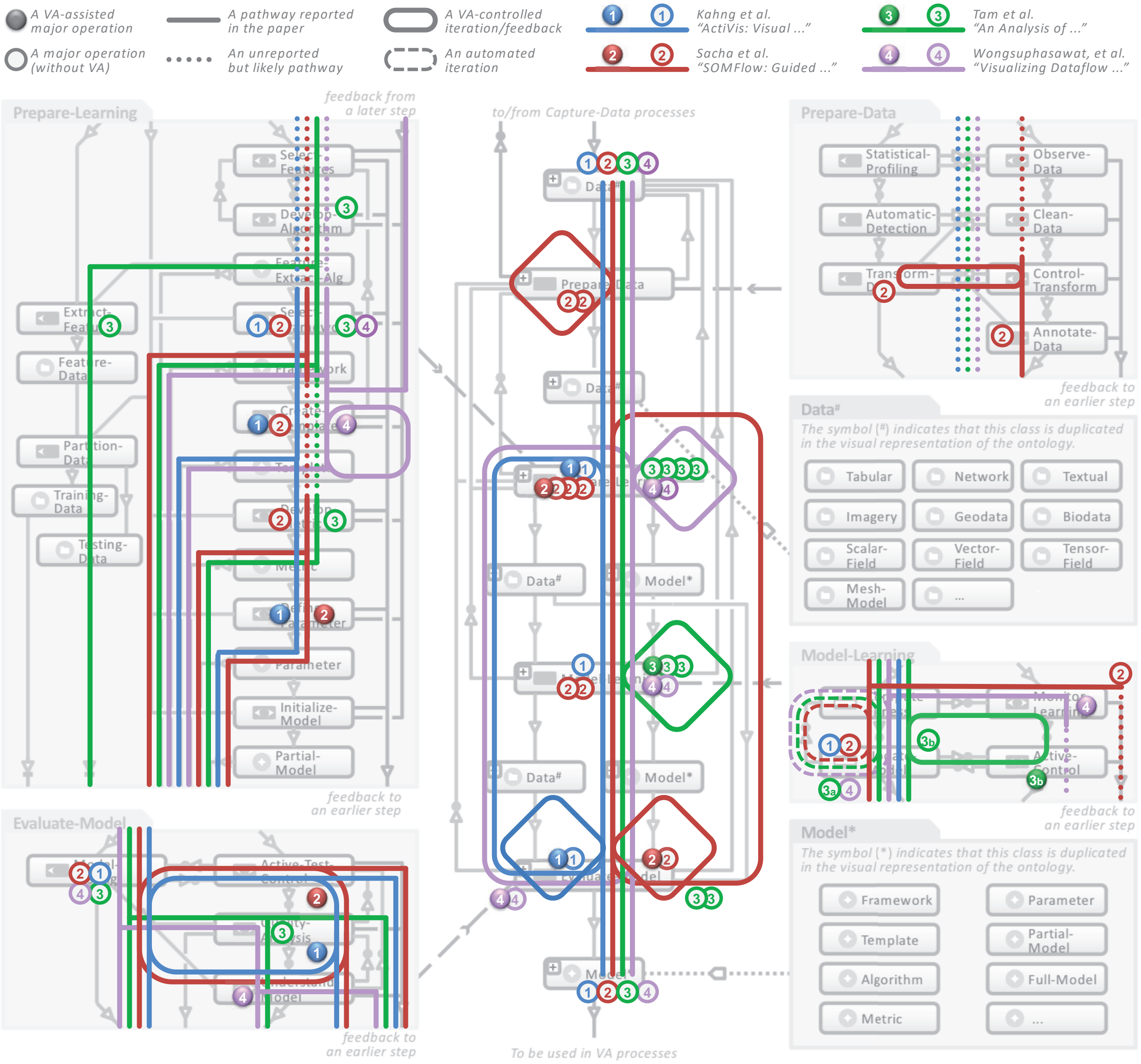

Fig. 6. Pathways within our ontology illustrated for the examples 1-4 of Table 1. The pathways are complemented with marked "bus stops" illustrating exactly at which stages visualization is used to improve the ML workflow.

lar instances of interest. The visualizations allow the user to explore and understand the model and the classification quality (Understand-Model, Quality-Analysis), while the node and instance selections allow an Active-Test-Control of the Model-Testing. Based on the observations (e.g., identifying neurons that did not activate or activate for all classes), the analyst can go back to the Prepare-Data step and improve the Template or the training Parameters.

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, the analyst visually analyzes the result and interacts accordingly by, e.g., refining cells of the Kohonen Map by training new SOM models by only considering subsets of the data. The resulting workflow is described by the red pathway No. 2 in Fig. 6. During the Prepare-Data step, the input data can be enriched (Annotate-Data) or transformed by, e.g., normalizing time series in the data. Subsequently, a SOM template can optionally be modified by configuring the distance metric or parameters (Prepare-Learning step). Then, the training process can be monitored visually (Model-Learning). After the training, a machine supported visual evaluation takes place in which the analyst can visualize different quality measures (Evaluate-Model). Dependent on this step and on the overall goal of the respective analysis, the analyst can refine the network or parts of it by training new models for subsets of the data (loop from Evaluate-Model to Prepare-Data). The development of the last trained model is comprehensibly displayed in a history graph which hierarchically embeds all previously created models (visualized as closed circuit at Evaluate-Model). Hence, in that approach it is possible to go back to a previous state in the evolution of the model and start over or develop the model in two separate directions in parallel.

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 No. 3 in Fig 6. The main difference between the two workflows is in the Model-Learning step. 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 various ML frameworks, 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 process 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 relations 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

[1] B. Alsallakh, A. Jourabloo, M. Ye, X. Liu, and L. Ren. Do convolutional neural networks learn class hierarchy? *IEEE Trans. on Visualization and Computer Graphics*, 24(1):152–162, 2018. doi: 10.1109/TVCG.2017.2744683

[2] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[3] N. Andrienko, T. Lammarsch, G. Andrienko, G. Fuchs, D. Keim, S. Miksch, and A. Rind. Viewing visual analytics as model building. *Computer Graphics Forum*, 2018. doi: 10.1111/cgf.13324

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[5] K. Brodlie, D. Duce, D. Duke, et al. Visualization ontologies: Report of a workshop held at the national e-science centre. *Report e-Science Institute*, 2004.

[6] M. Cannataro and C. Comito. A data mining ontology for grid programming. In *Proc. of the 1st int. Workshop on Semantics in Peer-to-Peer and Grid Computing*, pp. 113–134, 2003.

[7] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[8] S. Carpendale, M. Chen, D. Evanko, N. Gehlenborg, C. Goerg, L. Hunter, F. Rowland, M.-A. Storey, and H. Strobelt. Ontologies in biological data visualization. *IEEE Computer Graphics and Applications*, 34(2):8–15, 2014.

[9] D. Cashman, G. Patterson, A. Mosca, and R. Chang. Rnnbow: Visualizing learning via backpropagation gradients in recurrent neural networks. In *Workshop on Visual Analytics for Deep Learning (VADL)*, 2017.

[10] M. Chen and A. Golan. What may visualization processes optimize? *IEEE Trans. on Visualization and Computer Graphics*, 22(12):2619–2632, 2016. doi: 10.1109/TVCG.2015.2513410

[11] M. Chen, G. Grinstein, C. R. Johnson, J. Kennedy, and M. Tory. Pathways for theoretical advances in visualization. *IEEE Computer Graphics and Applications*, 37(4):103–112, 2017. doi: 10.1109/MCG.2017.3271463

[12] J. Choo and S. Liu. Visual analytics for explainable deep learning. *CoRR*, abs/1804.02527, 2018.

[13] D. J. Duke, K. W. Brodlie, and D. A. Duce. Building an ontology of visualization. In *15th IEEE Visualization 2004 Conference, Extended Abstract*, pp. 7–8, 2004. doi: 10.1109/VISUAL.2004.10

[14] D. J. Duke, K. W. Brodlie, D. A. Duce, and I. Herman. Do you see what I mean? *IEEE Computer Graphics and Applications*, 25(3):6–9, 2005. doi: 10.1109/MCG.2005.55

[15] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, P. Fiaux, and C. Andrews. The human is the loop: new directions for visual analytics. *Journal of Intelligent Information Systems*, 43(3):411–435, 2014. doi: 10.1007/s10844-014-0304-9

[16] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 36(8):458–486, 2017. doi: 10.1111/cgf.13092

[17] J. A. Fails and D. R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pp. 39–45. ACM, 2003. doi: 10.1145/604045.604056

[18] S. Falconer. Ontograf. *Protégé Wiki*, accessed in June 2018.

[19] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54, 1996.

[20] J. Fekete and R. Primet. Progressive analytics: A computation paradigm for exploratory data analysis. *CoRR*, abs/1607.05162, 2016.

[21] O. Gilson, N. Silva, P. Grant, and M. Chen. From web data to visualization via ontology mapping. *Computer Graphics Forum*, 27(3):959–966, 2008.

[22] T. M. Green, W. Ribarsky, and B. D. Fisher. Visual analytics for complex concepts using a human cognition model. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 91–98, 2008. doi: 10.1109/VAST.2008.4677361

[23] T. M. Green, W. Ribarsky, and B. D. Fisher. Building and applying a human cognition model for visual analytics. *Information Visualization*, 8(1):1–13, 2009. doi: 10.1057/ivs.2008.28

[24] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC Press, 2010.

[25] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *CoRR*, abs/1801.06889, 2018.

[26] A. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Proc. 3rd IEEE International Conference on Data Mining*, pp. 541–544, 2003.

[27] G. Jawaheer, P. Weller, and P. Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interactive Intelligent Systems*, 4(2):8:1–8:26, 2014. doi: 10.1145/2512208

[28] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau. ActiVis: Visual exploration of industry-scale deep neural network models. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):88–97, 2018. doi: 10.1109/TVCG.2017.2744718

[29] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Wrangler: interactive visual specification of data transformation scripts. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 3363–3372, 2011. doi: 10.1145/1978942.1979444

[30] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods – a survey. *ACM Computing Surveys*, 39(4), 2007.

[31] D. A. Keim, G. L. Andrienko, J. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. In *Information Visualization - Human-Centered Issues and Perspectives*, pp. 154–175. 2008. doi: 10.1007/978-3-540-70956-5_7

[32] D. A. Keim, J. Kohlhammer, G. P. Ellis, and F. Mansmann. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[33] S. Khan, U. Kanturska, T. Waters, J. Eaton, R. Banares-Alcantara, and M. Chen. Ontology-assisted provenance visualization for supporting enterprise search of engineering and business files. *Advanced Engineering Informatics*, 30(2):244–257, 2016.

[34] J. Kralj, P. Panov, and S. Džeroski. Expanding the OntoDM ontology with network analysis tasks and algorithms. *18th International Multiconference Information Society - Intelligent Systems Conference*, 2015.

[35] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E. Bertini. A workflow for visual diagnostics of binary classifiers using instance-level explanations. *CoRR*, abs/1705.01968, 2017.

[36] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):109–119, 2018. doi: 10.1109/TVCG.2017.2745178

[37] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. deFilippi, W. F. Stewart, and A. Perer. Clustervision: Visual supervision of unsupervised clustering. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):142–151, 2018. doi: 10.1109/TVCG.2017.2745085

[38] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu. Analyzing the training processes of deep generative models. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):77–87, 2018. doi: 10.1109/TVCG.2017.2744938

[39] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):91–100, 2017. doi: 10.1109/TVCG.2016.2598831

[40] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu. Visual diagnosis of tree boosting methods. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):163–173, 2018. doi: 10.1109/TVCG.2017.2744378

[41] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. Understanding hidden memories of recurrent neural networks. *CoRR*, abs/1710.10777, 2017.

[42] T. Mühlbacher, L. Linhardt, T. Möller, and H. Piringer. TreePOD: Sensitivity-aware selection of pareto-optimal decision trees. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):174–183, 2018. doi: 10.1109/TVCG.2017.2745158

[43] M. A. Musen. The protégé project: A look back and a look forward. *AI Matters*, 1(4):4–12, 2015. doi: 10.1145/2757001.2757003

[44] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks. doi: 10.23915/distill.00010

[45] P. Panov, S. Dzeroski, and L. N. Soldatova. Ontodm: An ontology of data mining. In *Workshop Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, pp. 752–760, 2008.

[46] P. Panov, L. N. Soldatova, and S. Dzeroski. Towards an ontology of data mining investigations. In *Discovery Science, 12th International Conference, DS*, pp. 257–271, 2009. doi: 10.1007/978-3-642-04747-3_21

[47] P. Panov, L. N. Soldatova, and S. Dzeroski. OntoDM-KDD: Ontology for representing the knowledge discovery process. In *Proc. of the 16th*

Int. Conference on Discovery Science DS, pp. 126–140, 2013. doi: 10.1007/978-3-642-40897-7_9

[48] A. M. Pérez, C. P. Risquet, and J. M. Gómez. An enhanced visualization ontology for a better representation of the visualization process. *ICT innovations*, 83:342–347, 2010.

[49] N. Pezzotti, T. Höllt, J. V. Gemert, B. P. F. Lelieveldt, E. Eisemann, and A. Vilanova. DeepEyes: Progressive visual analytics for designing deep neural networks. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):98–108, 2018. doi: 10.1109/TVCG.2017.2744358

[50] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of the Intern. Conf. on Intelligence Analysis*, vol. 5, pp. 2–4, 2005.

[51] J. Polowinski and M. Voigt. VISO: a shared, formal knowledge base as a foundation for semi-automatic infovis systems. *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 1791–1796, 2013. doi: 10.1145/2468356.2468677

[52] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[53] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):101–110, 2017. doi: 10.1109/TVCG.2016.2598838

[54] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):61–70, 2017. doi: 10.1109/TVCG.2016.2598828

[55] D. Sacha. *Knowledge Generation in Visual Analytics : Integrating Human and Machine Intelligence for Exploration of Big Data*. PhD thesis, University of Konstanz, Konstanz, 2018.

[56] D. Sacha, M. Kraus, J. Bernard, M. Behrisch, T. Schreck, Y. Asano, and D. A. Keim. SOMFlow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):120–130, 2018. doi: 10.1109/TVCG.2017.2744805

[57] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 2017. doi: 10.1016/j.neucom.2017.01.105

[58] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. P. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1604–1613, 2014. doi: 10.1109/TVCG.2014.2346481

[59] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017. doi: 10.1109/TVCG.2016.2598495

[60] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive Kohonen maps. *Information Visualization, Palgrave Macmillan*, 8(1):14–29, 2009. doi: 10.1057/ivs.2008.29

[61] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012. doi: 10.2200/S00429ED1V01Y201207AIM018

[62] G. Shu, N. J. Avis, and O. Rana. Investigating visualization ontologies. In *Proc. of the UK e-Science All Hands Meeting*, 2006.

[63] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *CoRR*, abs/1804.09299, 2018.

[64] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):667–676, 2018. doi: 10.1109/TVCG.2017.2744158

[65] K. Sudathip and M. Sodanil. Ontology knowledge-based framework for machine learning concept. In *Proc. of the 18th int. Conference on Information Integration and Web-based Applications and Services, iiWAS*, pp. 50–53, 2016. doi: 10.1145/3011141.3011207

[66] G. K. L. Tam, V. Kothari, and M. Chen. An analysis of machine- and human-analytics in classification. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):71–80, 2017. doi: 10.1109/TVCG.2016.2598829

[67] C. Upson, T. Faulhaber, Jr., D. Kamins, D. H. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. van Dam. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.

[68] J. J. van Wijk. The value of visualization. In *Proc. IEEE Visualization*, pp. 79–86, 2005.

[69] M. Voigt and J. Polowinski. Towards a unifying visualization ontology. *Technical Report, Technische Universität Dresden*, 2011.

[70] J. Wang, X. Liu, H.-W. Shen, and G. Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):81–90, 2017. doi: 10.1109/TVCG.2016.2598830

[71] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in TensorFlow. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):1–12, 2018. doi: 10.1109/TVCG.2017.2744878