Department: Visualization Viewpoints
Editor: Theresa-Marie Rhyne, theresamarierhyne@gmail.com

The Flow of Trust: A Visualization Framework to Externalize, Explore & Explain Trust in ML Applications

Stef van den Elzen[1], Gennady Andrienko[2], Natalia Andrienko[2], Brian D. Fisher[3], Rafael M. Martins[4], Jaakko Peltonen[5], Alexandru C. Telea[6], and Michel Verleysen[7]

[1] Eindhoven University of Technology, NL; [2] Fraunhofer Institute IAIS, Sankt Augustin, Germany and City, University of London, UK; [3] Simon Fraser University, Surrey, CA; [4] Linnaeus University, Vaxjo, SE; [5] Tampere University, FI; [6] Utrecht University, NL; [7] University of Louvain, BE

***Abstract*—We present a conceptual framework for the development of visual interactive techniques to formalize and externalize trust in Machine Learning (ML) workflows. Currently, trust in ML applications is an implicit process that takes place in the user's mind. As such, there is no method of feedback or communication of trust that can be acted upon. Our framework will be instrumental in developing interactive visualization approaches that will help users to efficiently and effectively build and communicate trust in ways that fit each of the ML process stages. We formulate several research questions and directions that include: (a) a typology/taxonomy of trust objects, trust issues, and possible reasons for (mis)trust; (b) formalisms to represent trust in machine-readable form; (c) means by which users can express their state of trust by interacting with a computer system (e.g., text, drawing, marking); (d) ways in which a system can facilitate users' expression and communication of the state of trust; and (e) creation of visual interactive techniques for representation and exploration of trust over all stages of a ML pipeline.**

■ **INTRODUCTION** The last two decades have been marked by the explosion of *data* sources ranging over virtually all application types, such as multimedia collections (images, text, sound, videos), data tables from databases having increasing diversity and size, and measurements from the physical world such as GPS and trajectory data. As the size, diversity, and complexity of the data increased, so did the awareness that higher-level *information* can be extracted from these sources. A particularly successful manner to infer such information from raw data is proposed by Machine Learning (ML). ML applications construct *models* of the phenomena from which data is acquired and aim to generate predictions related to these phenomena in the presence of new, unseen, data. ML applications covering classification and prediction are increasingly present in diverse contexts of decision support and task automation by generating outputs relevant to a human user in the given context.

As ML models become increasingly powerful, so does their engineering and inherent complexity. As such, an increasingly important research direction targets *explainable AI* (XAI), *i.e.*, the creation of methods and tools that shed light on the functioning of such models to their various users. However, while such techniques help users to understand how a model is structured and works, they currently do not directly cover building *trust* in the model (and/or the process leading to it). We consider XAI and trust to be loosely related but independent topics. Providing explanations may help to increase trust, but not necessarily: even if a system provides a perfect explanation of how its model works, the user may still not trust the system, due to *e.g.* wrong model decisions. The reverse also holds: although XAI might show a model's flaws, users might still have high trust in the system, due to *e.g.* faith in the authority or organization behind it, or because they simply lack (domain) knowledge to understand the explanation. As such, in current systems trust is typically represented implicitly, lacking *e.g.* explicit interaction and support feedback mechanisms. In this paper, we argue that trust (or the lack thereof) in ML applications is an aspect as important as – if not more important than – understanding the operation of such applications.

Currently, Visual Analytics (VA) & ML applications lack an interface for expressing trust and/or distrust. What is missing from current interfaces is both (a) ways for the user to express and explain (dis)trust, and (b) ways to capture and manage such (dis)trust in an explicit manner such that it can directly affect the visual interactive ML process. We believe that in complex systems, *expressing trust* (beyond a superficial overall level of trust) requires exploratory, interactive visualization support to discover the areas of trust and distrust along with their reasons.

As a first step, to create awareness, and to work towards treating trust as a first-class citizen in designing and reasoning about VA applications that use ML, we introduce a conceptual framework that captures the flow of trust. This framework lays a foundation for externalization, exploration, and explanation of trust using interactive visualization techniques during development of ML & VA applications and helps with post-hoc analysis of existing systems. The framework guides researchers and tool creators in making trust explicit by considering different trust elements: (a) content - what needs to be captured and explicitly represented; (b) target form of the content; (c) communication media (*e.g.,* text, drawing, marking); (d) facilitation (*e.g.,* prompting, templates); and (e) visualization techniques. Our contributions are:

- a conceptual framework that enhances the ML pipeline with a model that captures the flow of trust, and,
- guides the construction of visual analytics solutions that support and explicitly manage trust development;
- the application of our framework to examples of current ML models extended with interactive visualization support for evolution of trust;
- identification and discussion of research directions concerning trust.

A motivating example

To corroborate the need for a framework for externalizing, exploring, and explaining trust and to illustrate the presentation of the framework, we introduce a real-world example. It involves our experiences gained during the creation and usability testing of an optimization model for flight scheduling.

Domain problem

The airspace (particularly, in Europe) is divided into compartments, called sectors, within which the traffic is supervised by air traffic controllers. The sectors have limited capacities defined as the maximal safely manageable number of flights that can cross a sector in one hour. Flights are conducted according to plans. Initial flight plans are prepared by airlines intending to conduct the flights. It often happens that the demand for a sector, *i.e.,* the number of flights that need to cross it within an hour, exceeds the sector capacity and thus creates a so-called hotspot. For safety reasons, it is necessary to eliminate the hotspots by modifying parts of the flight plans. The most common modification is delaying a flight. It is sometimes possible to modify flight routes so that overloaded sectors are avoided while the route lengths do not increase significantly. The task of an optimization model is to create a daily flight schedule such that no hotspots will emerge. The input data consist of a set of initial flight plans; the output is a set of final flight plans [2].

Solution development

The model for solving the problem was built using historical data D for a large region of Europe and a time span of one year. For each day, there were sets of initial and final flight plans. A flight plan in D has the form of a trajectory consisting of geographic positions (waypoints) and time stamps. This format was not suitable for model development. The model developers (MD) defined a set of features (*i.e.,* numeric attributes) derivable from the original data and suitable for model building and thus transformed D to D'. Later on, it turned out that the derived features were not easily understandable to the domain users (DU). Also, the selection of these particular features was not properly justified.

MD built the model M by means of a reinforcement learning algorithm. The flights were modeled as agents taking decisions to delay for X minutes. Later on, this approach to modeling was questioned as the behavior of the resulting model did not match users' way of reasoning. Assuming that reinforcement learning was the right method to create a model, a better idea might be to model sectors as agents.

The built model M (a neural network) was not inherently explainable; therefore, MD created a surrogate model M' to explain the behavior of M. M' was a combination of decision trees with a depth up to 35 levels. The amount of information was far beyond the human capability to comprehend it. Although visualization developers (VD) invented some tricks to present M' in a simplified and aggregated form, it was not enough for a good understanding of the model behavior.

The execution of M is an iterative process of modifying an original flight schedule. Each step results in a version of the flight schedule that differs from the previous one in terms of flight delays and sector loads. VD created a visualization that presented an overview of the process with summarized changes from step to step and allowed to explore the details and compare different versions of the schedule. The visualization showed how hotspots were resolved at the cost of flight delays. At the overall level, the delays appeared to be justified; still, DU were not convinced that the delays were not longer than necessary, and there was no good way to check this. At the detailed level, DU questioned the choice of the flights to be delayed. Although XAI methods were used, and the explanations could be explored, trust in the model was still low.

The output of M was viewed and explored by means of a visualization showing the final flight schedule and enabling its comparison with the original schedule. M' was used for providing explanations for modifications of a particular user selected flight plan. The explanations were presented with decision rules. DU found them unsatisfactory: excessively long, hard to understand due to complicated non-intuitive features, and failing to explain the choice of the flights to be delayed. DU concluded that they are not convinced that the model operates properly and thus cannot adopt such a model for use in practice.

This project provided a number of lessons concerning possible trust issues along the process of model development and use. In brief, the model developers put too high trust in the chosen modeling method and in the capability of a surrogate model to explain the logic of the trained model. DU, in turn, did not trust the model as a whole due to lack of understanding of its behavior, and they did not trust the proposed solutions due to lack of evidence of the solutions being optimal.

Related work

The importance of users' trust in ML and the ways in which visualizations affect it have been discussed and summarized in a few survey papers in recent years. For instance, Endert et al. [13] identified *Enhancing Trust and Interpretability* as one of the open challenges and opportunities for ML and VA. According to the authors, analysts can build mental models of how ML models work via interactive visualization, which will increase trust. This happens in two different levels of cognition: a *qualitative* level, where the most important goal is to communicate information about the model in the most intuitive way, such as using classical visualization methods; and a *quantitative* level, to provide sound evidence to confirm the insights obtained in the previous level.

Sperrle et al. [14] provided a systematic analysis of how evaluations are carried out in *Human-Centered Machine Learning* papers, with trust as one of the important focuses of the survey. They identify trust issues in relation to the interaction between the performance and the presentation: even VA systems with the highest usability must consider the performance of their underlying ML models in order to remain useful, while, on the other hand, well-performing ML models might not be used to their full potential if users do not trust them. Trustworthiness is considered an important dimension of analysis of both model properties ("A model can be considered trustworthy when users believe it is correct") and the explanations themselves ("The ability for the explanation to be believed in or accepted by the user as an honest representation or correct description"). The authors indicate, however, that only a small percentage of the analyzed papers actually evaluate such characteristics: 10% for trustworthiness as a model property and 6% for trustworthiness in explanations.

Probably the most related work to ours is the survey by Chatzimparmpas et al. [15], where a comprehensive mapping of the currently available literature on using visualization to enhance trust in ML models is provided. The authors discuss which visualization techniques are used, how effective they are, and the domain areas they are applied to, including a conceptual discussion of what trust means in ML and what challenges are still open. However, the issue of explicitly expressing and/or managing trust within the VA pipeline itself is not discussed in any of these surveys or their analyzed papers. While most of the related works mention the increase of trust in ML as one of their most important goals, they do not discuss how to directly achieve (or manage) that in a concrete manner. We intend, in this paper, to bridge this gap by proposing and discussing the design decisions behind a concrete framework where trust is a first-class citizen within the VA workflow itself.

Table 1. Proposed trust framework key requirements.

Key requirement	Detailed explanation ("The framework should...")
Tasks	Support trust expression, explanation, development, and communication.
Coverage	Apply to all steps of the ML pipeline (model design, training, execution, and result usage).
Generality	Support any type of ML application (*e.g.,* classification, regression) and technique (*e.g.,* feature engineering, deep learning, supervised/unsupervised learning).
Versatility	Address a broad class of users (*e.g.,* scientists, ML professionals, nonspecialist users).

Trust as first-class citizen

Based on the motivating example and related work, we argue that trust should be considered a 'first-class citizen' throughout the entire process of constructing and using ML applications, much like data provenance has become a first-class citizen in visualization pipelines [3]. For this, we propose a conceptual framework to represent, express, explore, communicate, and develop trust. Table 1 lists the key requirements this framework aims to comply with, based on the authors' own experience in building VA & ML applications. This list is not exhaustive but shows the requirements we believe are minimally needed.

To build this framework, we start bottom-up by first considering the traditional ML process. Figure 1 (bottom) depicts this as a data flow pipeline (data = sharp-corner boxes, operations = rounded-corner boxes). It starts by (1) acquiring *training data* T for the intended ML application. Using T, (2) ML professionals build and train a ML model M for the problem at hand. The model is next evaluated by its intended customers (3). Eventually, these decide to deploy and execute it (4) on post-training data D (also called 'unseen' data in ML). Finally, the model produces a set of results $O = M(D)$ which are then used for the application at hand (5). By externalizing trust we connect the different user roles (for more details see Sec. The flow of trust). Note that we do not explicitly show model monitoring and retraining, as we consider these re-iterations of (parts of) the pipeline.

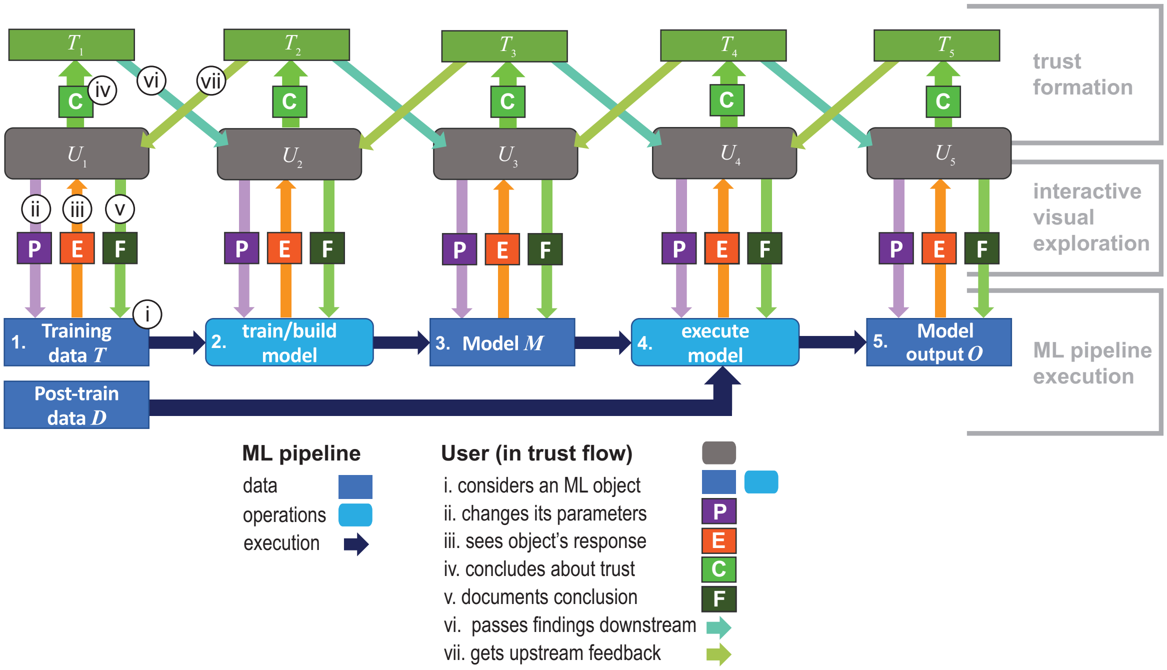

Figure 1. Trust modeling and flow throughout the construction and use of ML applications (see Sec. Trust as first-class citizen).

Each step of the ML pipeline can be characterized by five key elements (Figure 1 markers i-v, shown only for pipeline step 1 to limit drawing clutter): *Users* consider their object of interest in the ML pipeline (i). This can be either a tangible object (training data T, trained model M, or model output O) or a process (model building, model execution). To assess the object, they next change its various *parameters* (ii) and observe their *effect*, *i.e.*, how the object responds to parameter changes (iii). Based on this, they reach a *trust* conclusion (iv), which they next detail and document by providing *feedback* (v). These elements are described below (with additional examples in Table 2).

User roles: a role models the types of *activities* performed by a user involved in a given pipeline step. These can be taken by different, or the same, persons, depending on the application context, much like roles in the classical software engineering pipeline [12]. For instance, in a production setting, scientists or field researchers collect the training data (1); ML engineers construct the ML model (2) which is then deployed by IT professionals (3) and used in applications by the general public (4-5). In contrast, in a research or prototyping setting, all roles are often assumed by the same person.

Parameters: these describe how users *interact* with their object of interest (purple arrows marked P in Figure 1). For instance, training data can be re-sampled or transformed in various ways (1); training tunes various model hyper-parameters (2); a trained model is deployed on platforms having different computing power provisions (3-4); and the model's outputs are shown to the end user via various parameterized visualizations (5).

Effect: this captures how the object under study *reacts* to changes of its parameters P and is shown in Figure 1 by the orange arrows marked E. Effects can range from simple numerical results, *e.g.*, accuracy scores during training, to complex visualizations that depict the changing activations of units in a neural network during inference. Note that E also includes XAI techniques appropriate at each pipeline step. Exploring E allows users to form a mental model of the studied object and ultimately *explain* its behavior.

Trust: as users iteratively repeat the change-parameters-explore-results loop (ii-iii) outlined above, they build an increasingly clearer trust (or lack thereof, with all in-between nuances

Table 2. Examples of user roles, exploration parameters, explanation of ML behavior, trust aspects, and trust feedback mechanisms for the five steps of a generic ML pipeline.

Training data T	**User role**	Collects and curates training data from a given application area.
	Parameters	Affect the data representation (*e.g.,* sampling and reconstruction parameters).
	Effect	Shows data properties (outliers, clusters) and potential problems (errors, missing values, duplicates).
	Trust	Data are sufficient, of good quality, and capture well the modeled phenomenon.
	Feedback	User determines unfit training data, *e.g.* missing, wrong, or duplicate values or poorly samples the intended distribution.
Model building	**User role**	ML practitioner involved in architecting, coding, training, and testing the model M.
	Parameters	Feature selection and engineering; problem decomposition; hyper-parameters tuned during model engineering.
	Effect	Shows M's behavior in data and parameter spaces during training.
	Trust	Model works well for all applicable data and parameters and its sensitivity to data/parameters is understood.
	Feedback	Indicates that some of M's decisions (*e.g.* for specific samples) do not look correct and need improvement.
Model M	**User role**	ML practitioner; model evaluator (domain expert or certification body) determining model suitability for adoption.
	Parameters	Users explore model behavior by *e.g.* applying it to different inputs, which act as parameters changed by the user.
	Effect	Model specific methods *vs.* model agnostic methods. Depends on whether M is inherently interpretable or not [9].
	Trust	Model is sound – works correctly, is efficient, well explained, and suitable for its intended usage.
	Feedback	Some model blocks are not needed or too complex; M is (not) understandable / (not) applicable to user's context.
Model execution	**User role**	Domain expert/integrator building an end-to-end solution using a given model.
	Parameters	Control the model's execution (*e.g.,* memory and processor time available for a run).
	Effect	How the model modifies the solution during its execution process.
	Trust	Solution improves as the model runs; process converges fast enough; model avoids local minima.
	Feedback	The solution is evolving (in)appropriately.
Model output O	**User role**	End user of the ML pipeline (scientist, domain expert, ML engineer, non-specialist).
	Parameters	Control how the outputs are shown (*e.g.,* which text-based or visualization method is used).
	Effect	Bring insight how the model produces the output; XAI methods (LIME, SHAP, counterfactuals, local surrogate models).
	Trust	Based on domain knowledge, the output of M is plausible and in line with the users' mental model(s).
	Feedback	Selection of data items that comply to the users' mental model or not (continuous scale).

possible) of the objects under study. The actual trust *conclusion* formed by users is shown by the green boxes T_i in Figure 1 top. These conclusions can be simplistically represented by values on a binary (yes/no) or on an ordinal (low to high) scale, but the trust state may be more complex and nuanced (*e.g.*, not equal for different components or aspects of the object). Importantly, this trust forms up in the *mind* of the users (arrows marked C in Figure 1). As different user roles exist, it follows that *trust* has different meanings for the various pipeline steps (T_i, $1 \leq i \leq 5$, Figure 1 top). For example, a model engineer will trust a *model* M if it shows a good training convergence and it scores highly during ML testing scenarios; these aspects are not relevant for end users who will trust the *output* O of a ML pipeline if O is in line with their common expectations of what the pipeline should do.

Feedback: as explained, trust forms in the mind of a user. Modeling trust as basic ordinal values (see above) offers a simple way to communicate a user's conclusion trust-wise, but does not further explain *why* the user has reached that conclusion. This is important since both when trust T_i is high or low one needs to understand the reasons to react accordingly. Also, users may have unequal trust to different parts or aspects of the object of interest. We propose to solve this by making the above aspect explicit: A so-called *feedback* mechanism, denoted by the green arrows marked F in Figure 1, enables users to annotate their object of interest to explain what they (mis)trust and why. For example, end users can mark specific outputs O of a pipeline as untrustworthy, *e.g.,* too many delayed flights; model engineers can mark aspects of a training process as suspicious, *e.g.,* poor convergence curves or non-monotonic changes of performance indicators; and data scientists can mark samples of a training set as potentially incorrectly acquired or labeled.

The flow of trust

We have described so far how individual user roles arrive at achieving their own views of trust and how they can externalize these. In practice, this per-step formed trust next *travels* along the ML pipeline to connect user roles. We model this in Figure 1 by the diagonal arrows at the top. Arrows marked (vi) indicate trust *provisions* given by earlier pipeline steps to later ones, *e.g.*, a ML engineer providing statistical metrics of model performance and representation of the distribution of model errors to justify their trust in the model engineering they performed. Simply put, trust 'flows forward' in the pipeline to convince subsequent users that the objects they are provided with are trustworthy enough. Trust also propagates backwards: arrows marked (vii) indicate trust *requirements* set by later pipeline

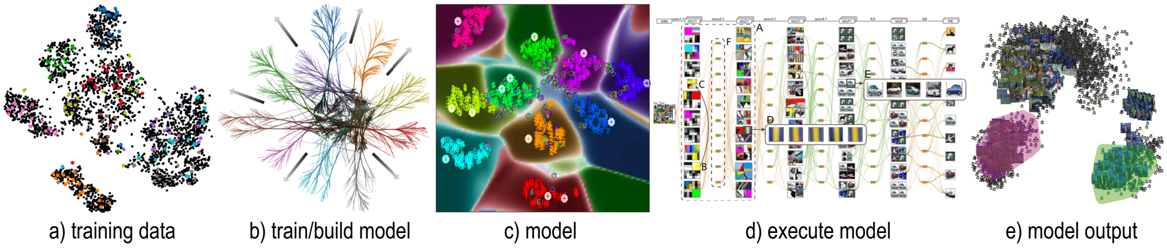
a) training data
b) train/build model
c) model
d) execute model
e) model output

Figure 2. Examples of using interactive visualization in the trust modeling and flow.

steps to earlier ones, *e.g.*, an end user telling his smart-driving car provider that they do not trust the car's behavior in certain conditions. Upon receiving such signals, users of earlier steps need to adapt their objects.

The flow of trust occurs by first passing the key conclusions *between* user roles (a U_i trusts object i this much, *i.e.*, to level T_i). Next, additional information on *why* the respective trust level was reached can be passed along to justify the conclusion. Such information can also include details, such as particular components or aspects of the object, or conditions this level of trust refers to. The communication of trust takes the form of passing the annotated objects (obtained via the feedback F) that motivate the respective trust conclusion. Also, note that trust typically flows over multiple layers and multiple times during the lifetime of a ML pipeline, *e.g.*, from the final users back to the scientists preparing the training data T. This is similar to the lifetime of software systems: the forward execution of the pipeline (and forward trust flow) is analogous to forward software engineering from requirements gathering until the first deployed version. The backward trust flow is analogous to the collection and processing of change requests during software maintenance [11].

Role of Interactive Visualization

Visualization plays a crucial role in our framework. First, it enables the *exploration* of the ML objects of interest by varying parameters P and observing effects E, since these objects are large, abstract, and complex. Secondly, interactive mechanisms allow users to select parts of these objects and annotate them to express their trust conclusions, thus to create the *feedback* F. And thirdly, visualization enables an explicit representation of the trust (*e.g.*, to track over time).

Tens of such visualization mechanisms exist – for a recent survey, see [4]. Figure 2 shows five such examples, one per pipeline stage. We selected techniques using dimensionality reduction (DR) as an underlining mechanism for ease of presentation and to demonstrate the model- and visualization-agnostic pipeline.

Training data: DR is the tool of choice in unsupervised learning to display large collections of high-dimensional samples to observe how these group (or not) into multiple clusters. In semi-supervised learning, labeled samples are colored by their labels (Figure 2a), enabling users to determine where in the training data to next perform annotations to enrich otherwise poorly-labeled training sets [5], and, thereby, improve their trust in such training sets.

Train/build model: DR can be used to visualize the evolving activations in the last hidden layer of a deep model (latent space). Figure 2b shows such evolutions as class-colored trails in a projection space which increasingly diverge as training progresses. The visual separation of trails allows users to gauge their trust in the training and also spot outlier samples for which training did not perform well [6].

Model: Classifiers can be assessed beyond typical aggregate metrics such as accuracy by plotting so-called decision boundary maps (Figure 2c). These enrich a classical scatterplot-like DR projection of the input data space by coloring every pixel of the projected space to show the label (and its confidence) inferred by the model at that location [7]. Bright areas indicate regions of low confidence where the classifier is to be less trusted.

Model execution: To understand how large deep models process unseen input data, one can use DR to cluster their neuron activations and next depict the most salient input-data patterns that these respond to [8]. Figure 2d shows such patterns overlaid atop a clustered network architecture which helps users gain trust by understanding how such black-box models actually operate.

Model output: Similar to the first stage, DR can be used to depict the output of a model, *e.g.* inferred classes, along with the input data (Figure 2e). This enables users to *e.g.* mark in which regions of the data space, *i.e.* for which kinds of inputs, they trust the model or not [10].

Intended use of the trust framework

The purposes of this conceptual framework are to define a new research area in visual analytics and to guide future research in this area. It is generally believed that VA can potentially help users to develop trust in ML models and, more generally, in various kinds of computational artifacts. However, the supposed help is currently limited to providing tools for interactive exploration of the artifacts (*e.g.,* with XAI techniques). Our framework states that trust formation depends not only on the information users can gain by exploring an object but also on the flow of trust along the pipeline of the object construction and use. Referring to Figure 1, previous research has been focused on supporting the operations *ii* (parameters) and *iii* (effects). Our framework shows the need to support also passing findings downstream (*vi*) and receiving upstream feedback (*vii*). The key challenge that needs to be solved for developing this kind of support is to enable and facilitate explicit expression of trust. In terms of Figure 1, the task is to enable the operation *v* (expressing trust feedback) so that its results can be passed through the links *vi* and *vii*.

The framework shows that the meaning and structure of trust may not be the same for the different kinds of objects along the pipeline. Consequently, it is necessary to consider the specifics of each kind of object for understanding what contributes to the formation of trust in it. Table 2 includes our initial ideas concerning the possible meanings of the trust. This understanding, in turn, enables researchers to think how the essential ingredients of trust can be expressed explicitly. In other words, for a given kind of object, researchers will define, first, a conceptual model of the object-specific trust and, second, a suitable language to represent the trust. On this basis, researchers should work on developing interactive visual interfaces to facilitate externalization of the trust by the user (using the conceptual model to guide the user) and representation of the externalized trust by means of the language.

Solving the problem of trust externalization enables further research on supporting the trust flow along the pipeline. Typically, uncertainty also plays a role here. Appropriately representing uncertainty and its propagation along the pipeline is important information for users to make conclusions about the degree of trust. However, like explanations in XAI, representation of uncertainty and evaluation of its impact on trust building is an established research topic [16]. In our framework, we assume that users receive all relevant information, including uncertainties, for making trust decisions. Our focus is trust expression and communication.

The key question is how to support users with different roles to use trust feedback from the previous and next steps of the pipeline in fulfilling their roles. A related question is how to capture the evolution of the trust of each user resulting from the trust flow. We would like to emphasize that the purpose of this framework is to define research directions and pose research questions but not yet to give answers to these questions. Let us re-consider our motivating example from the air traffic domain to ponder how the trust issues could be addressed according to the proposed framework with a post-hoc analysis.

In our motivating use-case, the model developers (MD) played the roles U_1, U_2, and U_3. The roles U_4 and U_5 belonged to the domain users (DU) helped by visualization developers (VD). Based on our framework, MD would be expected to pass their trust in the model they built further along the ML pipeline, *i.e.,* to VD and DU. MD would need to provide explicit trust feedback showing the reason for their trust, *i.e.,* they would need to present evidence that the model operates appropriately. This would motivate them to explore the model carefully in order to create annotated visualizations for the following users. Thus, to verify and express their trust in the model, MD could apply it to test cases and visualize the characteristics of model performance across the cases: how the counts of delayed flights, unresolved hotspots (if any), and the average and maximal delay duration depend on the original number of the hotspots and the number of involved flights. This would demonstrate to DU that the model performance is good.

In reality, MD were not used to doing visual explorations. Therefore, their trust was communicated implicitly without being supported by evidence. DU with the help of VD explored the model behavior and its solutions and found a number of reasons for mistrust, as described earlier. They provided their feedback orally and in written form. Since there was no convenient way for DU to complement their feedback with annotated illustrations, the comments were rather general and insufficiently informative for MD to understand and address the problems. If DU were enabled to interactively explore the visualization received from MD, in particular, consider details of selected test cases, they could mark the flights deemed to be excessively delayed and ask MD for providing justifications. In response, MD might visually demonstrate to DU how a decrease in the delays of the marked flights would lead to the appearance of unresolved hotspots. We believe that *explicit expression* and *appropriate representation* of the trust feedback would allow MD to better adapt the model to the needs of DU and also increase the DU's level of trust by communicating well-substantiated trust of MD forward along the ML pipeline.

Another example of the intended use of the framework (in a different domain) is sketched next. Assume an image classification model is built to predict item production faults. The end-user, who is responsible for picking out the faulty products from the assembly line, uses a VA system to identify faulty products. Imagine the following scenario: 1) The VA system reports a fault in the production. However, after inspection, it turns out that the product contains no faults and the user concludes that the ML model produced a misclassification. 2) After multiple misclassifications, the trust in the model decreases. The user expresses trust through direct manipulation of the trust object in the VA system, *e.g.,* a slider ranging from *no trust* to *full trust*. 3) Next, after some iterations, the trust drops below a predefined threshold. As a result, the misclassified items are annotated and passed downstream to the model stage where the responsible user role (*the model developer*) is notified. 4) The developer (visually) tests the generalization of the involved class, and unfortunately, the model does not generalize well for this class. Now, the trust in *the data* is lowered. The user passes the data distrust to the previous stage, along with (a visualization of) the data items of interest. 5) The responsible user role for the training data stage then inspects if the involved class labels are correct. This user concludes the labels are correct and expresses a high trust that is passed forward, along with the findings, to the model developer again. 6) The model developer can now trust that the data labeling is correct and starts improving the model by adding more instances of the problematic class to the training data.

This example is kept simple to demonstrate the main concepts; in reality the objects, models, and interactions are more complex.

Discussion

Explicitly modeling, interacting with, and visualizing trust in ML applications generates new questions and open areas for research. From the conceptual framework we derive and discuss five research directions for future work:

1) **Trust objects;** taxonomy of trust objects, trust issues, and possible reasons for (mis)trust.
2) **Formalisms** to represent trust in machine-readable form.
3) **Expression;** ways for users to express their state of trust by interacting with a computer system.
4) **Flow of trust;** ways to explore and develop trust over all stages of a ML pipeline using visual interactive techniques.
5) **Guidance;** ways to facilitate users' expression and communication of the state of trust using visual interactive techniques.

Trust objects: in this paper we identified and focused on the five trust objects of a traditional (classification/regression) ML pipeline: data, model development, model, model execution, and model output (see Figure 1, blue boxes). We believe our framework covers all main elements of the traditional ML pipeline at a high level of abstraction. The framework can be refined and applied to a broad range of ML model classes (classification, regression, optimization) as well as different methods of model building where trust objects are also likely involved (*e.g.,* reinforcement learning, active learning, self-supervised learning). As a first step towards development of applications with explicit trust, all trust objects should be identified and categorized using a taxonomy. For each trust object in this taxonomy, different trust challenges play a role *e.g.,* for the data object, trust in the data gathering/collection and subsequent labeling of the data plays a role; for the model output, trust in the model as well as (subsets of) the output is formed by the user. For a system that fully supports trust as intended with the conceptual framework (within and between each pipeline step), an identification and understanding of reasons for trust, or the lack thereof is needed.

Formalisms: currently trust is not expressed explicitly, but rather it implicitly forms in the mind of the user. As argued in this paper, we believe trust should be expressed externally (for storage, interaction, communication, and to act upon). Trust can be expressed in many ways (*e.g.,* through interactive widgets, emails from one user role to another, oral communication, or bug-reporting systems). To be able to reason about the most effective and efficient manner of externalizing trust, we need to devise generic formalisms to represent trust in machine-readable form.

Expression: An open area of research is the exploration of which visualization and interaction mechanisms are most effective to express trust. Next to visualization and interaction, the coarseness of trust needs to be researched – how many levels are appropriate, are they similar for each trust object, and is their scale linear? Also, we believe the expression of trust depends on the stage, user role, and task. A related question is how to support both expert and novice (non-ML) users. Furthermore, future research should focus on creating a convenient language for users to express their state of trust through interactions.

Flow of trust: an important aspect of the framework is the communication of trust between the different user roles. To support this flow of trust between user roles, we believe interactive visualization is crucial and can act as common ground between the different stages. For example, visualizations can be shared between two subsequent stages and serve as means of communication between both user roles. Next to design of interactive visualization techniques to support the flow of trust, also provenance plays a role here. A promising research area is how to capture, monitor, and visualize the evolution of trust over time, for exploration, analysis, and presentation.

Guidance: in similar spirit to exploratory visualization, where users are guided and steered towards interesting patterns, trust can also be used for guidance and assisting users in the analysis process of each stage. For example, users can focus on the subgroups with the most stable or highest trust by analyzing how trust evolved over time for a selected output (or group of outputs). Or if trust decreases over time, communicate this to the previous stage, such that this can be investigated and possibly fixed. For this, appropriate interactive visualization techniques should be developed. Similar to expressiveness, the methods and techniques should support guidance of both expert and non-expert users.

Conclusion

Up until now, trust has not been considered as an explicit element in the design and reasoning about visual analytics and machine learning applications. Rather, trust is an implicit process that takes place in the user's mind. We argue that trust should be externalized and treated as a first-class citizen. We present a framework that creates awareness and helps users to efficiently and effectively build and communicate trust in ways that fit each of the machine learning process stages. The framework is based on the traditional machine learning pipeline and extends this with elements of trust formation and interactive visual exploration. Key to our framework is the feedback loop *within* one stage through changing parameters, witnessing the effect or explanation, and providing trust feedback, and *between* stages, through passing or receiving externalized trust objects along the full pipeline (the flow or trust among different user roles). In addition to the framework, we identify and discuss five research directions for future work including trust objects, formalisms, expression, flow of trust, and guidance.

Acknowledgements

This work originates from the Dagstuhl Seminar on *Interactive Visualization for Fostering Trust in ML* (seminar 22351). Authors were supported by Academy of Finland decision 327352, by Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the *Lamarr Institute for Machine Learning and Artificial Intelligence* (Lamarr22B), by EU in projects *SoBigData++* and *CrexData*.

REFERENCES

1. N. Andrienko, G. Andrienko, L. Adilova and S. Wrobel, "Visual Analytics for Human-Centered Machine Learning", *IEEE Computer Graphics and Applications*, vol. 42, no. 1, pp. 123-133, Jan.-Feb. 2022.
2. Gennady Andrienko, Natalia Andrienko, Jose Manuel Cordero Garcia, Dirk Hecker, and George Vouros, "Supporting Visual Exploration of Iterative Job Scheduling", *IEEE Computer Graphics and Applications*, vol. 42, no. 3, pp. 74-86, May.-Jun. 2022.
3. Eric D. Ragan, Alex Endert, Jibonananda Sanyal amd Jian Chen, "Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes", *IEEE TVCG*, vol. 22, no. 1, pp. 31-40, Jan. 2016.
4. Rafael Garcia, Alexandru C. Telea, Bruno Castro da Silva, Jim Torresen, and Joao Luiz Dihl Comba, "A task-and-technique centered survey on visual analytics for deep learning model engineering", *Computers & Graphics*, vol. 77, pp. 30-49, 2018.
5. Barbara Benato, Alexandru Telea, and Alexandre Falcao, "Semi-Supervised Learning with Interactive Label Propagation guided by Feature Space Projections", *Proc. SIBGRAPI*, 2018.
6. Paulo Rauber, Samuel Fadel, Alexandre Falcao, and Alexandru Telea, "Visualizing the Hidden Activity of Artificial Neural Networks", *IEEE TVCG*, vol. 23, no. 1, pp. 101-110, 2016.
7. Alexander Schulz, Andrej Gisbrecht, and Barbara Hammer, "Using Discriminative Dimensionality Reduction to Visualize Classifiers", *Neural Processing Letters*, vol. 42, no. 1, pp. 27-54, Nov. 2014.
8. M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks", *IEEE TVCG*, vol. 23, no. 1, pp. 91–100, 2017.
9. Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, and Franco Turini, "A Survey of Methods for Explaining Black Box Models", *ACM CSUR*, vol. 51, no. 5, pp 1-42, 2019.
10. Paulo Joia, Fernando Paulovich, Danilo Coimbra, and Jose Albert Cuminato, "Local Affine Multidimensional Projection", *IEEE TVCG*, vol. 17, no. 12, pp. 2563-2571, 2011.
11. Priyadarshi Tripathy and Kshirasagar Naik, "Software Evolution and Maintenance: A Practitioner's Approach". Wiley, 2015.
12. Ian Sommerville, "Software Engineering". Pearson, 2015.
13. Endert, A., Ribarsky, W., Turkay, C., Wong, B. W., Nabney, I., Blanco, I. D., & Rossi, F. (2017, December). The state of the art in integrating machine learning into visual analytics. In Computer Graphics Forum (Vol. 36, No. 8, pp. 458-486).
14. Sperrle, F., El-Assady, M., Guo, G., Borgo, R., Chau, D. H., Endert, A., & Keim, D. (2021, June). A Survey of Human-Centered Evaluations in Human-Centered Machine Learning. In Computer Graphics Forum (Vol. 40, No. 3, pp. 543-568).
15. Chatzimparmpas, A., Martins, R. M., Jusufi, I., Kucher, K., Rossi, F., & Kerren, A. (2020, June). The state of the art in enhancing trust in machine learning models with the use of visualizations. In Computer Graphics Forum (Vol. 39, No. 3, pp. 713-756).
16. Sacha, D., Senaratne, H., Kwon, B.C., Ellis, G., & Keim, D.A. (2016, Januari) The Role of Uncertainty, Awareness, and Trust in Visual Analytics. In IEEE Transactions on Visualization and Computer Graphics (Vol. 22, no. 1, pp. 240-249).

Dr.ir. Stef van den Elzen is assistant professor of visual analytics at the Department of Mathematics and Computer Science at the Eindhoven University of Technology. His research interests include VA for explainable AI, Network & Event visualization. Contact him at s.j.v.d.elzen@tue.nl.

Dr. Gennady Andrienko (www.geoanalitycs.net) is a lead scientist responsible for visual analytics research at Fraunhofer Institute for Intelligent Analysis and Information Systems, PI of the Lamarr Institute for Machine Learning and Artificial Intelligence, and part-time professor at City University London. Gennady Andrienko is an associate editor of *Information Visualization* and *International Journal of Cartography*. Contact him at gennady.andrienko@iais.fraunhofer.de.

Dr. Natalia Andrienko is a lead scientist at Fraunhofer Institute for Intelligent Analysis and Information Systems, PI of the Lamarr Institute for Machine

Learning and Artificial Intelligence, and part-time professor at City University London. Results of her research have been published in two monographs, "*Exploratory Analysis of Spatial and Temporal Data: a Systematic Approach*" (2006) and "*Visual Analytics of Movement*" (2013), and in a textbook "*Visual Analytics for Data Scientists*" (2020). Natalia Andrienko is an associate editor of *Visual Informatics*. Contact her at natalia.andrienko@iais.fraunhofer.de.

Dr. Brian Fisher is a professor of Interactive Arts and Technology at Simon Fraser University, and a cognitive psychologist by training. He conducts scientific research on the role of interactive graphical information environments in expert reasoning. Contact him at bfisher@sfu.ca.

Dr. Rafael M. Martins is assistant professor at the Department of Computer Science and Media Technology of Linnaeus University, Sweden. His research interests lie mainly in the area of dimensionality reduction and its uses in complex visual analytics workflows, such as for explainable AI. Contact him at rafael.martins@lnu.se.

Dr. Jaakko Peltonen is full professor of statistics and data analysis at Tampere University, Finland. He is editor-in-chief of Scandinavian Journal of Statistics. His research interests are in statistical machine learning and exploratory data analysis including dimensionality reduction and visualization. Contact him at jaakko.peltonen@tuni.fi.

Dr. Alexandru Telea is full professor of visual data analytics at the Department of Information and Computing Sciences, Utrecht University. He authored the monograph “Data Visualization – Principles and Practice” (2008,2014). His research interests cover high-dimensional data visualization, visual analytics for explainable AI, image-based information visualization, and multiscale shape processing. Contact him at a.c.telea@uu.nl.

Prof. Michel Verleysen is full professor of machine learning at the ICTEAM institute of UCLouvain, Louvain-la-Neuve, Belgium. He co-authored the monograph “Nonlinear dimensionality reduction” (2007) and is the editor-in-chief of the Neural Processing Letters journal. His research interests cover high-dimensional data analysis and machine learning, nonlinear dimensionality reduction, small sample data analysis and biomedical applications of ML. Contact him at michel.verleysen@uclouvain.be.