Sharing and Automation for
Privacy Preserving Attack Neutralization

(H2020 833418)

D5.9 Demonstrator for uncertainty visualization (M21)

Published by the SAPPAN Consortium

Dissemination Level: Public

H2020-SU-ICT-2018-2020 – Cybersecurity

Document control page

Document file: D5.9
Document version: 1.0
Document owner: Franziska Becker (USTUTT)

Work package: WP5
Task: T5.5 Visualisation support for distributed and federated learning of models
Deliverable type: Demonstrator
Delivery month: M21
Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document History:

Version	Author(s)	Date	Summary of changes made
0.1	Franziska Becker (USTUTT)	15.12.2020	Added basic outline
0.2	Franziska Becker (USTUTT)	24.01.2021	Imported content from word document. Note that figures are old/temporary.
0.3	Franziska Becker (USTUTT)	27.01.2021	Extended content. Care: Figure numbers might not always be correct here.
1.0	Franziska Becker (USTUTT)	29.01.2021	Incorporated review comments.

Internal review history:

Reviewed by	Date	Summary of comments
Sebastian Schäfer (RWTH)	28.01.2021	Very minor comments on wording and some missing references.
Benedikt Holmes (RWTH)	28.01.2021	Minor grammatical mistakes and missing/wrong references.

Executive Summary

This deliverable details the efforts taken towards the visualization of uncertainty in the context of machine learning as described in task T5.5 “Visualisation support for distributed and federated learning of models”. The system developed in this deliverable intends to support non-expert users gain a better understanding of the results of machine learning models and thereby use them to make decisions. After the introduction, the deliverable starts with a description of the context for this work in the SAPPAN project and then covers related work from the research areas of uncertainty visualization and machine learning. Then, the overall design of the visualization system and its prototypical implementation is presented. Finally, future work is discussed, which concerns implementation aspects that need to be finished for the prototype, connections to collaborative learning that are part of task T5.5 and how the developed system may be evaluated.

Contents

1 Introduction

This deliverable reports on the efforts that were made to develop visualizations for the task T5.5 “Demonstrator for uncertainty visualization” as part of work package 5. First, we illustrate the context of this work in SAPPAN and then discuss related topics from the fields of uncertainty visualization and uncertainty in machine learning models. In the subsequent section, we detail the design and implementation of our visualization solution that aims to support security experts who have to make decisions based on the results of machine learning models. Finally, we discuss opportunities for future work which include the completion of the visualization system presented here and its possible adaption to the goals of deliverable D5.10, which is concerned with the visualization of uncertainty that arises in the context of different collaborative learning settings.

2 SAPPAN Context

The SAPPAN architecture distinguishes between the local (organization) and the global (sharing) level. On the local level, the dashboard allows each organization to explore their data using the dashboard’s different visualization components. On the global level, they can share their own data or use other participants’ shared data for their own purposes. The work presented in this report belongs to the local level, intended to be used by security experts to judge the decisions made by classifiers. Consequently, its visualizations are part of the dashboard being developed in work package 6.

3 Related Work

This section details related work that discusses uncertainty from the visualization and the machine learning perspective as well as the decision making process and reasoning strategies commonly employed during decision making tasks.

3.1 Uncertainty Visualization

Uncertainty can take many forms and stems from different sources. Bonneau [1] has identified three different sources of uncertainties that influence each other as shown in Figure 1. Pang and colleagues [2] make a similar distinction when they describe the introduction of uncertainty either at acquisition, at transformation or at visualization.

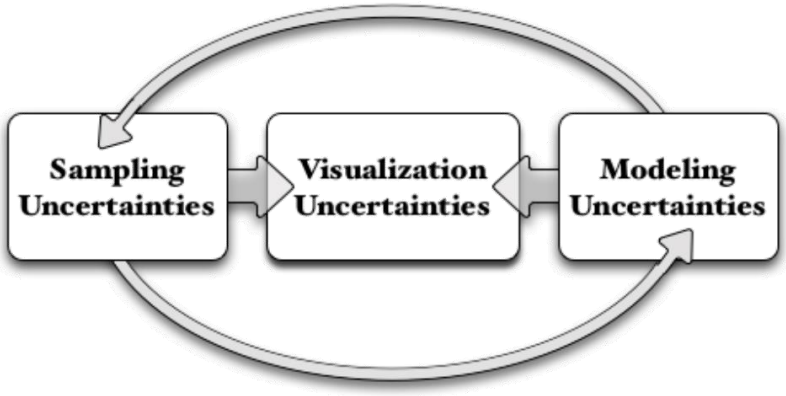

Figure 1: Sources of uncertainty and their relationship according to Bonneau [1].

Sampling uncertainty: This includes uncertainty in sampled data that has too much, too little or missing information. The latter in particular proves challenging since filtering out incomplete data may produce discontinuities or obscure patterns. Extrapolating from missing data by means of interpolation or other estimation techniques can also introduce errors. Such uncertainties may be approximated using quality metrics, providing more trust and confidence in the observations made from the visualization. An additional source of uncertainty in sampled data relates to meta information like the data source, age or gathering process.

Modeling uncertainty: Computational models most frequently contain uncertainties which may stem from the data they are based on, the process by which they are calculated or even the human factor involved in their design and creation. However, models may also contain some means of estimating the uncertainty in their predictions, ranging from a single error measures to confidence degrees where each possible output is associated with a likelihood.

Visualization uncertainty: Visualization can impact the propagation and perception of uncertainties in numerous ways. On the computational side, it is imperative to understand where sources of uncertainty can be found in calculations and input data. On the perceptual side, we must consider how cognition processes uncertainty visualizations and where there might be perceptual differences due to audience abilities, culture or applications tasks.

3.1.1 Common Uncertainty Visualizations

For scalar values, the most common strategy to make outliers visible is to visualize the data's distribution. The simplest and most popular visualization for this is a box plot (see left plot in Figure 2), which can display the mean or median, the first (25^{th} percentile) and third (75^{th} percentile) quartile and the minimum and maximum value. This collection of information is also referred to as the five-number summary. An extension of the box plot that incorporates the smoothed probability density of the data is given by the violin plot (see Figure 2 g)). This can be more informative than a box plot when the distribution has several peaks, but is also less readable and less well-known to the general population.

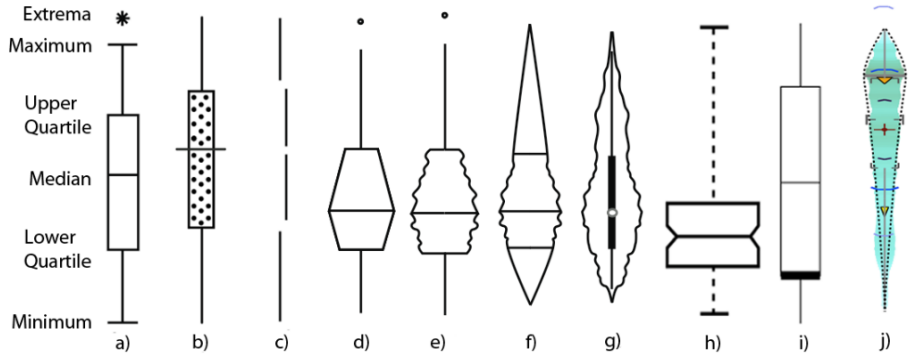

Figure 2: Variations of the box plot. Image and modified caption taken from [1]: "*a)* The construction of the boxplot *b)* Range plot *c)* Innerquartile plot *d)* Histplot *e)* Vaseplot *f)* Box-percentile plot *g)* Violin plot *h)* Variable width notched boxplot *i)* Skewplot *j)* Summary plot".

In the field of scientific visualization, uncertainty is often encoded using color, opacity or surface roughness. Figure 3 shows an example from medical visualization where different parts of a 3D volume are colored using a transfer function to indicate different levels of risk associated with classification. Such visualizations are often based on quantifiable uncertainty, searching for the right encoding to communicate these values effectively.

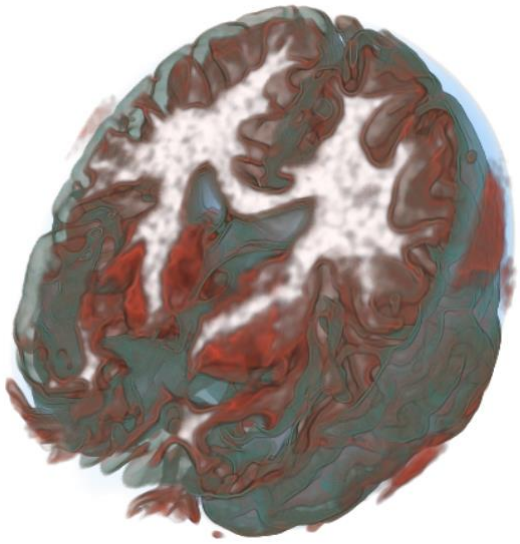

Figure 3: Image and caption taken from [1]: "A visualization of the brain using transfer functions that express the risk associated with classification."

3.1.2 Perception of Uncertainty Encodings

In order to visualize uncertainty, it is imperative to understand which visualization encodings effectively communicate the underlying uncertainty. However, which encoding is the most suitable can depend on the source of uncertainty and the task that the visualization is used for. For example, research [3, 4] has shown that blur is an intuitive encoding for uncertainty that aligns with common conceptual ideas of uncertainty as something unclear, cloudy and obscured. However, it has also been reported in [5] that blur can guide attention but is hard to quantify, with people only being able to differentiate between a few levels of blur and having trouble identifying objects with the same level of blur. Participants also reported that they disliked looking at blurred objects,

which may diminish their engagement. A more general insight from studies by [6] showed the participants preferred static displays that show uncertainty in parallel rather than toggling between an uncertainty and a value visualization.

3.1.3 Reasoning with Uncertainty

Deitrick [7] describes how the inclusion of uncertainty information changes judgements made by participants during decision making tasks based on static maps. He also found that the type of task may be more influential than the visualization method and that the inclusion of uncertainty information may decrease users' confidence in their decision. Skeels et al. [8] looked at how people from different fields and varying expertise view and deal with uncertainty in their profession, finding that they were "clearly aware of uncertainty at many levels in their data and expressed discomfort at their inability to be transparent about showing their uncertainty". In addition, they discovered that participants in their study dealt with uncertainty in one of two ways: accept the existence of uncertainty or try to become more certain. Participants adopt one of these strategies based on the potential impact of being wrong and the expected success of improving their certainty.

3.1.3.1 Reasoning Strategies

Reasoning is a core action performed by any potential user when deciding how correct or relevant a specific output of a deep learning model is. Reasoning has been studied extensively and has shown several strategies commonly employed by people to make sense of events, observations or concepts. In this section, we discuss reasoning methods gathered from previous research as presented by [9]. One family of such reasoning strategies are deductive, inductive and abductive reasoning. Deductive reasoning in general is to form a hypothesis and then search for confirming information. In the context of our use case, an example for such reasoning would be to form the hypothesis that the model's prediction is false and look for evidence to support this hypothesis. On the other hand, inductive reasoning reverses this concept by looking at observations and finding a matching hypothesis. Finally, abductive reasoning is a variation of the latter, where the simplest or most likely hypothesis is considered for a given set of observations.

Analogical reasoning is a frequently adopted strategy, based on the idea that when elements are similar in some respect, they might also be similar in others. It plays a central role in many aspects of everyday life, influencing for example memory access, learning and creativity. In addition, it is a form of reasoning that can easily be explained to another person since the underlying assumption that 'things that are similar may share even more similarities' does not require complex logical arguments which may be harder to communicate and understand. Finally, two forms of causal reasoning are contrastive "why not" and counterfactual "what if" reasoning. The focus for these types of reasoning is on the particular causes for an explanation and how these contrast with a given selection.

3.2 Uncertainty in Machine Learning

In machine learning, uncertainty is often discussed in terms of *aleatoric* vs. *epistemic* uncertainty. According to Hüllermeier [10], aleatoric uncertainty refers to randomness that is inherent to processes or data and cannot be captured or answered definitively.

An example for this is a coin flip, which can only be predicted via probabilities instead of precise results. In contrast, epistemic or systematic uncertainty stems from ignorance and can in principle be reduced with additional information. In the case of supervised learning, this type of uncertainty may be further divided into model and approximation uncertainty. Approximation uncertainty relates to approximation quality of the hypothesis the learner produces, which largely depends on how well the training data represents the ground truth and how large the dataset is. On the other hand, model uncertainty refers to the potential discrepancies between the actual hypothesis space and the hypothesis space induced by the choice of model and hyper-parameters.

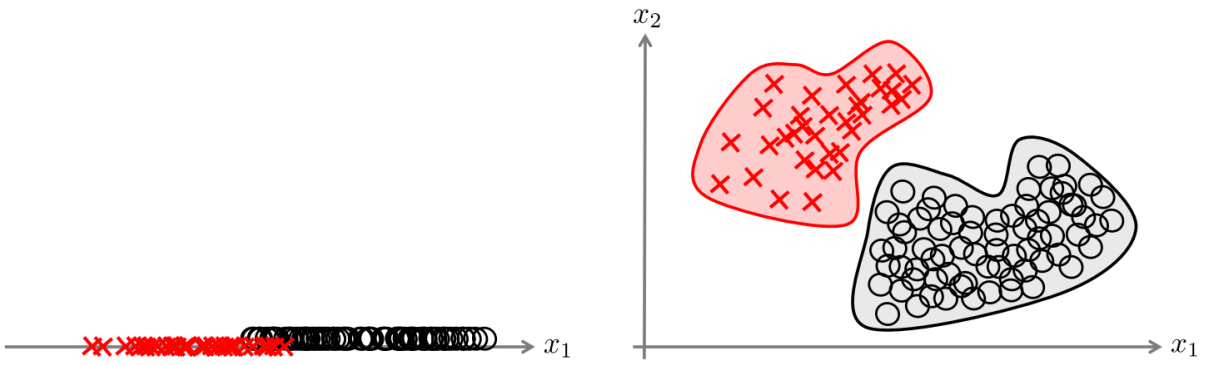

Figure 4: Caption from Hüllermeier [10]: "Left: The two classes are overlapping, which causes (aleatoric) uncertainty in a certain region of the instance space. Right: By adding a second feature, and hence embedding the data in a higher-dimensional space, the two classes become separable, and the uncertainty can be resolved"

However, aleatoric and epistemic uncertainty need not be absolute and depend on their specific context. Hüllermeier et al. give a good example of 1D data with two classes that overlap (see Figure 4), which is a source of aleatoric uncertainty since this overlap cannot be reduced by increasing the size of the dataset. However, adding another dimension to the data can make it separable, alleviating aleatoric uncertainty at the expense of increased epistemic uncertainty, as finding and fitting a model will likely require more data and be more difficult.

In regards to deep neural networks, epistemic uncertainty is seen as uncertainty about the (correct) model weights, while the probability distributions they commonly output capture the aleatoric uncertainty. Many models do not explicitly measure uncertainty and also cannot abstain from a prediction in cases of low confidence. While Bayesian neural networks and other methods solve this problem, there are currently no established methods that can be applied to different models after training to measure uncertainty.

3.3 Visualization for Machine Learning

Visualization for machine learning has picked up in popularity with the rise of state-of-the-art performance models in various domains. Many different visual analytic systems have been published to address specific tasks related to machine learning models such as classifier performance analysis [11, 12, 13, 14], explanation of deep learning model architecture and trained parameters [15, 16] or exploration of the dataset in combination with any of the former [17, 18, 19]. However, a large percentage of these works present complex interfaces tailored towards machine learning experts that wish

to understand and improve their models. Closest to the idea behind the visualizations developed for this deliverable are those that visualize performance in a detailed manner going beyond performance metrics, as performance acts a good indicator for uncertainty that stems from too little, incorrect or inherently confused data. In addition to visualizing uncertainty, the system should be user-friendly such that non-expert practitioners can work with it. Consequently, its design must not be overly complex or demand expertise in the fields of machine learning or visualization.

3.3.1 Non-expert practitioners

Chen et al. [20] performed a study to analyze how non-expert machine learning practitioners interact with different developed machine learning visualizations and what they valued or felt was missing. In line with traditional visualization methods, they observed that participants started with overview visualizations of model performance which helped them decide where to focus their attention next to find causes for errors. Almost all participants reported drill down capabilities to inspect specific instances as useful in order to identify patterns in the raw data. Participants also grouped instances and then compared these to find either uniting or separating patterns in the data that might account for model behavior. The authors also observed that participants used information from three different spaces: data space (raw data instances), feature space (features used by the model) and prediction space (predicted output from the model). All visualizations they presented to participants showed information from one or two of these spaces and subjects often wished for data from the missing space to be included. Another feature participants wanted was assistance, i.e. they wanted the visualization system to indicate, e.g. through highlight, problematic regions or errors that should be explored by the user. An example for such indicators might be to highlight incorrect predictions with high-confidence, since these represent the most 'severe' errors of the model. In the design phase of a visualization, this may be considered through appropriate visual encodings, i.e. in the context of a confusion matrix, erroneous predictions should have a distinct and attention-grabbing color scheme compared to correct predictions. Other issues that emerged in this study relate to the visualization's ability to scale to large data, number of features or number of classes and participants trust in the visualizations which they related to system transparency, i.e. how the visualization system modulates data to produce visualizations.

4 Visualization Design

The visualization design is heavily framed by the audience it targets and the context is should be used in: It is a visualization for security experts in order to gauge the plausibility of the prediction of a machine learning model. Consequently, the visualization is situated in the dashboard from task 6.1 in work package 6, together with other visualizations aimed at security experts. In addition, this requires the uncertainty that is visualized to be comprehensible to machine learning novices and users. For this reason, the visualizations focus on communicating the aleatoric uncertainty that stems from the training data. This has the advantage that extensive knowledge of machine learning techniques beyond the idea that models learn from data is not required. In accord with reasoning strategies, the core idea of our system design is to show several groups of instances with a specific relationship to the input instance and display how the model reacts to these groups in terms of performance. Given a specific instance of interest as input, we give the user context how the model views other instances

- where the class (ground truth) is the same as the prediction
- where the prediction is the same
- that are similar based on a specific metric
- that are dissimilar based on a specific metric

All of these groups allow the user to perform comparisons and investigations that align with different reasoning strategies. In addition, we visualize secondary information to provide transparency about selected data and visualizations we derive from them. Firstly, for each group, we show the percentage of the dataset this group covers and which classes it includes. Secondly, we display the overlap between the different groups, i.e. making it easily recognizable that confused instances are most likely confused due to their lexical similarity.

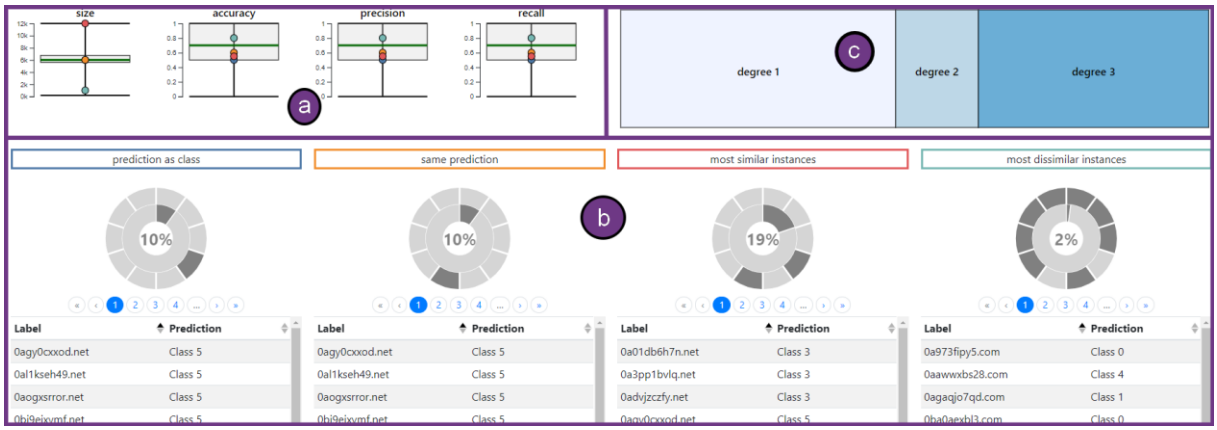

Figure 5: All visualizations developed to allow users to estimate classification uncertainty: a) shows an overview of classifier performance and class sizes in the dataset. In addition, each groups' value for these characteristics is indicated as circles. b) is the group visualization, showing which classes are contained in the group, how large this group is compared to the rest of the dataset and the raw data of the instances the group is made up of. c) displays overlap between the different groups, which allows the user to explore instances through interaction and dynamic level of detail.

4.1 Overview and Context

In the first step, we display an overview of the model's general performance and how the different groups we defined compare to it. In addition, the same is done for the size of each class in the dataset. We visualize these values as boxplots that indicate their distribution on a per-class basis. To indicate performance, we show accuracy, precision and recall. The different groups are indicated as colored circles in all the boxplots.

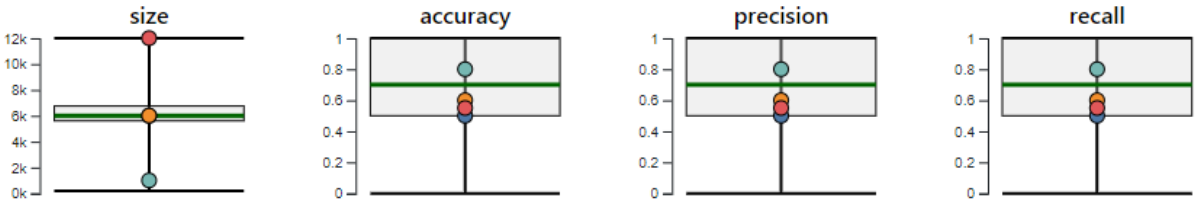

Figure 6: Performance and size overview using boxplots.

Hovering over any element of the box plot shows a small tooltip containing a textual description of the data it represents, e.g. hovering over the rectangle show the exact values for the 25 and 75 percentile values.

4.2 Group Visualization

For each group, we create a small box that contains its description, a sortable table of all connected instances and a visualization that displays how much of the complete dataset is contained in the group and which classes it includes. The latter is implemented as a simple nested donut chart. The outer donut displays all classes in the dataset, where each class has the same weight. Only those classes included in the group have full opacity, the rest is transparent. For the inner donut, the weight for each part (i.e. the group and the remaining dataset) equals the number of instances it contains. As for the outer donut, the element that corresponds to the size of the group has full opacity while the other one is transparent.

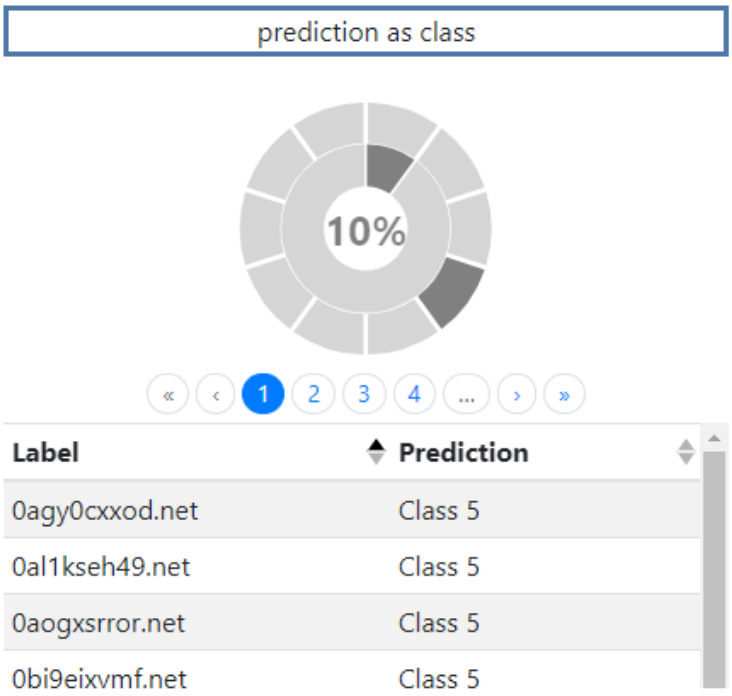

Figure 7: Group visualization with the outlined group description at the top, followed by the class inclusion and group size depiction in the middle and the raw data list at the bottom.

In the middle of the donut, we write the relative size of the group compared to the complete dataset. Hovering over any element of the donut chart displays a tooltip with the information the element represents and highlights the element by drawing an opaque black border. The table below the donut chart uses bootstrap [21], can be sorted by any column and uses pagination to avoid scalability problems with large groups.

4.3 Overlap Visualization

Using only the previous visualizations, the user cannot know exactly how these groups overlap, which may be of interest in order to understand inter-group connections. In addition, the user may want to inspect instances that belong to many or only a few of the groups in a quick and easy manner, without having to painstakingly browse through many pages of the different groups. This issue is addressed by the overlap visualization. It draws all instances of all groups (without duplicates) as rectangles in a grid layout, as seen in Figure 8. The instances are ordered by their degree of overlap within

the various groups. In addition, each rectangle is colored according to this degree along a sequential color scale from light to dark blue.

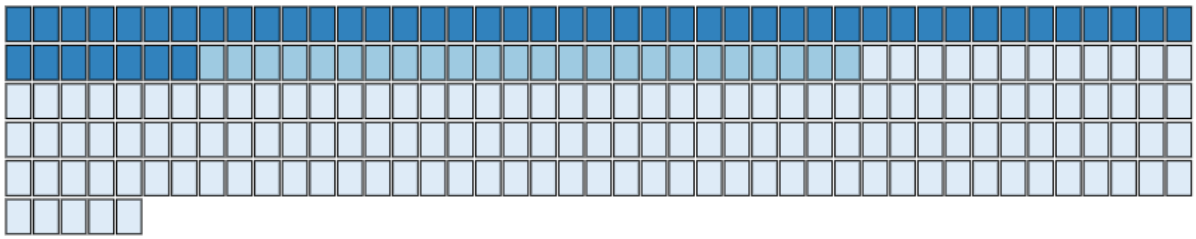

Figure 8: Group overlap visualization where each square corresponds to a single instance. The rectangle color indicates the degree of overlap with the different groups, where a darker color (more saturation) corresponds to a higher degree.

The data for each instance can be inspected by hovering over the respective rectangle, which shows a tooltip with the instance data and the associated groups. Unfortunately, this visualization does not scale well with a large number of instances, resulting in visual overlap that makes it unusable quite fast. To deal with this challenge, we devised two different variations of this design that work with larger groups. First, instances are divided into the different degree sets. Then each degree set is rendered as a rectangle, where its width encodes the cardinality of the respective degree set as shown in Figure 9. When the user clicks on a degree set, the visualization switches to another representation that is similar to the original design. The instances for the set are grouped together such that there is enough space to draw each of these groups as a rectangle without overlap (ref. Figure 10). The color of these rectangles is chosen according to the number of instances they group together. However, in order to distinguish this visualization from the original design, we choose colors from a grayscale color map.

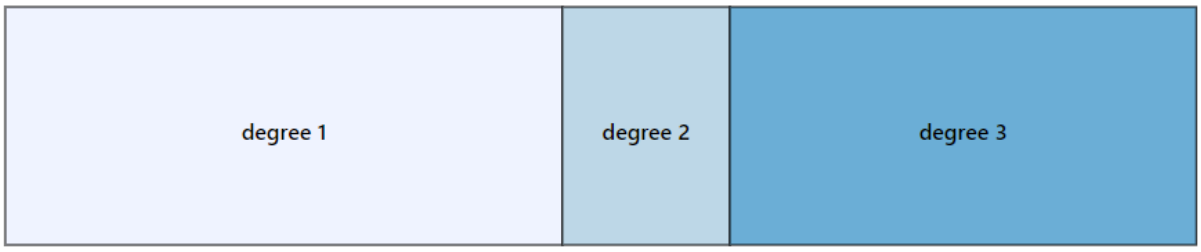

Figure 9: Group overlap visualization when the number of instances is too large to display each one as an individual rectangle. The color (value) indicates the degree of overlap with the different groups.

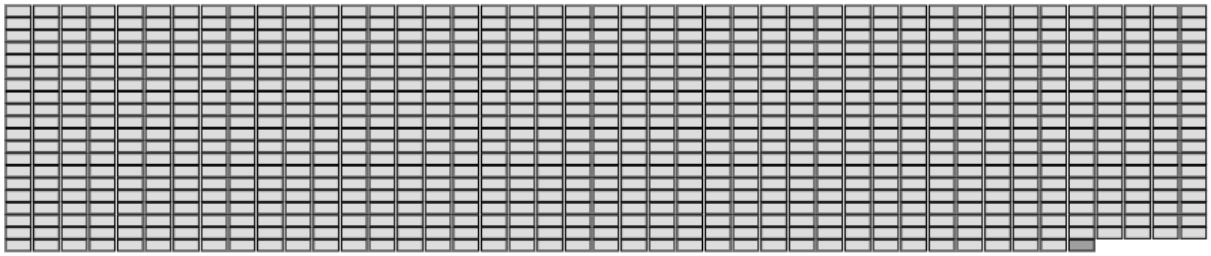

Figure 10: Group overlap visualization when the number of instances is small enough to display them. The color (value) indicates the number of instances grouped within a single rectangle.

4.4 Coordinated Views

We decided to connect the different components through interaction in a coordinated view fashion in order to increase both readability and comprehension of all visualizations. The primary result of any interaction is either to show exact values where they can only be estimated by looking at the visualization, or to connect the different components. For the former, we show exact data values when hovering over individual SVG elements in the overview visualization. The same holds true for the overlap visualization, where corresponding data is shown in a tooltip upon hovering over any element.

For the case where components are visually connected, we modify the overview and overlap visualizations whenever the user hovers over any element of the group visualization. First, this change is indicated by coloring the background of the group visualization in a light gray color. Then, in the overview visualization, we show a polyline (see Figure 11) that goes through all the points of the associated group in all of the four box plots and raises the matching circle elements such that they are drawn on top of any of the other circles, which avoids confusion due to occlusion.

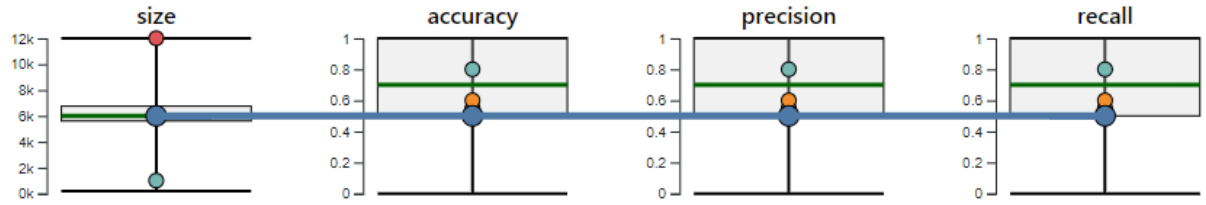

Figure 11: Performance and size overview using box plots with line highlight.

In the overlap visualization, we either reduce the opacity of the instances not associated with the group or add a bar indicating the percentage of instances for each degree set that belong to the group, depending on which overlap visualization is present (see Figure 12 to Figure 14). In the latter case, each group has a fixed band on the y-axis it is associated with, and the x-axis is used to indicate the number of instances in each degree set that belong to the selected group, colored according to the groups identifying color. Leaving the group visualization with the mouse cursor removes both effects, but they can be made permanent by clicking on the name of the group in the group visualization. Clicking on the same name once more reverses the permanent effect, while clicking on another name changes which group is used as reference.

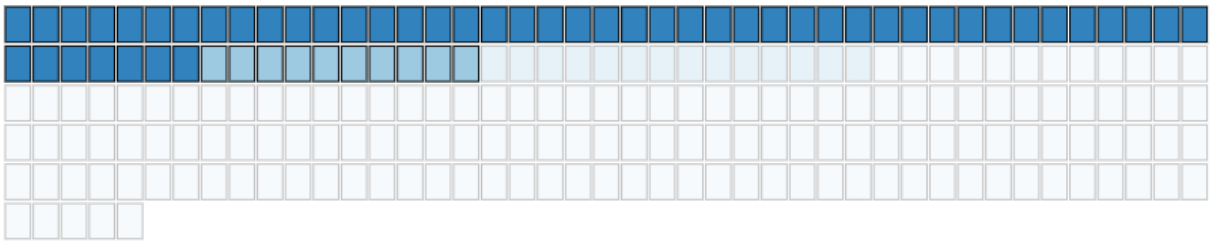

Figure 12: Highlight in the (small case) overlap visualization, emphasizing the instances contained in the selected group as opaque and others as transparent.

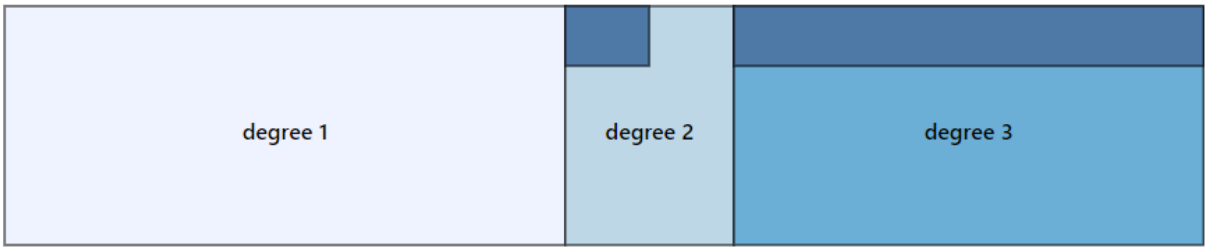

Figure 13: Highlight in the (large case) overlap visualization indicating for each of the sets of degrees how many belong to the selected group (on the x-axis).

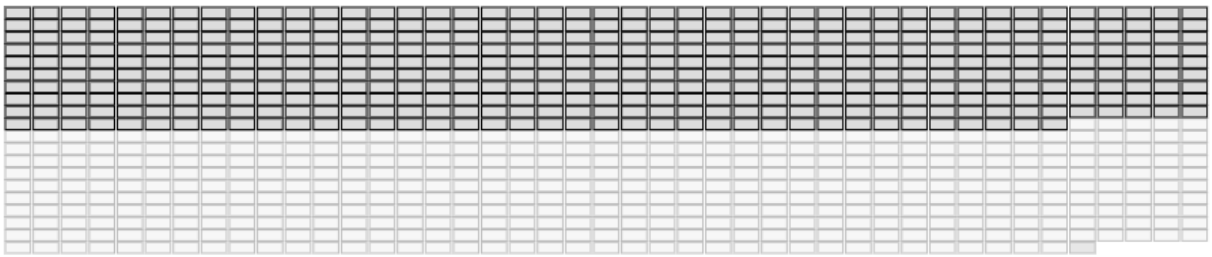

Figure 14: Highlight in the (grouped case) overlap visualization, emphasizing the instances contained in the selected group as opaque and others as transparent.

5 Prototype Implementation

The prototype for this design was implemented as a card in the web-based dashboard from work package 6. The frontend is therefore developed using D3.js [22] and the Vue.js framework [23] with TypeScript [24], which allows for a modular setup of all visualization components. The backend is implemented in C# using the ASP.NET Core framework, as detailed in deliverable 6.1. At the time of writing, the backend connection to a database that holds both the dataset used to train a model as well as related values for the model such as per-class performance values is not yet complete. Consequently, images included to illustrate the visualization concept use randomly generated dummy data.

Each one of the previously described visualizations is implemented as its own Vue component, managed by the over-arching ‘ExaML’ component which holds and distributes all the necessary data and propagates events between child components. Upon request by the user, the ‘ExaML’ component loads the required data from the database and then uses it to update its own data properties. In turn, this leads to an update of each child component, since each one of them has a property bound to a subset of this data. In addition to updating their data property, each child component has functions watching the respective data properties, which triggers the execution of the associated functions. These functions redraw the visualization of this component, using the updated data. The group visualization component consists of another component to visualize the included class and group size as well as bootstrap table that supports pagination to deal with large groups.

6 Future Work

In this section, we document into which directions future work can direct their efforts. In particular, we connect the work from this deliverable to related tasks in other deliverables.

6.1 Completion and Extension

Naturally, the most important future work that should precede all other efforts is to complete the connection to databases that contain real data. This should not create any problems, since the required information for the visualizations can easily be computed from a given deep learning model and corresponding dataset, as all visualizations rely either on the raw data or the model's prediction. Most of the data, i.e. accuracy, precision and recall values, can be pre-computed for all classes and predictions in the dataset, leaving only the mixed groups (similar and dissimilar instances) to be computed in real-time in the backend.

Another part that still requires work is the inclusion of interface elements to let the user configure both the metric and the maximum number of samples to consider for the 'similar' and 'dissimilar' groups. At the time of writing, these settings are hard-coded for convenience and illustration purposes.

6.1.1 Extensions

A promising direction we can imagine for this system is to allow for the specification of complex groups in a custom manner. While the predefined groups cover a larger portion of interesting subsets when viewed from a reasoning perspective, they cannot capture the complexity of interesting patterns and groups that may be present in the data. The challenge for such an idea is how to design the interface that lets the user specify such a complex subset. One concept we currently consider is to let the user concatenate simple filters using elementary operations also supported by common database systems such as 'equals', 'not equals', 'greater than', 'less than', 'includes' and the negation of any of these operation.

Another type of extension that seems like a straight-forward addition is the visualization of data features. While some datasets provided for the different use cases do not come with features, this could be solved by computing common features for the given data type, e.g. length for text, mean and standard deviation for numerical data and image characteristics like global contrast for image data. An even more interesting possibility this would open is to formulate and explore "What-If"-questions, i.e. how must features be adjusted to change the classifiers prediction? However, this requires the backend to directly execute predictions with the model, meaning it needs access to the trained model and a routine must be defined for each model how to prepare the data and how to use the model for prediction.

6.1.2 Uncertainty in Collaborative Learning

Another part of task T5.5 involves the implementation of visualizations for the communication of uncertainty generated through different kinds of collaborative learning processes. These different variations include sharing anonymized data or data features to train models locally and student-teacher approaches where a model is trained using labels determined by querying other parties' models with the same data. Although the visualizations developed in this deliverable target end-users rather than experts, it may provide useful to explore possibilities of including uncertainty generated by the collaborative learning, to provide better overall awareness of the model's creation process.

6.2 Evaluation

For an evaluation, the primary interest lies in finding out how well users are able to understand our visualizations and how their decisions are affected by them. Since we expect the result of this task to be extended to address uncertainty in collaborative

learning in some shape or form, we suggest to evaluate the visualization in its final form. For evaluation, we propose following some of the suggestions from [25]. They consider different types of use cases for explainable artificial intelligence visualizations and how they may be evaluated to measure their capacity to explain. Most interesting for the idea behind the visualizations developed in this task are the uses cases concerned with measuring how much participants' mental models align with the model's actual behavior. Conveying a model's functionality accurately is a key objective the visualizations should accomplish, since they serve as the end-users' proxy for the model. They suggest measuring the accuracy of a users' final decision when given the task to agree or disagree with a given set of inputs accompanied by model outputs. When compared with a control condition where the users are not supplied with the visualizations, but more conventional information in the form of a textual performance summary and a confusion matrix is provided, the visualization case should show improved accuracy. This task can be extended to also measure how fast participants are, how much they like the interface and how confident they are in their decision relative to the condition.

7 Summary

This report details the efforts taken toward the development of visualizations that let non-expert users explore a machine learning model's prediction in respect to the uncertainty that this prediction is accompanied by. We developed a visualization design based on common reasoning strategies that enable users to understand the prediction through analogy and contrast. The design and presentation is simple enough for users with different levels of expertise in regard to visualizations and machine learning to understand. In addition, it has the potential to scale to larger datasets, since all visualizations show either aggregated data or have a built-in mechanism to deal with a larger set of data. In the future, these visualizations may be extended to incorporate uncertainty from collaborative learning settings.

References

[1] G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliviera, K. Potter, P. Rheingans and T. Schultz, "Overview and State-of-the-Art of Uncertainty Visualization," in *Scientific Visualization. Mathematics and Visualization.*, London, Springer, 2014, pp. 3-7.

[2] A. T. Pang, C. M. Wittenbrink and S. K. Lodha, "Approaches to uncertainty visualization," *The Visual Computer,* pp. 370-390, 1997.

[3] A. M. MacEachren, R. E. Roth, J. O'Brien, B. Li, D. Swingley and M. Gahegan, "Visual semiotics & uncertainty visualization: An empirical study," in *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[4] N. Boukhelifa, A. Bezerianos, T. Isenberg and J.-D. Fekete, "Evaluating Sketchiness as a Visual Variable for the Depiction of Qualitative Uncertainty," in *IEEE Transactions on VIsualization and Computer Graphics*, 2012.

[5] R. Kosara, S. Miksch, H. Hauser, J. Schrammel, V. Giller and M. Tscheligi, "Useful Properties of Semantic Depth of Field for Better F+C Visualization," in *Proc. VisSym*, Goslar, Germany, 2002.

[6] M. Greis, A. Joshi, K. Singer, A. Schmidt and T. K. Machulla, "Uncertainty Visualization Influences how Humans Aggregate Discrepant Information," in *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, United States, 2018.

[7] S. A. Deitrick, "Uncertainty visualization and decision making: Does visualizing uncertainty information change decisions.," in *Proceedings of the XXIII International Cartographic Conference*, 2007.

[8] M. Skeels, B. Lee, G. Smith and G. G. Robertson, "Revealing uncertainty for information visualization," in *Information Visualization*, 2009.

[9] D. Wang, Q. Yang and A. Abdul, "Designing Theory-Driven User-Centric Explainable AI," in *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[10] E. Hüllermeier and W. Waegeman, *Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction,* arXiv preprint arXiv:1910.09457, 2019.

[11] M. Gleicher, A. Barve, Y. Xinyi and F. Heimerl, "Boxer: Interactive Comparison of Classifier Results," *Computer Graphics Forum,* 18 July 2020.

[12] S. Murugesan, S. Malik, F. Du, E. Koh and T. M. Lai, "DeepCompare: Deep Learning Comparison of Visual and Interactive Model Performance," *IEEE Computer Graphics and Applications,* pp. 47-59, September 2019.

[13] D. Ren, S. Amershi, L. Bongshin, J. Suh and J. D. Williams, "Squares: Supporting Interactive Performance Analysis for Multiclass Classifier," in *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[14] S. Amershi, C. Max, S. M. Drucker, L. Bongshin, P. Y. Simard and J. Suh, "ModelTracker: Redesigning Performance Analysis Tools for Machine Learning," in *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

[15] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kanhg and D. H. Chau, "CNN EXPLAINER: Learning Convolutional Neural Networks with Interactive," in *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[16] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu and S. Liu, "Towards Better Analysis of Deep Convolutional Neural Networks," in *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[17] T. Spinner, U. Schlegel, H. Schäfer and M. El-Assady, "explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning Thilo," in *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[18] N. Das, H. Park, Z. J. Wang, F. Hohman, R. Firstman, E. Rogers and D. H. (. Chau, "Bluff: Interactively Deciphering Adversarial Attacks on Deep Neural Networks," in *arXiv preprint arXiv:2009.02608*, 2020.

[19] S. Bullinger, C. Bodensteiner and M. Arens, "InstanceFlow: Visualizing the Evolution of Classifier Confusion at the Instance Level," in *arXiv preprint arXiv:2007.11353*, 2020.

[20] D. Chen, R. K. E. Bellamy, P. K. Malkin and T. Erickson, "Diagnostic visualization for non-expert machine learning practitioners: A design study," in *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Cambridge, UK, 2016.

[21] Team Bootstrap, "Bootstrap," Bootstrap Team, [Online]. Available: https://getbootstrap.com. [Accessed 29 January 2021].

[22] M. Bostock, "D3.js - Data-Driven Documents," [Online]. Available: https://d3js.org. [Accessed 29 January 2021].

[23] E. You, "Vue.js," [Online]. Available: https://v3.vuejs.org. [Accessed 29 January 2021].

[24] Microsoft, "TypeScript: Typed JavaScript at Any Scale.," Microsoft, [Online]. Available: https://www.typescriptlang.org. [Accessed 29 January 2021].

[25] D. Brittany, M. Glenski, W. Sealy and D. Arendt, "Measure Utility, Gain Trust: Practical Advise for XAI Researchers," in *IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*, Salt Lake City, UT, USA, USA, 2020.