VMS: Interactive Visualization to Support the Sensemaking and Selection of Predictive Models

Chen He
chen.he@helsinki.fi
University of Helsinki
Helsinki, Finland

Vishnu Raj
Hans Moen
Tommi Gröhn
vishnu.raj@aalto.fi
hans.moen@aalto.fi
tommi.i.grohn@aalto.fi
Aalto University
Espoo, Finland

Chen Wang
chwang@hit.edu.cn
Harbin Engineering University
Harbin, P. R. China

Laura-Maria Peltonen
lmemur@utu.fi
University of Turku
Turku, Finland

Saila Koivusalo
Helsinki University Hospital
Helsinki, Finland
saila.koivusalo@hus.fi

Pekka Marttinen
Aalto University
Espoo, Finland
pekka.marttinen@aalto.fi

Giulio Jacucci
University of Helsinki
Helsinki, Finland
giulio.jacucci@hiit.fi

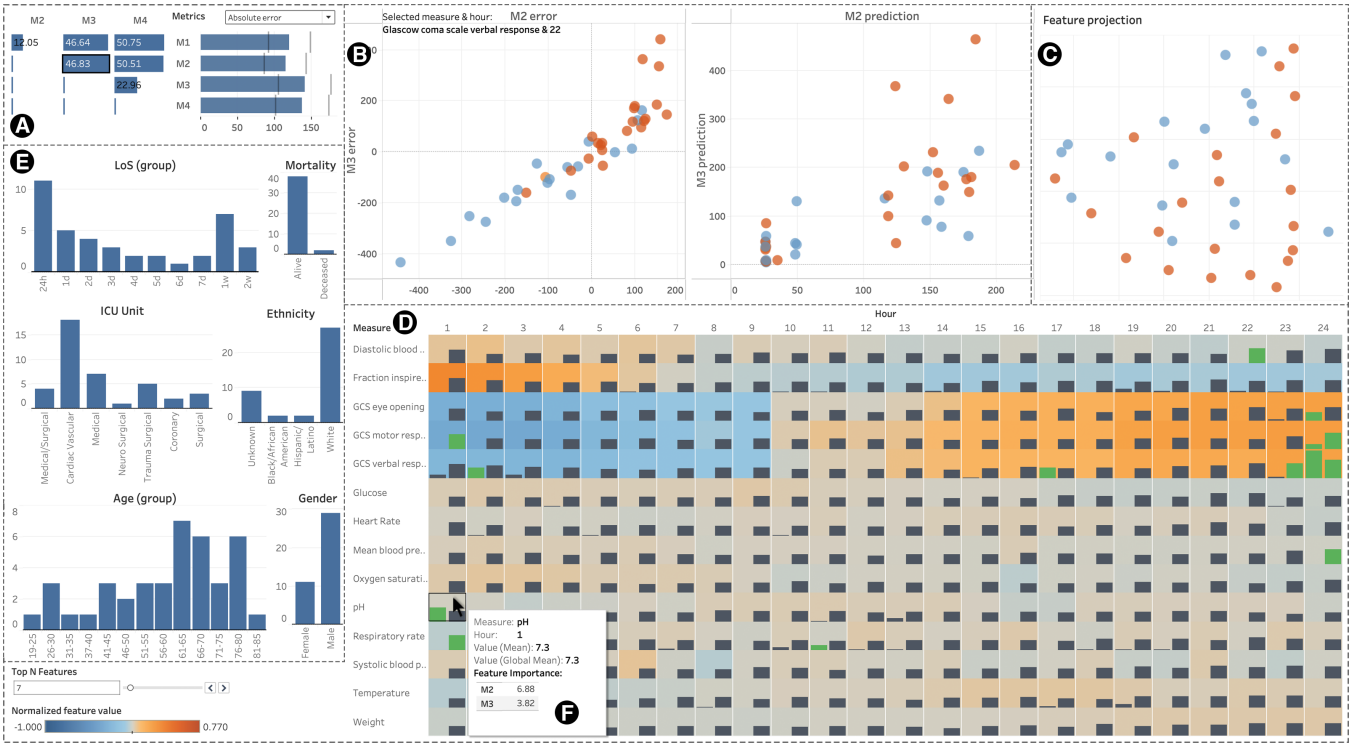

Figure 1: Screenshot of VMS comparing four machine learning models. (A) Model performance chart and prediction similarity matrix; (B) Individual prediction errors and values by the two models selected from (A); (C) Instance projection view; (D) Feature value & importance view on global/local scales; (E) Instance attributes as filters. (F) A tooltip on hovering indicates the feature information.

ABSTRACT

To compare and select machine learning models, relying on performance measures alone may not always be sufficient. This is particularly the case where different subsets, features, and predicted results may vary in importance relative to the task at hand. Explanation and visualization techniques are required to support model sensemaking and informed decision-making. However, a review shows that existing systems are mostly designed for model developers and not evaluated with target users in their effectiveness. To address this issue, this research proposes an interactive visualization, VMS (Visualization for Model Sensemaking and Selection), for users of the model to compare and select predictive models. VMS integrates performance-, instance-, and feature-level analysis to evaluate models from multiple angles. Particularly, a feature view integrating the value and contribution of hundreds of features supports model comparison on local and global scales. We exemplified VMS for comparing models predicting patients' hospital length of stay through time-series health records and evaluated the prototype with 16 participants from the medical field. Results reveal evidence that VMS supports users to rationalize models in multiple ways and enables users to select the optimal models with a small sample size. User feedback suggests future directions on incorporating domain knowledge in model training, such as for different patient groups considering different sets of features as important.

CCS CONCEPTS

• **Human-centered computing** → **Visual analytics**; **Empirical studies in visualization**.

KEYWORDS

XAI, interactive machine learning, MIMIC-IV

ACM Reference Format:

Chen He, Vishnu Raj, Hans Moen, Tommi Gröhn, Chen Wang, Laura-Maria Peltonen, Saila Koivusalo, Pekka Marttinen, and Giulio Jacucci. 2024. VMS: Interactive Visualization to Support the Sensemaking and Selection of Predictive Models. In *29th International Conference on Intelligent User Interfaces (IUI '24), March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3640543.3645151

1 INTRODUCTION

To compare and select machine learning (ML) models, typical averaging performance measures alone may not be sufficient. This is particularly the case where different subsets of the data and/or features, as well as different predicted results, may vary in importance relative to the task at hand. Incorporating eXplainable AI (XAI) techniques [31] and visual methods [2, 20] can increase the transparency and trustworthiness of the models.

IUI '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0508-3/24/03
https://doi.org/10.1145/3640543.3645151

Many visual analytics systems are proposed to address the demand for better-informed model comparison and selection (Section 2.2). However, most of these systems are designed for model developers, requiring extensive ML knowledge to use, and are not evaluated with target users on the tools' effectiveness in achieving the visualization goals (Table 1). This research designs interactive visualization to support model users to make sense of and select ML models.

Our target users, model users [14], are those who have some knowledge about ML and want to use ML to make predictions with their datasets. Suppose a case derived from practical scenarios [5] and research endeavors [1]: A person wants to make predictions using her dataset. Upon uploading the data and specifying the prediction task, she gets a list of candidate models trained on her data. How can the model user make sense of and select the optimal model(s) to use?

The problem can be broadly related to anyone since the knowledge of AI and the practice of open data is spreading rapidly. One example is the task we focus on in this work, where models are trained to predict patients' length of stay (LoS) in the hospital. If such models were to be used for resource allocation and cost planning, then it would be important for the model users to evaluate the models from more perspectives besides performance measures, such as how sensible the models are at predicting the most resource-demanding patients, to understand the reason behind the models' predictions and choose the optimal one(s) to use for the task at hand.

We propose a model-agnostic visualization to compare and select ML models with comprehensive instance- and feature-level analysis utilizing XAI methods. Still, the visualization is accessible to non-AI experts, i.e., model users, in rationalizing and selecting models, evidenced by a user study. The contribution of this research is twofold: First, we propose an interactive visualization, VMS (/'vmz/ Visualization for Model Sensemaking and Selection [18]), for model users to rationalize and select predictive models. To do this, we analyzed user needs and distilled six design requirements through close collaboration between the visualization designer and model users (Section 3). To address the requirements, VMS enables performance-, instance-, and feature-level visual analysis of ML models, allowing users to compare model pairs from multiple angles (Section 4). The views in VMS are interlinked, so users can reason from exploring, e.g., the correlation between instances' feature and prediction values and between feature value and importance for model sensemaking.

Second, we applied the method to compare four regression models predicting patient's LoS in the intensive care unit (ICU). A use case demonstrates the actual use of the application to support rationalized model selection (Section 5). A user study with 16 participants from the medical field (Section 6) reveals that 1) VMS facilitates the use of domain knowledge and supports model comparison and sensemaking in multiple ways, such as identifying feature interactions through color patterns and relating patient background information to assess feature importance, and 2) through the instance- and feature-level analysis with a small dataset, users tended to select the better performing models, validated using larger datasets. Section 7 discusses the limitations and future directions

of this work, such as bias in decision-making resulting from model users, data, and models; Section 8 concludes this research.

2 RELATED WORK

Involving humans in the ML process and making the models interpretable by humans is essentially the next evolutionary step AI is moving toward [35], let alone for legal reasons [21]. This section introduces interpretable ML methods, reviews existing model-agnostic visualizations for comparing models, and discusses the unique characteristics of this work.

2.1 Interpretable ML

Interpretable ML has been acknowledged as one important, arising field for statisticians [9]. Literature defines model interpretability as a capability to increase trust for complex models without relying purely on a single metric, such as predictive performance [8, 31]. Two main approaches exist to get a model interpretable: intrinsically interpretable models and post hoc interpretation methods. Intrinsically interpretable models include linear regression, decision trees, etc. However, complicated models, such as neural networks, have advantages in many application areas. Making these black-box models interpretable requires post hoc methods, which refer to methods that analyze the model after training.

Many post hoc methods have been proposed during the last few years, which can be generally categorized as model-specific versus model-agnostic and local versus global explanations. Model-specific methods, such as Integrated Gradients [37] and Tree SHAP [23], intend to explain the model through its internals, whereas model-agnostic methods investigate the relation between model input and output regardless of its internal structure. Local explanations, such as LIME [34] and counterfactual explanations, aim to interpret the prediction of a single instance, whereas global explanations, such as global surrogate models and permutation feature importance, attempt to explain the overall model behavior, such as how important the feature is to the model overall. The present study focuses on comparing any regression models; thus, we choose the model-agnostic explanation method SHAP [24], which also provides both local and global explanations.

Despite many proposals of such XAI methods, delivering these methods successfully to end users requires interdisciplinary efforts involving ML, human-computer interaction, and social science [30]. No universal interpretable system exists; target users with different roles have different needs for interpretability [41]. For instance, to interpret a ML system that provides medical advice to clinicians, model creators, such as the medical software company's employees, model users, clinicians in this case, and decision subjects – the patients, require different levels/aspects of understanding. Visualization needs to be carefully designed, addressing the needs of the target users.

2.2 Visualization to compare ML models

We review articles focusing on devising model-agnostic visual methods to support model comparison and selection. Table 1 shows the 16 systems we analyzed from their target user, prediction task, resulting views, and evaluation aspects. To categorize target users of the systems, referring to Hohman et al. [14], we identified two groups of people: **Model developers** who use visualization to refine and improve models at the development stage and **Model users** who want to select models to use. This involves different levels of ML knowledge while users interact with the visualizations. Four of the systems are designed for model users, and the rest is for model developers. For instance, three systems [3, 6, 12] visualized hyperparameters (not shown in Table 1), such as correlating them to model performance, to enable model developers to understand model behaviors. Regarding prediction tasks, five systems are for regression tasks and seven for classification, while four systems can support both types of tasks with some adjustment, such as changing the performance measures.

Skimming through the 16 systems, views for performance-, instance- and feature-level exploration are most common and naturally fall under the radar of our analysis. To provide an overview of the models, the systems support 1) model ranking on a user-selected [29] / user-weighted [3, 36] metric or 2) multiple metrics comparison in one or multiple views [10, 12, 29, 36, 40, 45]. For instance, multiple systems show models as rows and multiple metrics as columns [10, 12, 36, 45]; columns can be used to sort rows independently, similar to parallel coordinates, for performance exploration and comparison [10, 12].

For classification tasks, adapted confusion matrix [10, 12, 13, 33, 40] or parallel axes [29, 33, 39] are used to compare the models' class-level performance. Within the confusion matrix, each cell can contain the performance of multiple models. Depicting classes as parallel axes, the view can show each model's performance on each class linked by a line resembling parallel coordinates [33, 39]. Differently, ConfusionVis [39] proposed a class confusion view that integrated confusion matrix and parallel coordinates to compare models' class confusions.

To support instance-level analysis, several systems [3, 12, 29, 36] projected data using dimension reduction to show instance similarities. Manifold [43] devised a scatterplot to compare pair-wise model predictions over a class. Each axis represents one model depicting the prediction probabilities of the instances over the class, with each dot denoting an instance. Dots in each quadrant of the scatterplot indicate the same/different predictions by the models. The scatterplot could be easily repurposed for pair-wise comparison of regression models, visualizing prediction errors. To support the analysis of product demand forecasting, DFSeer [36] enables users to select a product to see its monthly forecast accuracy and compare past predictions of similar products to assess the models' risks. Generally, on instance-level analysis, users can select instances of interest to analyze model behavior in other views.

Feature analysis greatly helps model sensemaking. Visualizing the distribution of feature values [3, 7, 10] and the correlation of feature pairs [6] or between the feature and the target variable [6, 7, 45] helps users understand the data better and prepare features for prediction. Per-class feature distribution for classification tasks further enables users to identify discriminative features [29, 43]. For instance, with a heatmap showing covariance between the feature and the target variable, RegressionExplorer [7] allows users to select essential and optional features to build and compare model candidates.

As mentioned earlier, projecting high-dimensional features onto 2-dimensional plots allows users to explore similar instances

Table 1: A review of visualizations for model comparison and selection. In the target user column, D indicates the model developer, while U represents model users. For prediction tasks, C is classification, and R denotes regression tasks. In the analysis of the views, Y indicates the system contains this type of view. In comparison, our system is unique with all aspects combined as well as in some individual aspects described in Section 2.2.

System	Target user	Prediction task	Performance/ Class view	Instance view	Feature view	XAI method	Evaluation
Boxer [10]	D	C	Y	Y	Y	—	Six case studies
ClaVis [12]	D	C	Y	Y	—	—	Three case studies
ComDia+ [33]	D	C	Y	Y	—	—	—
ConfusionFlow [13]	D	C	Y	—	—	—	Three case studies
ConfusionVis [39]	U	C	Y	—	—	—	Two case studies & A user study
DFSeer [36]	U	R	Y	Y	—	—	Two case studies & Four interviews
LEGION [6]	D	R	Y	Y	Y	Y	Two case studies
Li et al. [22]	U	R	—	Y	Y	Y	A case study
LoVis [45]	D	R	Y	—	Y	—	A case study & A user study
Manifold [43]	D	C/R	—	Y	Y	—	Two case studies & Ten interviews
ML-ModelExplorer [40]	U	C	Y	—	—	—	A case study & A user study
ModelWise [29]	D	C	Y	Y	Y	Y	Two case studies
Partition-based framework [32]	D	R	Y	—	Y	—	A case study & A field study
RegressionExplorer [7]	D	C/R	Y	—	Y	—	Two case studies
SliceTeller [44]	D	C/R	Y	Y	Y	—	Three case studies & interviews
StackGenVis [3]	D	C/R	Y	Y	Y	Y	A case study & Three interviews
Our system	U	R	Y	Y	Y	Y	A case study & A user study

[3, 12, 29, 36]. Utilizing XAI methods, projecting the feature importance of all instances onto a 2-dimensional view reveals the structures in the model behavior, that is, how models treat the instances differently or similarly [4, 22, 29]. Colorcoding the instances by models, the projection allows users to explore the diversity and overlap of models' rationale [22]. With XAI methods, two systems, StackGenVis [3] and LEGION [6], showed each feature's overall contribution to the model prediction for feature selection. Two systems visualized the distribution of feature importance of selected datasets [22] or models [29]. As an example, ModelWise [29] devised a violin plot to depict feature importance distribution by models and classes to, e.g., identify high and low effect features. Further, Li et al. [22] correlated feature value and contribution in 2D plots to help inspect the consistency in the models.

With a different goal, several systems facilitate performance analysis under 1- [45], 2- [32], multi-dimensional [44], or hierarchical [7] data partitions to help users understand models' local performance. Of the 16 systems, only four had a formal study (controlled lab studies [39, 40, 45] or a field study [32]) to evaluate how well the system realized its purposes. Others had case studies to demonstrate the probable use of the systems or conducted interviews with target users for feedback.

Our system differs from prior systems in various aspects, individually and collectively: 1) The target users of VMS are model users who intend to choose a regression model to use for their dataset. Of the four systems targeting model users, two are application-specific [22, 36], and the other two are for selecting classification models [39, 40], while our system is for comparing any regression models. 2) VMS provides model comparison and analysis at all three levels, supporting performance, instance, and feature analysis. Three of the surveyed systems [3, 6, 29] support all three levels but are designed for model developers; the resulting visualization's effectiveness is not evaluated with target users. 3) At the instance level, VMS uses the same scatterplot view as Manifold [43] to compare model pairs. However, VMS also allows users to select critical features to color the dots by feature values to examine how feature values correlate to the predictions to understand model behavior. 4) Utilizing XAI methods, we devised an integrated view directly relating feature value and contribution in overlaid layers to facilitate visual inspection of hundreds of features for model sensemaking. Of the 16 surveyed systems, only one explicitly correlates feature values and contributions, using 2D charts [22], which can visualize many cases but not many features at once. Since users understand model behavior through their features, we prioritize an overview of the features. 5) A controlled lab study validated VMS' usefulness in helping understand the model rationale for model selection.

3 PROBLEM CHARACTERIZATION

To create an interface that enables model users to choose optimal models, a visualization designer and two ML experts, who are also authors of this article, closely collaborated and iterated on the prototype on a weekly basis over the course of four months. We used ICU monitoring data as the example case when designing, aiming to help users choose a regression model that predicts patients' ICU LoS. The two ML experts created predictive models for the example case and were considered substitutes for potential end-users during the visualization design process. We had weekly virtual meetings to evaluate and iteration on the prototype, discuss requirements as

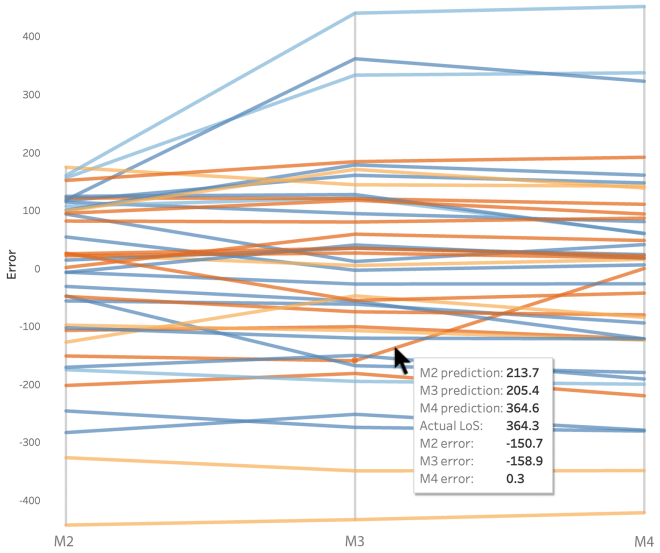

Figure 2: Design alternative — Parallel coordinates to compare multiple models' predictions.

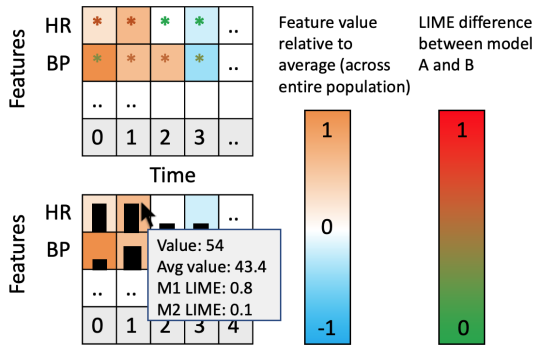

Figure 3: Design alternatives of the feature view showing features in a matrix. Table cells depict feature value and the two models' feature contribution differences in overlaid layers. The two alternatives use the color of the dots and black bars, respectively, to show the differences between the two models' feature contributions. HR and BP denote example features of heart rate and blood pressure.

model users with the goal of selecting an optimal model, and draw alternative designs together with editing tools, such as PowerPoint. In the end, we distilled the following six design requirements:

R1: Compare the overall model performance. Overviewing model performance is straightforward yet essential for any interfaces that aim to support model comparison. As discussed in Section 2.2, users can, e.g., select a metric [29] or view model ranks under multiple metrics [12, 36].

R2: Pair-wise comparison of model predictions and their errors on individual instances. From the model performance view, when two models have similar performances, we suggest users proceed to an instance-level analysis to directly compare individual prediction cases. A model user suggested having a scatterplot with the dots representing individual prediction cases and the axes depicting two models' predictions. So, we can easily see where the models disagree, indicated by the dots that deviate from the diagonal line. Users can select those cases to explore the feature contribution and assess which one makes more sense. The visualization designer offered an option to compare more than two models using parallel coordinates with each axis indicating the predictions of a model (Figure 2). In this case, horizontal lines indicate consistent predictions across the models, and bent lines expose prediction differences among the models. However, the two model users argued that it looked overly complex and preferred the basic pair-wise comparison and the ease of use of scatterplots, similar statements also in Manifold [43], which uses a scatterplot matrix to compare model pairs' prediction results.

R3: Select a pair of models for a detailed comparison. To support an instance-level comparison (R2), the visualization should support users flexibly selecting any model pairs. To do this, a model user suggested having a matrix indicating pair-wise model similarities, which could simply be the average prediction differences between model pairs. Then, users can click on a cell to select a model pair. As mentioned in Section 2.2, Manifold [43] provides pair-wise model comparison similar to ours in scatterplots, meanwhile visualizes the predictions of a user-selected model versus other models in pairs as small multiples. We instead visualize one selected model pair at a time with the consideration that as humans process information sequentially, comparing a model pair at a time avoids information overload.

R4: Inspect the global and local feature importance of the models. To make sense of model behavior, we found it necessary to use XAI methods to look at the features used and how the models weigh these features in making predictions. For instance, when predicting customer churn, understanding which factors contribute to customers' return could help improve the service. In the case of patients' LoS prediction, we used 14 measures inspected at each hour for 24 hours as features. The main idea of the feature view is to design a compact view that allows users to overview feature values and their importance at a glance rather than having individual views for individual features (e.g., [22]). Meanwhile, users can see whether this feature value affects the prediction negatively or positively. For instance, the often-used 2D charts for LIME and SHAP feature importance depict features on the y-axis, feature importance on the x-axis, and feature values as colors for users to evaluate models' rationale [19]. However, such charts could not display hundreds of features at once, nor do they facilitate the pair-wise comparison of models' explanations.

Upon brainstorming, we boiled down the ideas to using 1) a table to show measurements by hours and 2) two layers of visual encodings in table cells to depict feature value and importance, supporting direct comparison in the same feature space [15]. Figure 3 shows design alternatives as a result of our brainstorm, which encodes the feature value of an instance as color gradients at the back layer and depicts the difference in the model pair's feature contribution as the front layer by color or bar.

Further, global feature importance could be shown in the same feature space interactively. In this case, we can show feature value and importance as averages over the instances, though average feature values are not directly related to feature importance as the local feature explanations are. To conclude, the compacted feature matrix aims to show over 300 features relating the feature importance of a selected model pair to the feature values; aggregated feature value and contribution are shown when no instance or multiple instances are selected.

R5: Show instance similarities based on the features. To help understand what kind of data space users are exploring, depicting instance similarities on a 2D canvas using dimension reduction is frequently seen, as discussed in Section 2.2. Using features as vectors and projecting instances onto a 2D space using such as UMAP [27] and t-SNE [42] reveal instances' similarities spatially and intuitively allow users to select similar cases to inspect.

R6: Enable the above exploration under a subset of the data. Often, users need to check how models function in minority yet critical cases. For instance, situations when the models classify important emails as spam and malignant tumors as benevolent could have severe consequences. Enabling model reasoning under a subset of the data allows users to choose models under various conditions. For example, systems that project instances on 2D charts using dimension reduction enable users to select similar instances for detailed exploration [3, 12, 29, 36]. ComDia+ [33] allows users to select a cell from the confusion matrix to evaluate how other models do on these instances. We suggest subset selection in a more semantic structure, such as using feature distributions [10, 29] and other relevant attributes to explore subsets of interest.

4 DESIGN

To address the design requirements, we propose VMS, a visualization with five interlinked views to support the reasoning and selection of predictive models (Figure 1).

4.1 Model overview

To overview model performance (R1), we encode the models' performance value in a bar chart referring to Mackinlay's visual ranking [25] and visualize the uncertainty for a more informed comparison. To simplify the performance ranking for model users, we do not display/integrate multiple metrics but allow users to choose a metric to use. The right side of Figure 1A shows models' overall performance in blue bars under a user-selected metric with black lines indicating 95% confidence intervals.

As mentioned in R3, we decided to use a matrix to show pair-wise model prediction similarities and enable users to select a model pair. The left side of Figure 1A depicts pair-wise model similarities in a matrix. Similarity values are the average prediction differences between model pairs, which are encoded as bars as well: The smaller the values are, the more similar predictions the two models make. Users can click on a cell to select a pair of models to inspect in Figure 1B & D (R3).

4.2 Prediction and data view

As argued in R2, we use a scatterplot to depict the prediction consistency between a selected model pair. Model predictions are further compared to the ground truth to show errors. Upon selecting a model pair from Figure 1A, Figure 1B shows the two models' prediction errors and values in two scatterplots. Each axis depicts one model; each dot represents one prediction case. The more aligned the dots are on the diagonal line, the more similar predictions the two models make.

To expose instance similarities (R5), Figure 1C is a 2-dimensional projection of the instances from all their feature values using dimension reduction [27]. The three scatterplots are interlinked: Mousing over / selection of the dots highlights the same instance in the other two scatterplots. Hovering over an instance shows the instance's ground truth and prediction errors and values in a tooltip, the same for all three scatterplots (Figure 4F). Upon selection of an instance, Figure 1D & E show the feature information and other relevant attributes of the instance, respectively; a bar chart at the bottom left displays the selected case's predictions and ground truth (Figure 6c). Dragging to select multiple cases, users can see how the cases distribute in other scatterplots and inspect their features on a group level in Figure 1D. The color of the dots in scatterplots is described next.

4.3 Feature view

To relate feature value and importance in overlaid layers (R4), we use two distinct visual channels, encoding feature values using colors and feature importance using position and length as bars. We show models' feature importance as two bars next to each other so that users can inspect the importance of each model as well as the difference between the two models. Figure 1D shows the selected model pair's feature importance relating to feature values. It arranges features used for prediction in a matrix since the example case in Figure 1 uses time sequence features. The cells display feature information in two layers.

The back layer encodes feature values in colors: Red indicates values above the cohort average, while blue denotes values below the average. The front layer of two bars in each cell indicates the feature importance of the two selected models (R4). The top seven features of each model are highlighted in green; users can adjust the number of top features to highlight at the bottom left part of Figure 1. For local feature importance, Figure 6D shows the features of the instance selected from the scatterplot. **Negative** importance indicates the model thinks this feature value **decreases** the prediction value, whereas **positive** importance means the model thinks the feature value **increases** the prediction.

When no case is selected, or multiple cases are selected from the scatterplots, colors represent the average feature values; bars depict the average feature importance of the current (selected) cases. Global feature importance is not directly related to the feature value; that is, we can only say how important this feature is overall to the model in making predictions. Empty cells have zero feature importance.

When clicking on a cell in the feature view, users can see a feature's value distribution across the cases in scatterplots in the same red-blue diverging color schema. For instance, Figure 1B has the feature Glascow Coma Scale (GCS) verbal response at the 22nd hour selected; models seem to overpredict cases when values of

this feature are above average in red and underpredict when the values are below average.

4.4 Filter view

To enable users to explore the dataset in subsets (R6), Figure 1E shows relevant attributes of the instances in bar charts. Besides helping users understand the attributional distribution of the instances, the bars can be used as filters for users to explore instances of interest (R6). The example case of patients' LoS prediction in Figure 1 contains attributes including patients' ICU units, age, ethnicity, etc. Users can click on the bars to filter the prediction cases. Multiple selections in one bar chart indicate OR filtering; multiple selections in different bar charts compose AND filtering.

The bar charts are linked: Once filtered, the blue bars in all attribute charts are updated, showing the attribute distribution of the filtered data, with grey bars at the back showing the cohort distribution (Figure 4E). Other views (Figure 4A-D) are also updated to show the information on the filtered data. Additionally, with a case selected in the scatterplots, red lines in the bar charts highlights the attributes of the selected case (Figure 6E). Supplemental video 1 demonstrates VMS' functionalities.

5 CASE STUDY

Healthcare is an area in which ML can be of tremendous help. As an example, the widespread of electronic health records offers great opportunities for planning health resource allocation and forecasting patients' hospital LoS to help improve services. However, high-stakes decision-making, which can impact people's lives, demands transparent and trustworthy models. VMS, in this case, can fill in the gap. For this case study, we exemplify VMS with ICU monitoring data to allow users to rationalize and select models predicting patients' ICU LoS.

5.1 Prediction task

We used a well-known, freely available ICU database, MIMIC-IV (Medical Information Mart for Intensive Care IV [17]), containing vital data from patients who were admitted to the ICU. Referring to its available benchmarking tasks [11], we chose to predict patients' LoS considering its potential use in resource allocation. The LoS benchmark task uses 17 clinical variables to make predictions. Upon analysis, we removed three of them: GCS total, which strongly correlated to the other three GCSs, eye-opening and motor and verbal response; capillary refill rate, which had plenty of missing values; and height, as it had little variation throughout patients' stay. We used the remaining 14 variables for the prediction. Similar to the benchmark task, we used each patient's first 24 hours' hourly measure of the 14 variables as features, a total of 336 features, to predict patients' LoS by hours as a regression task.

As preprocessing, we filtered patients who stayed for at least 24 hours, aged over 18 years old, weighed between 30 and 180 kilograms, and had no missing data regarding the 14 variables. With 14,753 patients remaining, we used 80% as the training set and the rest 10% each as validation and test sets. We trained four regression models: Decision Tree (DT), Random Forest (RF), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU), and abstracted them as M1-M4 accordingly to ease comparison. For this case study and user study, we used MIMIC-IV's publicly available demo data [16], eliminating the need for user credentials from study participants, which resulted in 40 patients upon filtering. Models were tested with the 40-patient data and visualized in VMS. For each model, we also calculated SHAP feature contributions to be visualized as local and global feature explanations.

5.2 Views

Figure 4A shows the metrics to rank the models, including percentage and absolute errors; the similarity matrix shows the average absolute prediction difference (in hours) between model pairs. Upon selecting a model pair from the matrix (M3 & M4 in Figure 4), users can see instance-level prediction errors and values in Figure 4B. The x-axis depicts M3, the y-axis encodes M4 predictions, and each dot represents one patient. Figure 4C projects individual patients onto a 2D canvas using all 336 features through the widely used technique UMAP [27]; users can see patient similarities through their position proximity on the canvas.

The feature view depicts the 14 measures as rows, 24 hours as columns, and the feature value and importance relations as overlaid layers in table cells (Figure 4D). For the 14 measures, except for GCS scores, which are ordinal, ranging from 1 to 5, indicating no response to normal, others are all quantitative. To display feature values, they are first normalized to 0 to 1 as different measures can have different ranges; the normalized feature value is further compared with the normalized cohort average of this measure. We then encode the processed feature values in red-blue diverging colors with red indicating values above the cohort average of this measure and blue denoting values below the measure's cohort average. For the front layer, local feature importance directly codes SHAP feature contributions as bars, which can be negative or positive (e.g., Figure 6D). Global feature importance uses the feature's average absolute SHAP contributions of the cases under exploration, so there are no negatively valued bars when showing the global feature importance. Hovering over feature cells displays the current measure, hour, feature value (mean for a group of cases), the average measure of this cohort, and the feature importance in a tooltip (Figure 4F).

Patient attributes, including their actual LoS, mortality, ICU units, etc., are used to show patient distributions and filter patients to explore data in subsets (Figure 4E). The prototype is implemented using Tableau and accessible at http://tinyurl.com/5ckfuaeu.

5.3 Usage scenario

This usage scenario is adapted from the cases that happened during the user study. With this tool, a user wants to select a model predicting ICU patients' LoS, specifically focusing on cardiac vascular patients aged between 66 and 70. She filters this dataset by selecting the corresponding bars from Figure 4E. As a result of the filtering, there remain five patients who are all male and left the ICU alive in less than two days; Figure 4A-D now shows the information relating to the five patients.

The model overview shows that M3 and M4 have the lowest percentage error. The model similarity matrix also indicates M3 and M4 are most similar in making these predictions among the model pairs. She selected the two models for comparison. From

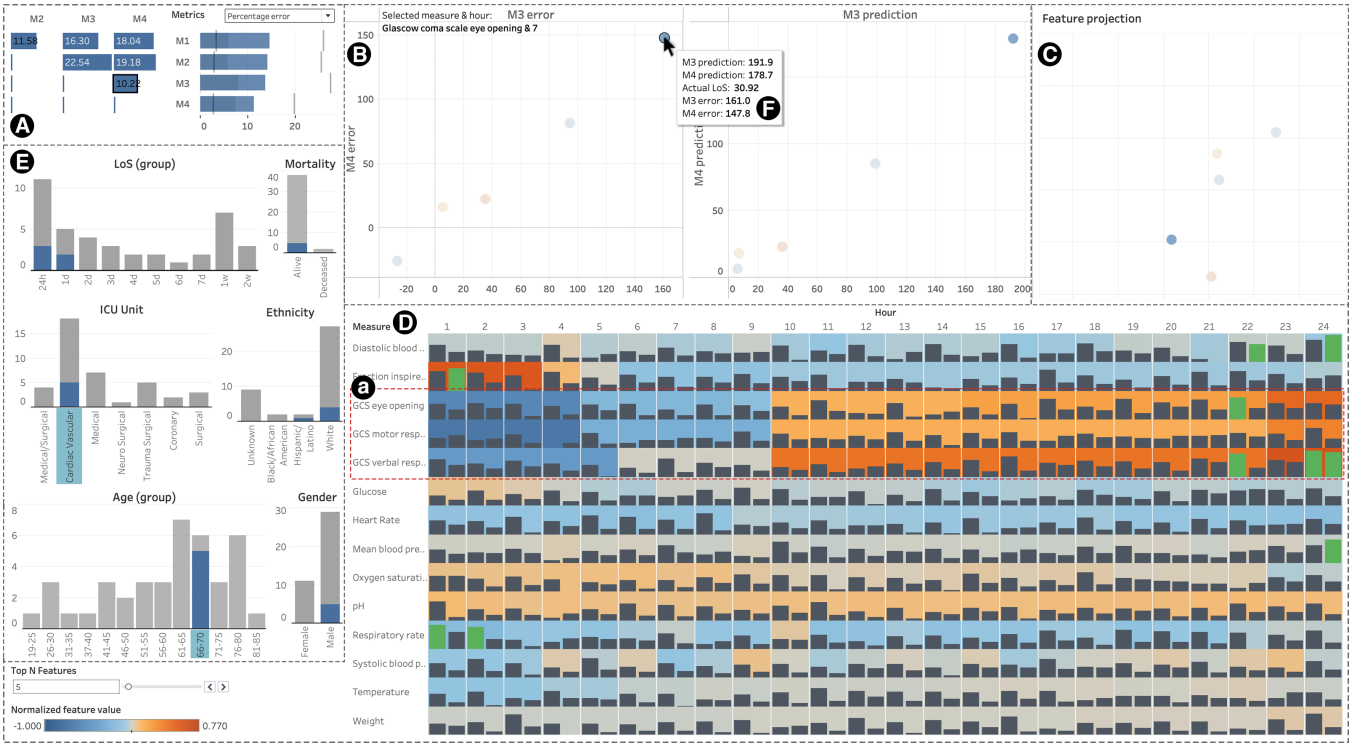

Figure 4: Case study of VMS comparing four ML models (M1-M4) predicting ICU patient's LoS. Upon filtering (E), focusing on cardiac vascular patients aged between 66 and 70, the user selects the two best models (M3 & M4) from (A) to compare their global feature contributions and see which model makes more sense (D). GCS scores transition from blue to red colors, indicating these patients generally went from coma to normal states (a), which makes sense as these patients' actual LoSs are within two days (E). Feature contribution bars indicate how important the features are for the models to predict the current patient set. M3 (bars on the left side of the table cells) generally has higher importance bars than M4 and highlights GCS scores three times and respiratory rate twice, whereas M4 includes mean and diastolic blood pressures as top features.

the prediction error chart, she sees the two models have similar largest errors, around 150-160 hours. The feature chart shows the five patients' average feature value and global feature importance (Figure 4D). GCS scores indicate these patients generally went from a coma state to completely oriented, from blue to red colors (Figure 4a). Fraction-inspired oxygen indicates these patients received high doses of oxygen supply in the first three hours. The user then adjusts the controller at the left bottom to analyze the top 5 important features of the models. The left-side bars represent M3, which generally has higher importance on all features than M4. While M3 highlights GSC scores three times at different hours, M4 spreads top important features more diversely, including fraction-inspired oxygen and blood pressures (green bars on the right side of the cells). She then lists several reasons to choose M4 over M3 for this patient group:

- Overall, M4 has a lower percentage error.
- M3 focuses too much on GCS scores, which are highlighted three times. To cardiac vascular patients, GCS is less relevant than cardiac features such as heart rate and blood pressure.
- M4 highlights more diverse and relevant features like mean and diastolic blood pressures. These features are important to decide this patient group's LoS.
- Fraction-inspired oxygen at the first three hours indicates patients were with machine-assisted breathing, which could affect respiratory rate. In this case, the respiratory rate is unreliable to predict LoS, which M3 highlights twice.

6 USER STUDY

We conducted a controlled study with 16 subjects from the medical field, asking users to analyze model behavior and select models using VMS to investigate the following questions:

Q1 How do users use VMS to understand model rationale?
Q2 How effective is VMS in supporting model selection?
Q3 What do users find positive and negative about the XAI approach provided in VMS?

6.1 Participants

We recruited 16 participants (Age range: 23-41, median: 26.5; Female: 7) from the medical field at one university. Upon completing the study, each one received a 20-euro voucher from a local supermarket chain as compensation. Figure 5 displays their background information. Consistent with their age distribution, the majority (75%) were at the beginning of their medical career (e.g., master's or

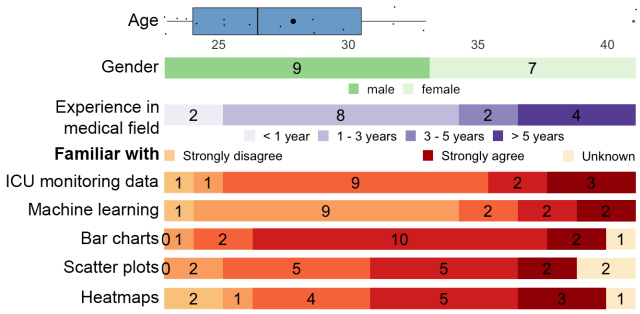

Figure 5: Participants' background, such as their age distribution, gender, and number of years experience in the medical field. We also inquired about their familiarity with ICU data, ML, and visualization techniques in 5-point Likert scales, which is depicted from least to most familiar with color gradients; participants who did not answer (unknown) are in the lightest color category. Except for age, the numbers on the bars indicate the number of participants in that category.

early-stage doctoral students). We inquired about their familiarity with the different aspects that appeared in the study on 5-point Likert scales. Results show they had knowledge about ICU monitoring data (Median: 3), were not so familiar with ML (Median: 2), but were familiar with bar charts (Median: 4), scatterplots (Median: 3.5), and heatmaps (Median: 4). Upon inquiry, 11 of the participants did not use ML in their work; the rest worked with ML on different levels, including creating models (2), using models (2), and providing data (1).

Before the actual study, the experimenter trained the participants with some ML knowledge so they were empowered to use the tool. We believe these participants fit well as target users as they knew about the data and had sufficient ML knowledge for the study tasks.

6.2 Procedure and tasks

The study consists of three stages: an interactive tutorial, three tasks, and a questionnaire & interview. Participants proceeded to the tutorial after signing the consent form, being informed about our data collection and analysis method and agreeing to act as research participants. The interactive tutorial, created using the intro.js library [38], introduces the tool in 19 steps in order of the model overview, filters, feature view, scatterplots, and the interactions between the views. As per instruction, users were prompted to interact with the charts, such as selecting a certain filter and dragging to select multiple dots on the scatterplots. After the tutorial, they could freely explore the tool and ask questions before continuing to the tasks.

Table 2 shows the three tasks and answers participants needed to fill in. The three tasks intended to focus on different parts of the interface, including the scatterplots and the local and global feature views, to explore Q1-3. T1 asked users to select an important feature and see how it relates to models' predictions by analyzing the scatterplot; T2 required users to compare two models by analyzing the feature view of a case where the model pair made similar errors; T3 was to rank the four models for a subgroup of patients by analyzing the scatterplots and the feature view. Though VMS is not created to support model ranking, T3 encourages users to compare multiple model pairs to enrich our data collection. Participants were asked to think aloud during the tasks. To prompt users to rely more on their domain knowledge to reason and select models rather than following the numbers given by the computation, we hid some functionalities of the tool for this study (Figure 6a-c).

The study was conducted with a 13.3-inch MacBook Air with an M1 chip and 16GB memory. The participants had the laptop in front and an external display in front as well above the laptop display. We made sure that the external display could show the whole visualization without the need to scroll. During the tasks, the laptop display showed the online task sheet; each task came with a table structuring the answers that needed to be filled. The experimenter, with another laptop having the same task sheet open, helped fill in the answers when the participants thought aloud and asked for more details if the answer was not clear.

Since they were not seasoned in computer science, during the tasks, the experimenter answered their questions relating to ML and visualization techniques if they had any. After the tasks, they completed a questionnaire regarding their background, followed by an interview on their comments on the tool, the prediction tasks, and the features used (Table 3). The whole study took around one hour.

6.3 Data collection and analysis

During the tasks, we recorded the screen to capture participants' mouse and keyboard interactions and think-aloud voices; the experimenter also noted their answers to the tasks and interview questions. To answer Q1, we analyzed user rationale in solving the tasks. Their answers to model selection/ranking in T2 & T3 were used to answer Q2; we compared the user selection/ranking of the models to machine ranking via performance measure with a larger set of instances. User comments during the tasks and interview were coded to answer Q3.

To code the answers, the experimenter went through the notes to generate the coding schema, using the video recordings to help clarify the notes when necessary. Based on the created coding schema, the experimenter counted the answers. By analyzing the video recordings, participants seemed to have different strategies in answering T3, which were categorized as well. As the coding is straightforward, requiring little interpretation, we did not seek inter-rater reliability [26]. For instance, T1 asked users why they selected a feature as important; the mentioning of green bars or high feature importance implies they replied on machine intelligence, but if they talked from their own judgments, such as "if systolic blood pressure is not stable during the first 24 hours without medication, the patient will stay longer", we say they used their domain knowledge to select. As another example, to code user rationale in model selection for T2 & T3, we count user reasoning such as "M3 has a clear indication on diastolic and systolic blood pressures", "M3 highlights features correctly", and "M4 also focuses on respiratory rate and blood pressure, so M4 is better" as "the model highlights relevant features".

Table 2: Tasks and the structure provided for answers.

T1	For patients who **stayed within 24h**, select the model pair M3-M4; Explore the color patterns in the **three scatterplots** by selecting **important measure & hour combinations** in predicting patients' LoS **based on your knowledge** (Select at least two features).
A1	• Measure & hour combination you have selected. • Why do you choose this combination? • Describe the color pattern (Which view? What discovery?). • Does this pattern make sense? How?
T2	Select a patient that M3 and M2 have similar prediction errors. **Use your domain knowledge** to explain and compare their top 3 important features. Overall, which model do you prefer in this case?
A2	• Case description (patient's LoS, ICU unit, age, etc.). • Measure & hour you analyze: Describe the relation between the feature value and importance, e.g., M4 thinks this feature value (1.0) increases the prediction (6.45). • Does this relation make sense? How? • Overall, which model makes more sense in this case? Why?
T3	Rank the **four** models predicting **cardiac vascular** patients aged **66-70**. To rank the models, use **your domain knowledge** to inspect the scatterplots and the feature view, and write down your discoveries exploring the views.
A3	• Model pair, your discovery, and your preference among the two models. • Overall rank and the rationale.

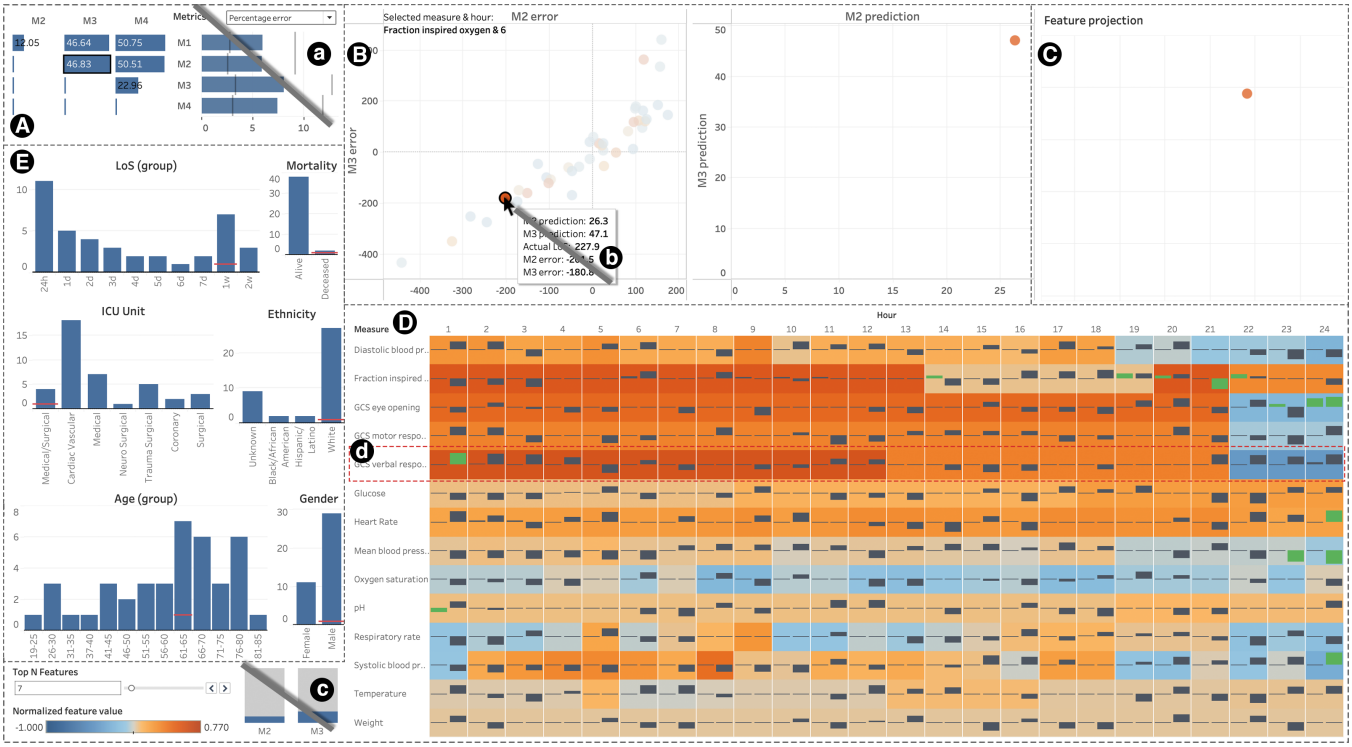

Figure 6: VMS for the user study. Upon selecting an instance from the scatterplot, users can see the background information of this patient indicated by red lines in (E) and the instance's feature values and feature contributions regarding the selected model pair in (D). The highlighted feature row (d) reveals that the feature value can stay the same for several hours, but the feature importance can shift between positive and negative, indicating perceived inconsistencies in model behavior. The model performance metric (a), the scatterplots' tooltip on hovering (b), and the prediction chart of a selected instance (c) were removed to stimulate participants' domain knowledge for model selection.

Table 3: Interview questions.

(1) Do you use machine learning models in your work? (If yes, what tasks do the models fulfill?)
(2) What new knowledge have you learned from exploring the tool?
(3) What are the most important features to predict LoS? In the first 24 hours' stay, what period is most important in deciding LoS?
(4) What functionalities do you want to see but are not in the tool?
(5) What are your suggestions on the prediction tasks and the features used in the prediction?

6.4 Results

This section presents the user study results in order of the tasks and interview comments. Table 4 shows the results of T1. Participants selected important measure & hour combinations to explore the color patterns in the scatterplots. GCS scores, blood pressure, and respiratory rate were the most often selected measures, which is consistent with their interview answers. The last few hours (22-24h) and the first couple of hours (1-2h) were most selected among the first 24 hours, echoing the interview answers as well. Four stated that the first 1-6 hours were critical for treatment based on their domain knowledge.

Often, participants made the selection based on their domain knowledge (9 participants) or the green bars shown in the feature view (7). Two intended to select something different from the first choice, such as a quantitative measure after GCS's ordinal measure. Interested in the progress of the same measure, two selected the beginning and end hours of one measure. Based on the color pattern of the feature values, one selected a red square in the middle of the blue squares.

Since the number of instances for this task was small (11), the charts showed either identical colors in the scatterplots or mixed colors without clear patterns. For instance, one participant selected fraction-inspired oxygen at the 6th hour; the red and blue dots seemed scattered in the scatterplots without a clear indication of the feature's influence on the predictions. The participant stated that it made sense, as "this value in the hospital is not very accurate and can change a lot during 1-6 hours depending on doctors' experience, so it is difficult to use this feature to predict." However, participants provided valuable suggestions on how models can consider these features in predictions, which we discuss later with the interview answers. To conclude T1, **VMS enables users to select important features from multiple perspectives, combining domain knowledge, machine intelligence, feature attributes, and time series exploration to get an overview of how a feature affects models' decisions.**

For T2, the majority of the participants preferred M3 over M2 through the feature analysis of a selected case (Table 5). Nine stated the reason that M3 highlights important features, and three expressed that M2 highlights confusing hours. For example, the feature value stays the same for hours, but M2 highlights twice (2). Three preferred M2, and one chose neither of the models with diverse arguments. For instance, contrary to M3, with M2, the majority of the features have zero importance; one participant preferred this selective consideration with the argument that a "specific combination of features has better predictive power."

A Wilcoxon Signed Rank test shows that M3 was statistically significantly ranked higher than M2 (Effect size: 0.58, $p = 0.022$). With the 40-patient set we studied, M2 showed less percentage error than M3 (M2 error: 5.86, M3 error: 8.02). However, percentage errors with a larger set (1475 instances) show M3 produced less error instead (M2 error: 8.11, M3 error: 4.10) with statistical significance (Effect size: 0.27, $p < 0.0001$). Thus, the feature view enables users to choose a better model with a small sample size. To conclude T2, **the analysis of the relation between feature value and importance on an instance helps users rationalize model behavior and informs model performance on a larger scale with a small subset of instances.**

To answer T3, participants used three different strategies to rank the models: **S1)** seven examined the global feature importance, including the black and green bars of model pairs to select models, among which two combined the examination with the error scatterplots, **S2)** six compared the top global feature importance, only examining the green bars, among which three also assessed the error scatterplots, and **S3)** two evaluated individual patients' feature views since only five instances remained for this task after filtering.

Participants exhibited similar reasoning to T2 in ranking the models. The rationale of model ranking behind S1 & 2 included using more diverse/relevant features and making less error. When these two aspects contradicted, two leaned toward the error chart, and two relied more on the feature importance; one tended to give equal importance to the feature and error charts, for instance, the participant opinioned that M3 uses more features while M2 makes less error, but since the difference in error is small, M3 is better. Highlighting different hours of a measure is considered positive by two and negative by another two participants based on the context. Two adopted S3 and tried to see which value & importance relation does not make sense in model pairs similar to what they did in T2 to eliminate one of the two models.

Figure 7 shows the ranks of the models by the participants in boxplots: The smaller the ranks are, the better the models were considered by the participants. A Friedman test revealed a small but statistically significant difference in the models' ranking (Effect size: 0.18, $p = 0.045$). A pair-wise comparison using Wilcoxon Signed Rank tests showed large and moderate differences in the ranking of M2-4 compared with M1; the differences among M2-4 are small. After the Holm correction, the differences are not statistically significant. Model performance ranking using their percentage errors with 126 instances of the patient set is (from the best to the worst): M4 (2.85), M3 (3.39), M1 (4.99),

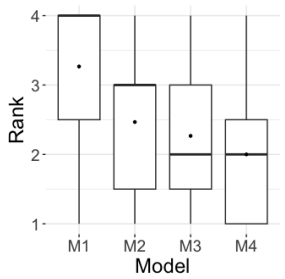

Figure 7: Participants' ranking of the models in T3.

Table 4: T1 results, including participants' selected important features, their rationale behind the selection, and their answers accordingly during the interview. The number in brackets indicates the number of participants in each item.

Selection		Interview		Rationale of the selection
Measure	Hour	Measure	Hour	
GCS (12); Blood pressure (8); Respiratory rate (4); Fraction-inspired oxygen (3); Oxygen saturation (1); Glucose (1)	22-24h (16); 1-2h (11); 6-7h (4)	GCS (14); Blood pressure (8); Respiratory rate (7); Heart rate (3); Oxygen saturation (3)	First few hours (9); Last few hours (3); The whole period (1)	Domain knowledge (9); Green bars (7); Select something different from the first choice (2); Interested in the progress within 24h (2); Color pattern (1)

Table 5: T2 results with participants' overall preference on the model regarding a selected case and their rationale.

Preference	Rationale
M3 (12)	M3 highlights important features (9); M2 highlights confusing hours (3).
M2 (3); Neither (1)	The changes of M3's feature importance are not consistent with the changes in feature values (2)(e.g., Figure 6d); M2 highlights GCSs, which is critical for trauma patients and cannot be intervened while other measures can (1); M2: Specific combination of features has better predictive power (1); M2 does not use important features in prediction (1).

Table 6: T3 results with participants' strategy and rationale when ranking the models. For the first two strategies, user rationale behind model ranking is analyzed together.

Strategy	Rationale
All global features (7) [+ error chart (2)]	The model uses more diverse/relevant (top) features (11); the model has less error (5); the model highlights different hours of the measure (positive) (2); For the same feature value, the model highlights multiple hours (negative) (2).
Top global features (6) [+ error chart (3)]	
Individual instances (2)	Feature value & importance relation makes sense or not.

and M2 (5.15). Except for M3 & M4, the differences in errors between other model pairs are statistically significant after the Holm correction. Users generally choose the better model conforming to the performance analysis, except for M2, which is ranked the lowest based on performance measures but the second lowest by the users. We discuss the complexity of the analysis in the next section. To conclude T3, **three strategies were used to compare models, analyzing global (top) feature importance or local feature importance relating to (average) feature values. User ranking of the models generally conforms to model performance ranking with a larger set of instances, but not without exceptions. The complexity of the analysis needs to be further considered (See Section 7.2).**

During the tasks and interview, participants provided valuable comments on the tool regarding the learning models, features used, and the interface, which we summarize in Table 7. Three applauded the interface as well-integrated. Seven mentioned the usefulness of the feature view, either good to see the changes of lots of vital data (5) or the view highlighted the importance of the features (3). Five requested more training to get familiar with VMS' functionalities before the tasks. As the interface was new to them, nine had no suggestions on the functionalities; others' suggestions centered around the feature view, such as more explanation (2) and manual reordering (1) of the measures.

We categorized user feedback on the models and features. Most feedback requests user control on the weights of the features to improve model predictions. Experts' domain knowledge can be applied by tuning the weights of specific measures, such as lowering the importance of diastolic blood pressure, if one considers this shall contribute less to the prediction, or by conditional settings. For instance, based on user feedback, the tool can allow users to set a condition lowering the weight of mean blood pressure if the value is in the normal range. If fraction-inspired oxygen is higher than 0.7, models can lower the weights of oxygen saturation and respiratory rate in making the prediction; as patients were with machine-assisted breathing, some users consider these values, in this case, not reliable to make the predictions.

However, there could be the potential overuse of domain knowledge, which can hinder models' assumptions about the data. The counter feedback from ML experts is that modeling the data can result in some unrealistic assumptions as a simplification of the complex interactions among features; it is reasonable to accept this unrealisticness, especially if the observed false assumption is not that critical in reality. To conclude user feedback, **VMS well integrates the functionalities to help evaluate models but exhibits a learning curve. As a next step, VMS can allow users to set critical assumptions about the data to improve predictions, but a balance between the use of domain knowledge and machine intelligence shall be promoted.**

7 DISCUSSION

We proposed an interactive visualization, VMS, to support model users to make sense of and select ML models from performance-, instance-, and feature-level analysis. Particularly, 1) in the scatterplots, which compare a model pair's predictions, instances could be

Table 7: Participants' comments regarding the model, features, and the interface collected during the tasks and interview.

Model	• Feature values in the normal range shall not be important, and abnormal values are more important (6), as feature values could be controlled by treatment, such as blood pressure, respiratory rate, and pH, which makes them unreliable to make predictions (2), especially when the values are normal under control. • Value change should indicate importance (6); unstable values, such as local minima, are unimportant (1). • Inconsistency exists between feature value and importance (4). For instance, when the verbal response value stays the same for hours, M3 feature importance can be negative or positive (Figure 6d). • Different patient groups need to consider different features as important (2). • If the value is consistently low, highlighting the last as important makes sense (1). • Two cases can be close in the projection view but have very different predictions; why? (1)
Feature	• Whether the patient is intubated or not is important to consider in prediction, which may affect verbal response, respiratory rate, etc. (3) • Mean/Systolic blood pressure is more important than the diastolic one (3). • Fraction-inspired oxygen, respiratory rate, and oxygen saturation are interlinked (2). For instance, when the patient is with machine-assisted breathing, indicated by the high values of fraction-inspired oxygen, this can affect the respiratory rate and oxygen saturation. • Fraction-inspired oxygen is inaccurate and can change a lot during 1-6 hours based on doctors' experience (1). • Participants recommended many other features to incorporate, such as C-reactive protein concentration (1), natural killer cells (1), blood sodium (1), and cigarettes & alcohol history (1).
Interface	• There is a learning curve; more training is required (5). • The interface is well-integrated and can handle lots of vital data in a short time (3). • More explanation could be provided on the measures, such as how they are measured (2). • It is useful to select a range of hours or combine all hours for the color pattern exploration in scatterplots (2). • It is good to customize the order of the measures considering cultural differences and users' recognition of their importance levels (1).

color-coded by feature values to overview the correlation between the feature and model prediction; 2) the feature view correlating the values and contributions of hundreds of features on either local or global scales allows users to understand and compare the model rationale. We exemplified this method to compare four regression models predicting ICU patients' LoS. A controlled study with 16 participants from the medical field answers the following questions and reveals the significance of VMS:

Q1 How do users use VMS to understand model rationale?
The study witnessed various ways VMS was used to help make sense of the models. First, users can select important features from multiple perspectives, such as using their domain knowledge or machine intelligence, to explore how the feature value relates to model predictions in scatterplots. Then, with the feature view, the analysis of feature importance relating to the feature values in the measure and hour dimensions supports model sensemaking. Example cases are 1) the feature value changes reflected in the color patterns inform users about the interactions among features to help assess feature contributions (An example elaborated in Section 5.3) and 2) the visualized background information of the patient (Figure 6E) facilitates users to use their domain knowledge to identify important features of this patient to evaluate models.

Q2 How effective is VMS in supporting model selection?
Study results provide evidence that reasoning at the instance and feature levels using VMS, users could select the better-performant models without knowing the models' performance with only a small subset of test data, validated by model performance under a larger test set. However, there are uncertainties in rationalization, inviting support to help users make sense of the models in a more systematic and holistic way, which we discuss later.

Q3 What do users find positive and negative about the XAI approach provided in VMS?
Several users considered VMS as well integrated but had a learning curve. Particularly, the usefulness of the feature view stands out from user feedback. Interface suggestions include more explanation on the measures and manual re-ordering of the feature rows to prioritize certain measures. User feedback on models and features suggests the next steps of this research on how to apply users' knowledge of the data to improve model predictions, balancing with machine intelligence, such as adjusting the weights of critical features and conditional settings specifying the interactions among features to steer predictions.

7.1 Implication

With the increased number of open-sourced models, tools allowing non-AI experts to make sense of and select models are in great demand. We expect VMS to be useful in various scenarios: ML developers can use VMS to assess model behaviors with model users, such as domain experts, to gain feedback on how to improve model performance. Model users can custom model selection for decision subjects, such as for cardiac vascular patients. Using VMS requires test samples, including their predictions and feature values & contributions; that is, models need to be pre-trained and their feature importance pre-computed. Next, we discuss limitations and future extensions of this work.

7.2 Model sensemaking with VMS

In T3, users' choice of a better model is not so explicit, as the difference among the models' rankings is not statistically significant.

Moreover, users did not foresee the worse performance of M2 tested with a larger dataset. We give some thoughts on this. Kaur et al. [18] pointed out that in model sensemaking with feature explanations, users might have an unstructured way of exploration to search for plausible rather than accurate explanations that support their personal beliefs. We noticed this type of behavior from users. First, users rationalized in different ways, using varied strategies discussed earlier. Some defects of a model discovered by some users, which lowered their trust in the model, could be overlooked by others. Also, ambiguity could happen in rationalization, which can leave model selection to chances. For instance, M3 & M4 use more features than M1 & M2, which also makes them prone to criticism, such as inconsistency in feature value and importance variations. Facilitating model comparison and selection in a more systematic and holistic manner with increased confidence requires extra design considerations. For instance, we can use computational power to assist rationalization, asking users to input their assumptions with varied weights so the machine can look for visual patterns systematically to help evaluate the models.

On the other hand, bias in ML can occur in various situations [28]. For instance, if a model is trained with a dataset dominated by white males, the model can exhibit bias toward predicting other sex or ethnic groups. ICU data could be biased toward older populations, certain ICU units, etc., which leads to bias in predicting the less represented groups. The design of VMS did not particularly take bias diagnoses into account. Instead, VMS helps users understand model behavior, allowing them to evaluate using their prior knowledge whether the logic of the model makes sense or not. User study results show that VMS triggered user thoughts on model fairness: Several participants suggested that different ICU units need to be trained separately to improve results (group fairness [28]); a user found that instances close in the projection view can have very different predictions (individual fairness [28]).

However, as discussed, users' prior knowledge and their exploration patterns can also induce bias in model selection. We need to consider model sensemaking not as an individualized process but as a process involving social and organizational contexts [18]. For example, different cultures can have different practices and standards in ICU. Moreover, decision subjects' feedback after the adoption of model predictions can change users' view of the prediction and the explanations [18]. Involving multiple users viewing model explanations in different contexts can help model sensemaking with increased resilience.

7.3 Classification

VMS can be relatively easily applied to binary classification tasks; applying to multi-label or multi-class classification tasks requires more adjustments. The model overview would show performance on a metric for classification models, such as accuracy and F1 scores; the similarity matrix can display the percentage of agreements between two models. Upon selecting a model pair, the instance view can adopt the scatterplot design in Manifold [43], depicting instance prediction probabilities by the model pair; each class requires one scatterplot. Upon selecting an instance, the local feature view can remain the same to compare model pairs' feature contribution weights to the predicted class relating to the feature values. The global feature view can have segments on each feature corresponding to the number of classes. Feature values in the segments would show the average (the predominant category for categorical features) of the instances whose ground truth equals the corresponding class, whereas feature contributions aggregate the contribution on the probability of the model predicting the corresponding class, whether true positive or false negative. Users can see how models weigh features differently for predicting the corresponding class. Extending VMS to handle such models is a direction for future work.

7.4 Scalability & Generality

With the increased number of models, Figure 1A will expand: the performance chart will increase linearly, while the similarity matrix will be polynomial. We suggest that VMS can support up to ten models. With more models, there could already be filtering before entering the system, so users only need to compare the best-performing models. For the features, we used time-series data in the case study, visualizing 14 measures assessed hourly for 24 hours, a total of 336 features. If the models have more features, VMS can choose to visualize the features that best differentiate the selected model pair, that is, the features that show the most difference in the model pair's feature contributions.

We believe VMS can be best applied to compare regression models using time-series features, such as forecasting energy consumption using weather data and predicting crop productivity using data from the sensors in the soil. The user study demonstrates how VMS could be used in a real-world case. However, due to the limited number of participants and their background, mostly as junior researchers from one university, results can represent several typical use cases but cannot be generalized to the general population or other application cases. We leave the application and evaluation of VMS in other areas as future work.

8 CONCLUSION

To enable model users to compare and select ML models, this research proposes VMS, a model-agnostic visualization approach that supports performance-, instance- and feature-level model analysis. VMS addresses six design requirements distilled through the close collaboration between the visualization designer and model users. We exemplified VMS to compare four regression models predicting patients' ICU LoS using 14 measures assessed hourly during the first 24-hour stay as features. A user study with 16 model users 1) indicates that VMS allows users to understand the model rationale in various ways, linking predictions, feature values & contributions, and instances' background information, 2) provides promising evidence that through instance and feature analysis with a few test samples, users can select the optimal models, and 3) validates that VMS is accessible to non-AI experts, though several reported a learning curve. Users suggest utilizing domain knowledge in training to improve performance. Research results also showed that biases could emerge in the VDE process resulting from users, data, and models. From the interface design viewpoint, we need to help users uncover bias in the data & models and support model comparison systematically and holistically involving social and organizational

contexts. Future work also includes the extension of VMS to classification models and other application areas.

9 SUPPLEMENTAL MATERIAL

Supplemental video 1 is a video that demonstrates VMS comparing four regression models predicting patients' ICU LoS. (MP4 74,571 kb)

ACKNOWLEDGMENTS

This research is funded by the Strategic Research Council at the Research Council of Finland [Grant Number 358247]

REFERENCES

[1] Dylan Cashman, Shah Rukh Humayoun, Florian Heimerl, Kendall Park, Subhajit Das, John Thompson, Bahador Saket, Abigail Mosca, John T. Stasko, Alex Endert, Michael Gleicher, and Remco Chang. 2019. A User-based Visual Analytics Workflow for Exploratory Model Analysis. *Comput. Graph. Forum* 38, 3 (2019), 185–199. https://doi.org/10.1111/cgf.13681

[2] Angelos Chatzimparmpas, Rafael Messias Martins, Ilir Jusufi, and Andreas Kerren. 2020. A survey of surveys on the use of visualization for interpreting machine learning models. *Inf. Vis.* 19, 3 (2020), 207–233. https://doi.org/10.1177/1473871620904671

[3] Angelos Chatzimparmpas, Rafael Messias Martins, Kostiantyn Kucher, and Andreas Kerren. 2021. StackGenVis: Alignment of Data, Algorithms, and Models for Stacking Ensemble Learning Using Performance Metrics. *IEEE Trans. Vis. Comput. Graph.* 27, 2 (2021), 1547–1557. https://doi.org/10.1109/TVCG.2020.3030352

[4] Dennis Collaris and Jarke J. van Wijk. 2023. StrategyAtlas: Strategy Analysis for Machine Learning Interpretability. *IEEE Trans. Vis. Comput. Graph.* 29, 6 (2023), 2996–3008. https://doi.org/10.1109/TVCG.2022.3146806

[5] The Hugging Face Company. 2016. AutoTrain – Hugging Face. https://huggingface.co/autotrain.

[6] Subhajit Das and Alex Endert. 2020. LEGION: Visually compare modeling techniques for regression. In *Visualization in Data Science (VDS)*. 12–21. https://doi.org/10.1109/VDS51726.2020.00006

[7] Dennis Dingen, Marcel van 't Veer, Patrick Houthuizen, Eveline H. J. Mestrom, Hendrikus H. M. Korsten, R. Arthur Bouwman, and Jarke J. van Wijk. 2019. RegressionExplorer: Interactive Exploration of Logistic Regression Models with Subgroup Analysis. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2019), 246–255. https://doi.org/10.1109/TVCG.2018.2865043

[8] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608 [stat.ML]

[9] Andrew Gelman and Aki Vehtari. 2021. What are the most important statistical ideas of the past 50 years? *J. Amer. Statist. Assoc.* 116, 536 (2021), 2087–2097.

[10] Michael Gleicher, Aditya Barve, Xinyi Yu, and Florian Heimerl. 2020. Boxer: Interactive Comparison of Classifier Results. *Comput. Graph. Forum* 39, 3 (2020), 181–193. https://doi.org/10.1111/cgf.13972

[11] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, and Aram Galstyan. 2019. Multitask Learning and Benchmarking with Clinical Time Series Data. *Scientific Data* 6, 1 (2019). https://doi.org/10.1038/s41597-019-0103-9

[12] Frank Heyen, Tanja Munz, Michael Neumann, Daniel Ortega, Ngoc Thang Vu, Daniel Weiskopf, and Michael Sedlmair. 2020. ClaVis: An Interactive Visual Comparison System for Classifiers. In *International Conference on Advanced Visual Interfaces*. ACM, 9:1–9:9. https://doi.org/10.1145/3399715.3399814

[13] Andreas P. Hinterreiter, Peter Ruch, Holger Stitz, Martin Ennemoser, Jürgen Bernard, Hendrik Strobelt, and Marc Streit. 2022. ConfusionFlow: A Model-Agnostic Visualization for Temporal Analysis of Classifier Confusion. *IEEE Trans. Vis. Comput. Graph.* 28, 2 (2022), 1222–1236. https://doi.org/10.1109/TVCG.2020.3012063

[14] Fred Hohman, Minsuk Kahng, Robert S. Pienta, and Duen Horng Chau. 2019. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Trans. Vis. Comput. Graph.* 25, 8 (2019), 2674–2693. https://doi.org/10.1109/TVCG.2018.2843369

[15] Waqas Javed and Niklas Elmqvist. 2012. Exploring the design space of composite visualization. In *IEEE Pacific Visualization Symposium*, Helwig Hauser, Stephen G. Kobourov, and Huamin Qu (Eds.). IEEE Computer Society, 1–8. https://doi.org/10.1109/PacificVis.2012.6183556

[16] Alistair Johnson, Lucas. Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2022. MIMIC-IV Clinical Database Demo (version 1.0). (2022). https://doi.org/10.13026/jwtp-v091

[17] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data* 10, 1 (2023), 1.

[18] Harmanpreet Kaur, Eytan Adar, Eric Gilbert, and Cliff Lampe. 2022. Sensible AI: Re-Imagining Interpretability and Explainability Using Sensemaking Theory. In *ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 702–714. https://doi.org/10.1145/3531146.3533135

[19] Zoumana Keita. 2023. Explainable AI - Understanding and Trusting Machine Learning Models. https://www.datacamp.com/tutorial/explainable-ai-understanding-and-trusting-machine-learning-models.

[20] Biagio La Rosa, Graziano Blasilli, Romain Bourqui, David Auber, Giuseppe Santucci, Roberto Capobianco, Enrico Bertini, Romain Giot, and Marco Angelini. 2023. State of the Art of Visual Analytics for eXplainable Deep Learning. *Computer Graphics Forum* 42, 1 (2023). https://doi.org/10.1111/cgf.14733

[21] Francesca Lagioia and Giovanni Sartor. 2020. The impact of the General Data Protection Regulation (GDPR) on artificial intelligence. (2020).

[22] Yiran Li, Takanori Fujiwara, Yong K. Choi, Katherine K. Kim, and Kwan-Liu Ma. 2020. A visual analytics system for multi-model comparison on clinical data predictions. *Visual Informatics* 4, 2 (2020), 122–131. https://doi.org/10.1016/j.visinf.2020.04.005 PacificVis 2020 Workshop on Visualization Meets AI.

[23] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence* 2, 1 (2020), 56–67.

[24] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[25] Jock D. Mackinlay. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.* 5, 2 (1986), 110–141. https://doi.org/10.1145/22949.22950

[26] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum. Comput. Interact.* 3, CSCW (2019), 72:1–72:23. https://doi.org/10.1145/3359174

[27] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3, 29 (2018), 861. https://doi.org/10.21105/joss.00861

[28] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2022. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* 54, 6 (2022), 115:1–115:35. https://doi.org/10.1145/3457607

[29] Linhao Meng, Stef van den Elzen, and Anna Vilanova. 2022. ModelWise: Interactive Model Comparison for Model Diagnosis, Improvement and Selection. *Comput. Graph. Forum* 41, 3 (2022), 97–108. https://doi.org/10.1111/cgf.14525

[30] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007

[31] Christoph Molnar. 2022. *Interpretable Machine Learning* (2 ed.). https://christophm.github.io/interpretable-ml-book

[32] Thomas Mühlbacher and Harald Piringer. 2013. A Partition-Based Framework for Building and Validating Regression Models. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 1962–1971. https://doi.org/10.1109/TVCG.2013.125

[33] Chanhee Park, Jina Lee, Hyunwoo Han, and Kyungwon Lee. 2019. ComDia+: An Interactive Visual Analytics System for Comparing, Diagnosing, and Improving Multiclass Classifiers. In *IEEE Pacific Visualization Symposium*. IEEE, 313–317. https://doi.org/10.1109/PacificVis.2019.00044

[34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[35] Ben Shneiderman. 2022. *Human-centered AI*. Oxford University Press.

[36] Dong Sun, Zezheng Feng, Yuanzhe Chen, Yong Wang, Jia Zeng, Mingxuan Yuan, Ting-Chuen Pong, and Huamin Qu. 2020. DFSeer: A Visual Analytics Approach to Facilitate Model Selection for Demand Forecasting. In *CHI Conference on Human Factors in Computing Systems*. ACM, 1–13. https://doi.org/10.1145/3313831.3376866

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[38] Intro.js team. 2020. Intro.js. https://introjs.com/.

[39] Andreas Theissler, Mark Thomas, Michael Burch, and Felix Gerschner. 2022. ConfusionVis: Comparative evaluation and selection of multi-class classifiers based on confusion matrices. *Knowledge-Based Systems* 247 (2022), 108651. https://doi.org/10.1016/j.knosys.2022.108651

[40] Andreas Theissler, Simon Vollert, Patrick Benz, Laurentius Antonius Meerhoff, and Marc Fernandes. 2020. ML-ModelExplorer: An Explorative Model-Agnostic Approach to Evaluate and Compare Multi-class Classifiers. In *Machine Learning and Knowledge Extraction (Lecture Notes in Computer Science, Vol. 12279)*, Andreas Holzinger, Peter Kieseberg, A Min Tjoa, and Edgar R. Weippl (Eds.). Springer, 281–300. https://doi.org/10.1007/978-3-030-57321-8_16

[41] Richard Tomsett, Dave Braines, Dan Harborne, Alun Preece, and Supriyo Chakraborty. 2018. Interpretable to whom? A role-based model for analyzing interpretable machine learning systems. In *ICML Workshop on Human Interpretability in Machine Learning*.

[42] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[43] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S. Ebert. 2019. Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2019), 364–373. https://doi.org/10.1109/TVCG.2018.2864499

[44] Xiaoyu Zhang, Jorge Piazentin Ono, Huan Song, Liang Gou, Kwan-Liu Ma, and Liu Ren. 2023. SliceTeller: A Data Slice-Driven Approach for Machine Learning Model Validation. *IEEE Trans. Vis. Comput. Graph.* 29, 1 (2023), 842–852. https://doi.org/10.1109/TVCG.2022.3209465

[45] Kaiyu Zhao, Matthew O. Ward, Elke A. Rundensteiner, and Huong Ngo Higgins. 2014. LoVis: Local Pattern Visualization for Model Refinement. *Comput. Graph. Forum* 33, 3 (2014), 331–340. https://doi.org/10.1111/cgf.12389