Intuitively Assessing ML Model Reliability through Example-Based Explanations and Editing Model Inputs

Intuitively Assessing ML Model Reliability through Example-Based Explanations and Editing Model Inputs

Harini Suresh
hsuresh@mit.edu
MIT CSAIL

Kathleen M. Lewis
kmlewis@mit.edu
MIT CSAIL

John V. Guttag
guttag@mit.edu
MIT CSAIL

Arvind Satyanarayan
arvindsatya@mit.edu
MIT CSAIL

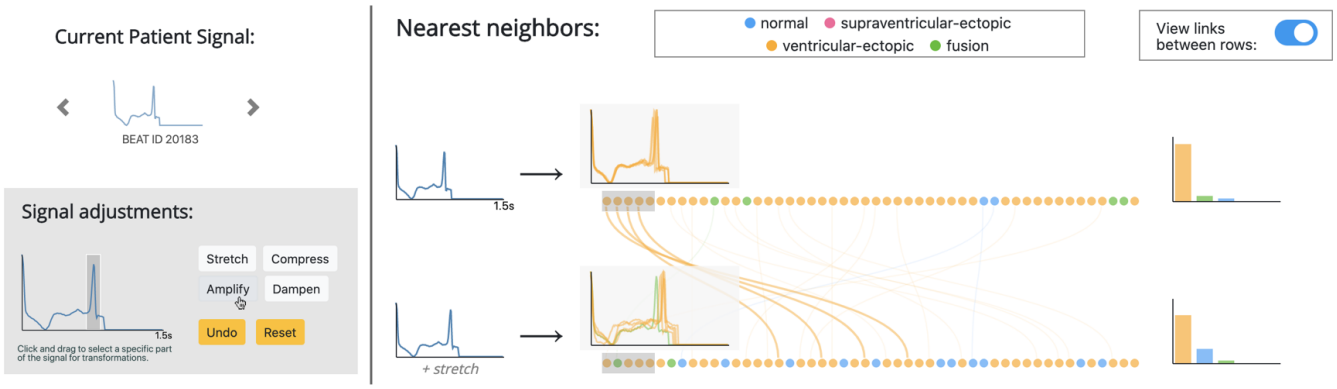


Figure 1: An example of the proposed interface for an electrocardiogram (ECG) case study. The output of the machine learning model consists of raw and aggregate information about the input's nearest neighbors. With the editor in the bottom left, the user can apply semantically-meaningful manipulations to the input and see how the output changes.

ABSTRACT

Interpretability methods aim to help users build trust in and understand the capabilities of machine learning models. However, existing approaches often rely on abstract, complex visualizations that poorly map to the task at hand or require non-trivial ML expertise to interpret. Here, we present two interface modules that facilitate intuitively assessing model reliability. To help users better characterize and reason about a model's uncertainty, we visualize raw and aggregate information about a given input's nearest neighbors. Using an interactive editor, users can manipulate this input in semantically-meaningful ways, determine the effect on the output, and compare against their prior expectations. We evaluate our approach using an electrocardiogram beat classification case study. Compared to a baseline feature importance interface, we find that 14 physicians are better able to align the model's uncertainty with domain-relevant factors and build intuition about its capabilities and limitations.

IUI '22, March 22–25, 2022, Helsinki, Finland

ACM ISBN 978-1-4503-9144-3/22/03.
https://doi.org/10.1145/3490099.3511160

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Interactive systems and tools**; *Information visualization.*

KEYWORDS

interpretability, machine learning, visualization, nearest neighbors, example-based explanations

ACM Reference Format:
Harini Suresh, Kathleen M. Lewis, John V. Guttag, and Arvind Satyanarayan. 2022. Intuitively Assessing ML Model Reliability through Example-Based Explanations and Editing Model Inputs. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3490099.3511160

1 INTRODUCTION

Machine learning (ML) systems are being developed and used for a broad range of tasks, from predicting medical diagnoses [37] to informing hiring decisions [3]. Many are intended to be part of larger sociotechnical processes involving human decision-makers. In these cases, in-domain accuracy is not enough to guarantee good outcomes — the people using a particular system must also understand the model's reliability (i.e., when its predictions should be trusted, both on average and on a case-by-case basis) and modulate their trust appropriately [35, 64]. *Model interpretability*, which is

broadly intended to give insight into how a particular ML model works, can play an important role here.

Many existing approaches to model interpretability, however, require a non-trivial amount of ML expertise to understand, and thus are often only used in practice by ML developers [7]. While tools for developers are certainly needed, the people who will actually deal with model predictions during decision-making are often a distinctly different set of users. Unfortunately, methods that are intended to be simpler and more understandable to such users — for example, reporting feature weights or displaying more information about the model and dataset — have not improved decision-making in experimental studies [11, 36, 43, 59, 70].

In this paper, we introduce two interface modules to facilitate more intuitive assessment of model reliability. First, we use nearest neighbors (NN) to ground the model's output in examples familiar to the user [60]. Alongside the overall distribution of neighbors, a unit visualization depicts individual examples, encoding their class and similarity to the original input according to the model. An interactive overlaid display provides a more raw visualization of the examples for more detailed comparison. Second, we introduce an interactive editor for probing the model. Users can apply transformations corresponding to semantically-meaningful perturbations of the data, and see how the model's output changes in response. Using these modules together, users can iteratively build their intuition about the model's strengths and limitations. By interactively examining individual neighbors, they can investigate questions like whether variation amongst the neighboring examples is expected for the domain, or if it indicates unreliability; whether the commonalities amongst neighbors align with domain knowledge; or whether these neighbors reveal limitations or biases in the data. Similarly, by interactively modifying the model's input, users can pose and test hypotheses about the model's reasoning, checking that its behavior aligns with domain expectations — for example, ensuring that the model is not overly sensitive to small input modifications that should be class-preserving.

These interface components must be tailored to the model's domain — for instance, different data modalities will require corresponding visualizations of nearest neighbors, and the tools the input editor offers must map to domain-specific operations — but the principles that underlie their design are general-purpose. We briefly illustrate how our interface modules can be instantiated in a diverse range of data domains (including natural language passages on Twitter, and image classification with ImageNet and Quick, Draw!) but devote the bulk of our attention to a medical case study of classifying electrocardiogram (ECG) heartbeats with different types of irregularities. This case study allows us to perform an application-grounded evaluation [19] with representative real-world decision-makers who have prior knowledge and investment in the domain. We conducted think-aloud studies with 14 physicians, observing the way they interacted with our interface as well as a feature importance baseline. When working with the baseline, participants often rationalized incorrect predictions — for example, back-tracking on their initial assessment and seeking out things in the input that justified the model's incorrect prediction. In contrast, the NN visualizations help participants grasp prediction reliability — for example, by being able to determine whether variations between neighbors was the result of natural ambiguities in ECG data, or whether it reflected the model not learning the right representations for the task. Moreover, by exploring neighbors from different classes, participants were able to relate the model's uncertainty to clinically-relevant concepts to guide decision-making — for example, pulling out higher-level pathologies that differed amongst neighbors from different classes to understand why the model would be split between those classes. Finally, participants used the input editor to iteratively form hypotheses about the model's reasoning and test them, using the results to investigate how the model worked and whether its reasoning was clinically sensible.

Our proposed interface modules contribute to the growing work on designing human-centered interfaces for ML systems that highlight both model strengths *and* weaknesses, and that encourage critical engagement with the system. We highlight several important design goals to this end, including grounding visualizations in examples familiar to the user, enabling comparison across examples, and allowing interactive probing of the model. Importantly, we align visual components and modes of interaction with users' existing conceptual models of the domain, and show that this facilitates more intuitive understanding of the model and its reliability. Our work suggests several promising directions for future research aiming to improve human-ML interaction, from better conveying data limitations upfront to balancing user input with automated methods when probing the model.

2 RELATED WORK

2.1 Interpretability Methods for Human Understanding

ML interpretability aims to provide information that helps people understand how a model works, either on a global or case-by-case level [25, 28]. Such efforts can serve a number of different goals, such as aiding in decision-making, helping debug or improve a system, or building confidence in the model [32]. A major area of focus has been on developing methodologies for computing and presenting such explanations [17].

Some methods try to visualize the internals of a particular model to reason about how it is operating [15, 49, 78]. This can be useful for theoretical ML understanding and model development, but can be too abstract and complicated to help people without knowledge of such models and how they work. Others try to produce explanations more grounded in the features of the data, such as a ranking of features important for the prediction or a decision-tree approximating the model's logic [20, 48, 62]. A growing body of work that has tried to empirically measure the efficacy of many of these methods has shown that they often do not actually affect or improve human decision-making [2, 11, 36, 43, 59], and in practice are primarily used for internal model debugging [7].

To understand the discord between proposed interpretability methods and their suitability for real-world users, we can draw from well-established theories in cognitive psychology that describe how people think about problems and organize information using different "cognitive chunks" [50]. For example, a physician might think about diagnostic decisions in terms of concepts that are higher-level than individual features, or relate features to each other in more complex ways than independently ranking them by importance. This idea manifests in theories of HCI stating that

effective and engaging interfaces should allow users to view and interact with them in a way that feels *direct* — i.e., the visualizations and interactive mechanisms available to users should align with their cognitive chunks. Specifically, Hutchins et al. [34] describe "the gulf of execution," arising from a gap between the available mechanisms of an interface and the user's thoughts and goals, and "the gulf of evaluation," arising from a gap between the visual display of an interface and the user's conceptual model of the domain. Our aim is to narrow both of these gaps.

To this end, example-based (also referred to as instance-based) interpretability methods, which produce explanations in terms of other input examples, are of particular interest. Research in cognitive psychology and education supports the idea that people often use past cases to reason about new ones when solving problems [1] and that utilizing examples can help people understand complex concepts, build intuition, and form better mental models [60, 61].

Different types of example-based explanations for ML models have been proposed. Many of these are computed *post hoc*, i.e., they are generated after a prediction is made to try and explain that prediction. For example, counterfactual examples [27, 73] use gradient-based methods to generate the closest example(s) to the input that are predicted to be a different class (defining appropriate measures of "closeness" is an open question). Influence functions [41] try to trace a model's predictions back to the data it was trained on, identifying the examples that were most influential to the prediction. Normative explanations [12] present users with a set of training examples from the predicted class. Xie et al. [77] include both counterfactual and normative explanations in the context of radiologic image diagnosis, and find that providing specific examples can help physicians understand model results. There are limitations of these approaches as well; for example, technical constraints make quickly generating influential examples quite difficult in practice [5, 7], hidden assumptions about actionability in counterfactual explanations can be misleading [4], and normative explanations can be confusing when there is intra-class variation [77].

Others compute example-based explanations by modifying the inference process of a trained model to produce predictions based directly on similar training examples. For example, Caruana et al. [16] and Shin and Park [65] use a trained neural network model to improve a KNN classifier, either through using the model to create a weighted similarity function or through computing similarity in the embedding space of the model, respectively. The class label making up the majority of nearest neighbors can be interpreted as the prediction, and the nearest neighbor examples used as an explanation. Recently, Papernot and McDaniel [58] extended this methodology to compute neighbors using embeddings from multiple layers of a neural network, demonstrating additional uses for improving the model's robustness and confidence estimates.

In our proposed interface, we compute neighbors using the method of Caruana et al. [16]; this could be easily extended to calculate neighbors in a weighted input space as in Shin and Park [65], or to use embeddings from multiple layers of the neural network as in Papernot and McDaniel [58]. Similarity could also be calculated with other, domain-specific metrics; e.g., Fang et al. [21] retrieve similar sensor data examples using a symbolic time series representation. Prior work has focused on developing optimal ways for the trained neural network to inform a KNN classifier, implying that the nearest neighbors would then serve as an explanation. Here, we focus on a relatively unexplored part of this claim, investigating how the resultant output should be presented to the user in an interactive interface to narrow the gulfs of execution and evaluation. We explore a specific case study to more clearly define the ways in which this type of example-based explanation can improve trust and understanding for users.

2.2 Interactivity and Visualization for Interpretability

For interpretability to be useful in practice, effectively communicating information to the user is a critical step. In a literature review of interpretability systems and techniques, Nunes and Jannach [57] found that the vast majority of papers presented explanations in a natural-language-based format (e.g., a list of feature weights). Other types of visualizations include simple charts (e.g., bar plots indicating feature importances) [62] or highlighting/denoting sections of the input (e.g., displaying important pixels of an image in a different color or opacity) [43, 69]. With respect to example-based explanations, the visualizations used are often a table of features if the data is tabular [31, 53, 73, 75] or a list of images if the data is image-based [12, 40, 41]. Here, we explore visual encodings that convey more information and allow for more interaction than does merely listing examples.

Other work specifically focuses on visualizations of latent embeddings within a neural network model. Many of these utilize 2 or 3D plots to visualize distance between different examples in the embedding space [8, 29, 47]. Liu et al. [47] additionally visualize examples along 1D vectors corresponding to user-defined concepts, and Boggust et al. [8] provide the ability to compare embeddings of two different models by viewing and interacting with the two plots side-by-side. Particularly relevant to our work, some of the visualizations of text embeddings proposed in [29] aim to display a given word's nearest neighbors in an embedding space. They plot the nearest neighbors as points along a 1D axis that encodes distance, and provide the ability to compare the nearest neighbors across different embeddings.

With respect to interactivity in these interface, prior work has primarily studied using human feedback to modify or filter the information that is shown [13, 39, 42, 67]. Here, our goal is instead to provide users with a way to probe the model and test hypotheses about its behavior. The tool described in Wexler et al. [75] similarly allows modifying the input to observe how a model's output changes, though it is intended primarily for users familiar with ML.

Like these prior works, our approach aims to facilitate understanding by allowing users to visualize and interact with examples from the data. However, while they are primarily intended for general exploration of what a model has learnt, or for uncovering underlying structure in data, the goal of our interfaces is to help users assess the reliability of predictions on a case-by-case basis.

3 INTERFACE MODULES FOR INTUITIVE MODEL ASSESSMENT

We introduce two kinds of interface components for intuitively assessing the reliability of ML models. In Sec. 3.1, we outline the goals

that guide our designs. The proposed modules utilize general ideas that can be customized to different domains, and we illustrate them primarily with a concrete instantiation of an ECG beat classification task introduced in Sec. 3.2. We then describe the visual components of each module: a display of the model's output in terms of an aggregate and an individual-level view of nearest neighbors (Sec. 3.3), and an editor with which users can interactively modify model inputs and observe how the output changes in response (Sec. 3.4). In Sec. 3.5 we walk through specific ways that users can interact with the interface modules to more intuitively assess the model and its predictions. Finally, in Sec 3.6, we briefly sketch instantiations of our approach for two other domains.

3.1 Design Goals

To facilitate intuitive assessment of model behavior, our overarching goal is to narrow the *gulfs of evaluation* and *execution* for the users of our interfaces [34]. Drawing from prior work in cognitive and social psychology [24, 51, 60, 61], case-based reasoning [1], sociology [35], and HCI [7, 14, 30, 32, 44, 45, 56, 72], we identify several design sub-goals to this end:

- **G1. Ground visualizations in examples.** To narrow the gulf of evaluation, the visual components of our interface should facilitate reasoning that aligns with users' existing conceptual models. We draw from research suggesting that reasoning through prior examples can aid in problem-solving [1], understanding, and mental model-building over time [60, 61]. For users who are more familiar with the application domain than the mechanisms of ML models, using examples is likely to facilitate more intuitive reasoning than approaches based on model components or individual features (consider reasoning about anatomical structures in an x-ray versus individual pixels). Therefore, we aim to use examples as the building blocks of our visualization.
- **G2. Facilitate comparisons across examples.** To further facilitate interaction aligned with users' existing modes of thinking, we are motivated by literature suggesting that *contrastive* reasoning (i.e., reasoning based on what makes a particular case different than similar cases) is an important way that people understand and explain things [46, 51]. Building on this, we aim to make it straightforward for users to compare specific examples in terms of meaningful high-level concepts in the data, enabling them to build understanding with contrastive reasoning.
- **G3. Visualize distributions over predicted classes.** Often, the output of ML-based systems consists only of a single predicted class, which may convey a false sense of certainty and prompt over-reliance, as some studies have found [24, 44]. Conveying model uncertainty can help users align model behavior with their understanding of inherent challenges or ambiguities in the task [14, 72]. Indeed, research on human trust suggests that in addition to conveying assurances of certainty, acknowledging when systems are *un*certain is also an important factor in building effective trust [35]. Providing a probability score along with the prediction is one way to convey uncertainty, though understanding how to interpret abstract probability values is itself challenging for people. Instead, we aim to visualize the output from the model as a distribution over classes at multiple levels of granularity. For example, visualizing an overall probability distribution alongside the specific examples belonging to each class may help users better grasp the sources of the model's (un)certainty and reconcile it with their own understanding of the task.
- **G4. Enable interactive probing of the model in terms of domain-relevant concepts.** Prior work interviewing ML stakeholders has found that one way to build trust is to provide users with ways to confirm that the model is using sensible logic that aligns with their expectations [7, 14, 32, 45, 72]. To facilitate this process, we are motivated by the call to design for "contenstibility," i.e., to make questioning and probing the model an integrated part of the system, rather than an "out-of-band activity" [30, 56]. Interactive capabilities for exploring and querying the model can encourage this kind of engagement — prompting a back-and-forth process where users develop hypotheses and test them, confirming that the model's behavior aligns with their domain knowledge or uncovering unexpected issues. To minimize the gulf of execution, it is also important that users can form such queries in terms of domain-relevant and semantically-meaningful concepts.

3.2 ECG Beat Classification Case Study

Our design goals and proposed interfaces are general-purpose and intended to be adapted for different domains. Here, we present a specific case study, classifying electrocardiogram (ECG) beats, to concretely instantiate and evaluate our ideas. This task allows us to perform an application-grounded evaluation of our system using a realistic task that people (i.e., physicians) are familiar with [10, 19]. ECG beat classification, in particular, is an area where machine learning has been widely applied and yielded good performance [38, 63, 79].

The specific task we implement is classifying a single ECG heartbeat into one of four categories: normal, supraventricular ectopic, ventricular ectopic, or fusion. The latter three classes are different types of arrhythmias, or heart rhythm problems. We use a preprocessed version of the MIT-BIH Arrhythmia Dataset [52] available on Kaggle [22]. Each sample in the dataset is an individual heartbeat sampled at a frequency of 125 Hz, and padded to a maximum length of 1.5 seconds. The available dataset contains a fifth class, "unknown," which we exclude here.

We replicate the convolutional neural network (CNN) classification model from Kachuee et al. [38]. We do not use data augmentation since we are interested in seeing whether our visualizations can elucidate that certain classes are underrepresented. The model was trained for ten epochs on the training set (n = 81,123), resulting in a final overall accuracy of 98.3% on the test set (n = 20,284). The breakdown of classes and performances on each is in Table 1.

3.3 Grounding Model Output in Nearest Neighbors

The NN module displays the model's output for a particular example in terms of its nearest neighbors in the data. The nearest neighbors are computed similarly to prior work [16, 58, 65]: Given

Class	% of Examples	Test Set Accuracy
Normal	89.3%	99.6%
Supraventricular Ectopic	2.7%	70.5%
Ventricular Ectopic	7.1%	95.7%
Fusion	0.8%	70.4%
Overall	–	98.3%

Table 1: Classes used in the ECG beat classification task, along with their distribution in the dataset and the model's test set performance.

a neural network model trained to perform the classification task (the *classification model*), we first define an *embedding model*, whose output is the activations of one of the model's hidden layers (see Figure 2). We use this to embed all the training examples. Then, for a given new input example, we embed it and return the most similar training examples in this learned representation space.

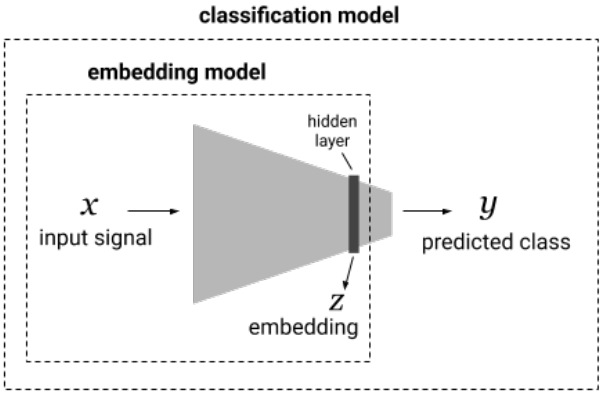


Figure 2: To compute nearest neighbors, we extract an embedding model from the original classification model, where the output is a learned representation (i.e., the activation of a hidden layer). We use it to embed the training data examples and rank them by similarity to the input in this learned embedding space, returning the most similar.

Computing nearest neighbors in the learned embedding space of the classification model provides the advantage of harnessing the classification model's representational capacity. Since this learned space encodes higher level features relevant to the task, these features are taken into account when calculating similar examples. This step is particularly important to our goal of narrowing the *gulf of evaluation* [34] since it provides a way for users to understand the model's output in terms of higher-level concepts that align with how they think about the task. The model output can then be visualized in terms of the nearest neighbors.

Different visual components display the nearest neighbors at varying levels of granularity, which together address our design goals G1, G2 and G3. They include an aggregate view of the neighbors' class labels, a unit visualization of individual neighbors that encodes their class and distance from the input, and a display of the raw input examples associated with each neighbor.

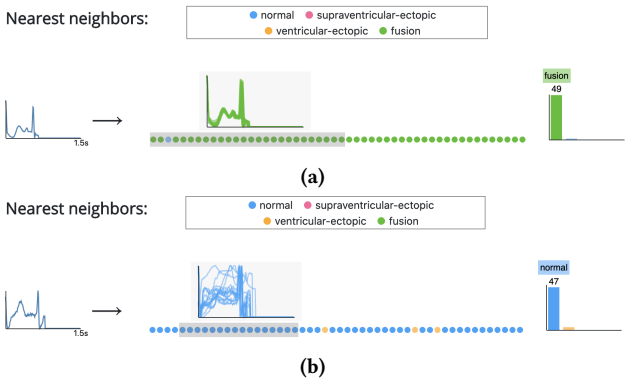


Figure 3: Examples of the NN module. On the left is the input signal, and on the right is a histogram of class labels for the 50 nearest neighbors. In the center, each dot represents an individual nearest neighbor, ordered by similarity to the input. The plot above overlays the signals in the selected region. (a) shows an example where the neighbors are very consistent, and (b) shows an example where they are much noisier.

ECG Case Study. For the ECG beat classification task, we use the CNN classification model described in Sec. 3.2, and we define the embedding model as the output of the activations from the final hidden layer (a 32-dimensional vector). We use Euclidean distance in this space to rank the embeddings of the training examples by their similarity to a particular input. We retrieve the 50 nearest neighbors for visualization.

Figure 3 shows example ECG beats in the interface. Throughout the interface, color encodes class labels (e.g., orange waveforms, dots, and bars correspond to ventricular ectopic examples). The aggregate view is a histogram of class labels present in the nearest neighbors, ordered by class frequency to identify the majority class and distribution of other classes. The exact count of each class appears on hover for each bar in the histogram. The unit visualization of individual neighbors is a series of dots arrayed horizontally and ordered by similarity to the input. Users can see, for example, within the nearest neighbors, if certain classes are more similar to the input. When prototyping this component, we also considered designs that encoded the absolute similarity (e.g., placing two neighbors that were more similar nearer to each other). However, we decided against this, since the absolute similarity (i.e., Euclidean distance in the learned embedding space) is not a value that is meaningful or familiar to the user. Additionally, the distribution of these values is more complicated to visualize, since the distances between neighbors are inconsistent. In our prototypes, for example, there were often clusters of points that densely overlapped and did not facilitate selecting and viewing individual examples.

To visualize the raw input examples, users can brush over specific segments of the ordered dots. The brush is initialized to the first five neighbors, since these represent the most similar examples. Because the ECG data is signal-based, we choose to visualize the neighbors by overlapping signals on a single plot that appears above the brush. This allows users to visually assess consistency amongst the neighbors. If the neighbors are very consistent, the overlaid plot

will look very similar to a single signal; if they are more varied, the overlaid plot will appear comparably noisy. Outliers are also visible, since they appear as a distinct waveform that does not follow in the same pattern as the other signals. By moving and adjusting the brush to cover specific segments of the neighbors, users can home in on and compare examples from specific classes or individual outliers.

3.4 Interactively Editing Model Inputs

To address our final design goal (G4), the editor module allows users to apply meaningful transformations to the input and re-run the modified input through the model to see how the output changes. For example, users can apply transformations that they expect to be class-preserving and check whether the model's output changes drastically.

The available transformations should help narrow the gulf of execution in the interface by providing transformations that align with users' existing ways of thinking about the data and task. For example, in a dataset of photographs, a transformation that inverts colors is not something that would occur naturally and probably does not reflect users' mental models of the domain. We also would not want to provide transformations like editing individual pixels, which operate at a much lower-level than a person looking at an image would consider. To come up with transformations that are data-specific (meaning they reflect how users think about modifying a specific type of data, like images or ECG signals), relevant to the task (meaning they reflect higher-level factors that users consider important to the task at hand), and aligned with the target users' level of understanding, we emphasize the importance of working with domain experts and other intended end users to design them.

ECG Case Study. For the ECG beat classification task, the editor consists of four transformations which we arrived at through discussion with a cardiologist: amplify, dampen, stretch, and compress. These transformations can be applied to the entire input signal, or to specific user-defined regions using the brushing functionality. Together, they allow for a large space of possible adjustments to the input signal. There are other options that could be explored here, such as automatically detecting certain important sections of the signal (e.g., "P wave" or "QRS complex") to transform instead of having users select them themselves.

Once the transformation has been applied, a new row appears below the original output, displaying the new output. The color encoding as well as highlighting on hover enables tracing how the class distribution changes overall, while links between neighbors that are shared across rows enables tracking how individual examples shift in similarity. The editing toolbar is pictured in Figure 4, and an example of the output after several transformations is in Figure 5.

3.5 Enabling an Integrated Workflow

Using the ECG case study, we expand upon several specific ways that a user can interact with the interface modules to assess a model's reliability, understand why it is uncertain, and check whether its reasoning aligns with domain knowledge:

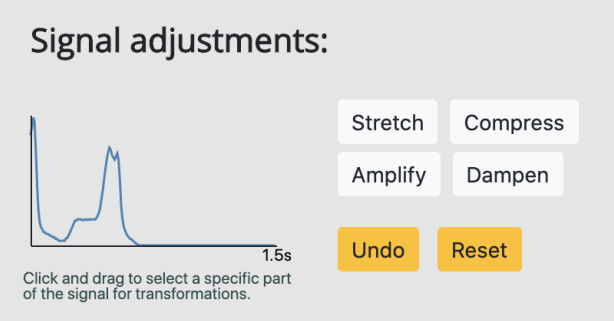


Figure 4: The editing toolbar allows users to apply specific transformations or combinations of transformations to the input signal. The transformations can be applied to the entire signal, or to a specific user-selected region. This allows users to select and transform clinically-meaningful segments of the signal (e.g., "stretch the QRS complex").

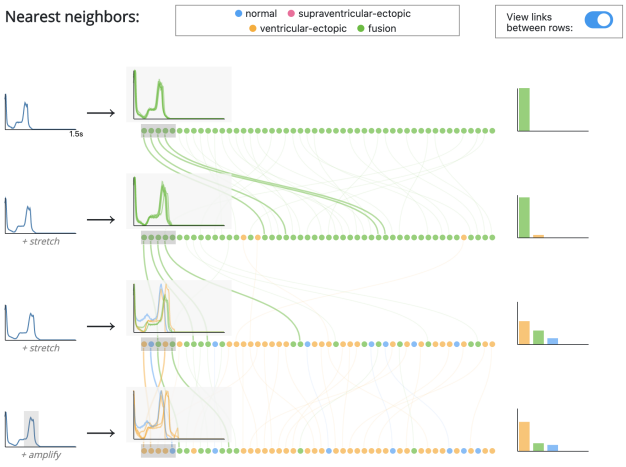


Figure 5: As transformations are applied, new rows appear with the transformed input and corresponding output. Links between each row indicate neighbors that are shared. Links originating from a row's selection are more visible, while the rest are more transparent. Users can get a general sense of how much the nearest neighbors change (by assessing the overall density of links) as well as the specific movements of particular neighbors or sets of neighbors.

3.5.1 Assessing consistency among nearest neighbors to understand prediction reliability and data limitations. Users can assess the reliability of the prediction in multiple ways. First, the aggregate distribution of class labels can convey the model's uncertainty in the prediction (i.e., the majority class label). For example, if 45 neighbors are normal, this conveys more certainty about the prediction than if only 25 neighbors are normal, and the rest are spread out across other classes.

Second, by viewing the class labels of the unit visualization representing individual neighbors, users can see how similar the neighbors from non-majority classes are to neighbors from the

majority class. For example, if there are 40 neighbors labeled normal and 10 neighbors labeled fusion, are those 10 the most similar to the input? Or do they appear closer to the latter end of the nearest neighbors? If the neighbors from the non-majority class are the 10 most similar, this might indicate further unreliability of the 'normal' prediction.

Third, visualizing the variance or consistency amongst the waveforms themselves can give insight into whether the input example is well-represented in the training data and whether the model is picking up on sensible high-level features common in the neighbors. For example, if the overlaid plot of nearest neighbors shows examples that are very consistent and similar to the input in semantically meaningful ways (see Figure 3a for an example), it implies that the input is well-represented in the training data and that the model is picking up on the right concepts for this input. On the other hand, if the plot of nearest neighbor signals shows examples that are non-overlapping or not similar to the input (see Figure 3b for an example), it implies that either examples like the input are not well-represented in the training data, or that the model is not learning the right features and therefore not finding those similar examples.

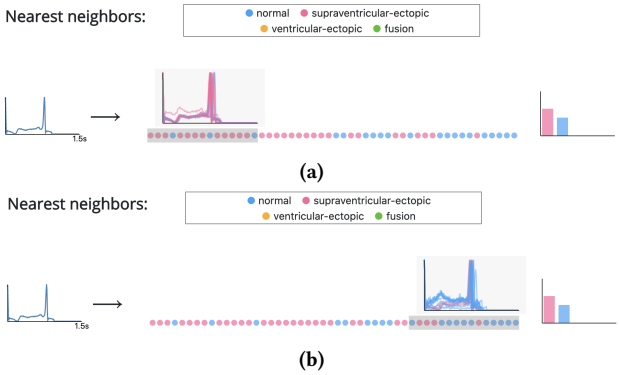


Figure 6: The user can home in on different examples to better understand the model's uncertainty. The view of the first 15 neighbors in (a) suggests that some of the model's uncertainty is arising from the fact that normal beats can look similar to supraventricular beats. Viewing the normal neighbors in (b) suggests that another reason for uncertainty is ambiguity around whether the input has a significant T-wave (the spike at the beginning of the signal).

3.5.2 Investigating neighbors from non-majority classes to characterize prediction uncertainty. Typically, a classification model outputs a probability score indicating its certainty in its prediction. Probability scores can alert the user to some uncertainty in the model, but they don't give the user any additional information to understand *why* the model is uncertain.

In the NN module, one way the model's certainty is conveyed is through the aggregate distribution of class labels. Beyond this, though, the user can further investigate why the model is uncertain by viewing and comparing examples from non-majority classes. Brushing over specific selections of dots representing individual neighbors allows the user to better compare neighbors from different classes. Consider the example in Figure 6: 30 of the neighbors have the class label supraventricular ectopic, and 20 have the label normal (these counts are visible upon hover in the aggregate histogram). In Figure 6a, brushing over the first 15 neighbors reveals that most of them follow the same general pattern and look similar to the input. The 3 normal neighbors in this selection also seem to follow this pattern — so some of the model's uncertainty is arising from the fact that in the training data, there are normal beats that can look similar to supraventricular beats. In Figure 6b, brushing over the last 15 neighbors reveals that most of them follow the same general pattern, but have a more elevated T-wave (the spike at the beginning of the signal) than the supraventricular ectopic neighbors. A user might reason, then, that the model is split between supraventricular and normal, and one of the factors driving the uncertainty is whether or not the input has a significant T-wave.

They could then use their domain knowledge to reason about how to proceed. In this example, they might examine the input and decide that the T-wave is significantly depressed, making the input more similar to the supraventricular ectopic examples, and more confidently proceed with supraventricular ectopic as the correct class. Or, they might decide that the different classes present in the neighbors reflect legitimate ambiguities about what the correct beat type is, and choose to consult a second option or run additional tests.

3.5.3 Comparing examples and labels against domain expectations to prompt critical questioning around the data. If neighboring examples or their labels do not align with the user's expectations, it can prompt questions from the user about the details of the data and how it was collected or labeled, areas that are too often not engaged with after a model's deployment. Crucially, seeing the signals themselves facilitates this type of critical thinking for people who are likely more familiar with the data and what it should look like than they are with concepts like feature weights.

In the ECG case study, for example, the data was annotated by physicians who had access to additional information about the beats preceding and following the input. As a result, there are some examples in the dataset that look extremely similar but are labelled differently (perhaps because of the information available during annotation that the model does not see). In some cases, this leads to nearest neighbors that have different classes but look very similar (see Figure 7). Viewing the neighbors for a particular example can prompt questions about how the data was annotated and the subsequent limitations of the model, which would likely not arise if users were not able to view and compare specific similar examples.

3.5.4 Applying input transformations to check if model reasoning aligns with domain knowledge. Checking if the model's reasoning aligns with prior expectations of domain experts is important for building trust, especially in the clinical domain [14, 72]. The editor module allows users to form hypotheses about how particular transformations should change the model's output, and build confidence and intuition around the model's reasoning by seeing if these hypotheses hold. For example, the beat in Figure 8 is initially classified as supraventricular ectopic. The user might hypothesize

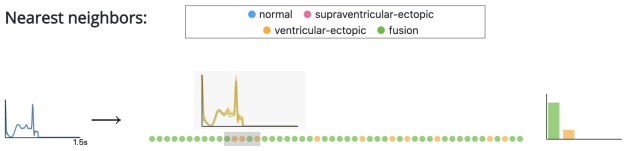


Figure 7: An example of neighbors that look similar but have different labels, caused by a difference in the additional information available during annotation versus at test-time. Alerting users to such cases through viewing nearest neighbors can help prompt questions about the data, the annotation process, and limitations of the model.

that since one indicator of supraventricular ectopic beats is narrowness, and this particular beat is narrow, this is what the model is picking up on. Therefore, stretching the beat should change the model's output, making it shift more towards normal. The user can apply this transformation in the editor to test their hypothesis. In this case, the model's output does change to reflect more normal neighbors, confirming both the original hypothesis and that the model's behavior aligns with the user's expectations from a clinical perspective.

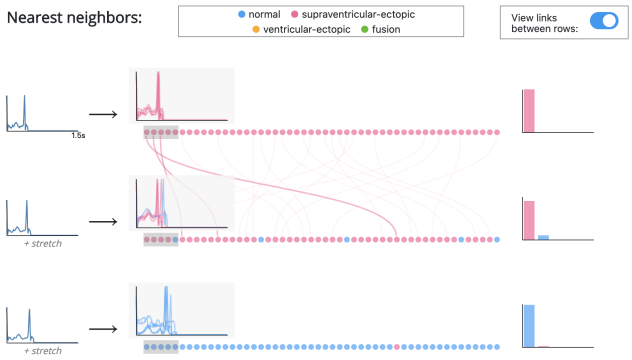


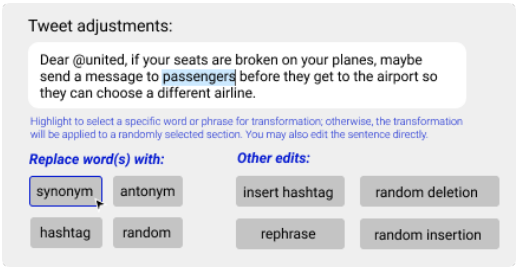
Figure 8: An example of using the editor to check if the model's reasoning aligns with domain expectations (i.e., stretching out a supraventricular ectopic beat should shift the prediction towards normal).

3.5.5 Applying transformations to assess the model's sensitivity to small perturbations. Aside from specific hypotheses about how a particular series of transformations should change the output, a user can gauge the reliability of a particular prediction by performing *ad hoc* sensitivity analyses. If the output changes drastically when the input is slightly tweaked, this can alert users to the fact that the prediction is precarious and encourage them not to be overly reliant on it. On the other hand, if the output is relatively stable, this can be an additional indicator of model reliability.

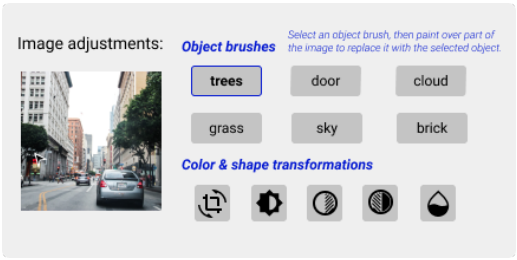
3.6 Instantiations for Other Domains

Although we have focused on the ECG case study thus far, our design goals and interface components are general-purpose and can be adapted for other domains. To do so, one must identify appropriate domain-specific operations to surface in the input editor as well as approaches to visualize and compare nearest neighbors. Here, we briefly demonstrate how our contributions can be applied to two alternate domains: textual passages from Twitter and images from ImageNet [18] and the Quick, Draw! dataset[1].

To identify meaningful transformations for the input editor, we can build on existing work in data augmentation [23, 66] and image generation [6]. For example, for Twitter data, the editor could allow users to edit the text directly, or to apply a range of NLP data augmentations — for example, replacing selected words with synonyms, antonyms, or hashtags. These transformations could be computed using predefined thesauruses, word embedding models, or techniques like back-translation [23]. Depending on the user group, the method of computing augmentations might be predefined, or open to user specification. We show a mockup of an editor for Twitter data in Figure 9a. For image data, on the other hand, users might apply traditional affine or color-based transformations (e.g., rotate, crop, saturate) as well as edit meaningful high-level concepts in the example. For instance, Bau et al. [6] show how activating specific sets of neurons in a generative model can allow users to edit an image with object-level control (e.g., realistically replacing a user-specified section of an image with trees). Drawing from their web-based demo[2], we mock up a potential editor for natural images in Figure 9b.



(a)



(b)

Figure 9: Mockups of the editor module for (a) textual data from Twitter, where edits might consist of replace words, rephrasing the example, or random insertions, and (b) natural image data, where edits could include color and shape transformations or object-level painting as in GANpaint [6].

Similarly, to facilitate comparing NNs and assessing variance, different data modalities will require different techniques. We build on insights from Gleicher et al. [26], who identify juxtaposition,

[1] https://quickdraw.withgoogle.com/
[2] http://gandissect.res.ibm.com/ganpaint.html

superposition, and explicit encodings as fundamental buildings blocks used for visual comparison. While our ECG case study primarily uses superposition (i.e., overlaying signals), instantiations of the NN interface for other data modalities might employ different techniques. For image-based data, for example, side-by-side juxtaposition of examples might be better suited. In Figure 10, we show a screenshot of an interactive prototype we built for the Quick, Draw! dataset (consisting of crowdsourced drawings).

Figure 10: An interactive prototype of the NN interface for the Quick, Draw! dataset juxtaposes neighbors side-by-side instead of overlaying them. As the data consist of line drawings, input editor operations might include drawing, erasing or adding shapes. In this figure, for example, we show how using an "erase" tool to remove inner rings from the input image (an onion) changes the neighbors to almost all blueberries instead, suggesting that the model has learned a correlation between inner circles and the onion class.

Other data modalities might combine visual techniques suggested by Gleicher et al. — for instance, an instantiation of our interface with natural language data might employ both juxtaposition (i.e., viewing examples separately) as well as explicit encodings. WordTree [74], for example, visualizes textual data in a tree-like structure that illustrates commonalities amongst sentences as well as areas of high variance, and Tempura [76] groups sentences with templates that replace specific tokens with abstract linguistic ones. Similarly, Strobelt et al. [68] evaluate a palette of techniques for highlighting text which could be adapted to indicate word-level differences between nearest neighbors.

4 EVALUATIVE STUDIES WITH MEDICAL PROFESSIONALS

To understand how effectively our interface modules help users build intuition for ML model reliability, we return to our ECG beat classification case study as it allows us to conduct an application-grounded evaluation [19] with real-world domain experts and a simplified task. In particular, we recruited 14 participants through our personal and professional networks: 3 fourth year medical students (P1-P3) and 11 physicians (P4-P14). The studies were certified by our institution as exempt from IRB review under Category 3.

4.1 Study Design

In order to study the effect of each of our modules independently, each participant experienced three conditions. The first two conditions were randomly ordered between our NN visualization (without the editor) or a baseline feature-importance visualization, to understand the impact of example-based explanations on building intuition about the ML model. To understand the impact of interactively editing inputs, participants experienced a third condition featuring the NN visualization *with* the input editor. We chose to use feature importance as our baseline since it is a widely researched alternative to example-based explanations [7, 20]. The baseline condition, shown in Figure 11, emulates the design of our NN visualization, and feature importance is calculated using LIME [62], a commonly-used open-source method. In particular, LIME results are shown as highlighted regions that overlay the waveform, in line with existing approaches for visualizing ECG feature importance [55, 71]. We plot the feature importance values that are both above the 80th percentile and part of a continuous segment of neighboring important features, to better align with physicians' existing ways of thinking about regions of an ECG signal.

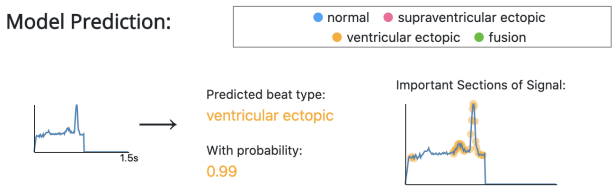


Figure 11: The baseline visualization consists of the predicted beat class, the probability with which that class was predicted, and highlighted segments of the beat considered most important for the prediction.

Each condition was pre-populated with 12 input beats chosen from the test set and equally distributed among the four classes. We select beats such that 30% in each condition have incorrect predictions (for the baseline condition, the prediction is the class with highest probability; for the NN condition, the prediction is the class that makes up the majority of nearest neighbors). These incorrect predictions were aligned with the model's actual performance (e.g., we did not include incorrect predictions for normal beats since there are very few of those; we included more incorrect predictions for supraventricular ectopic since the model's performance for that class is worse).

All studies were conducted via video conferencing. Participants were informed that their participation was voluntary, that they could decline to continue at any point, and that their identities would remain anonymous in any research output. Audio and video was recorded with their consent. The average study length was 52 minutes. Participants were compensated with a $30 gift card.

At the start of each study, participants were told which four categories of beats they would be working with including the granular information about beat types included with the original dataset (e.g., there are multiple pathologies that fall under the umbrella of "ventricular ectopic"). We described that they would see ECG beats

one-by-one, along with output from a machine learning model that had high overall performance. Participants were asked to imagine a scenario where their workplace had adopted such a tool for beat classification, and they were both trying to consider the model's output to make the best decision about a particular beat, as well as get a general sense of how the model worked. We introduced each interface as using a separate model to mitigate participants carrying over preconceptions from prior conditions. For each condition, participants were given a brief demo and were then sent a link to open the interface on their computer and asked to share their screen. We prompted them to click through the beats and, for each one, think out loud about how they were coming to a decision about the beat's class, how they were incorporating the model's output, and whether their perceptions about the model changed. At the end of each condition, we debriefed participants with questions about their general impressions of the model's capabilities, the interface, and the strengths and weaknesses of both.

4.2 Quantitative Results

We recorded the percent of cases in which participants agreed with the model (versus when they disagreed or were not sure). For cases in which the prediction was correct, the agreement rate was similar across conditions; however, when the prediction was incorrect, we found that participants were less likely to accept the model's prediction when they were using the NN interface, with or without the input editor (Table 2). Often in these cases, they did not explicitly "disagree" with the model, but wanted additional information about the signal and/or patient before committing to an answer. We expand on how our interface prompted these additional considerations in the following section.

Pred. Accuracy	*Baseline*	*NN*	*NN + Editor*
Correct	0.64 (0.2)	0.7 (0.16)	0.67 (0.12)
Incorrect	0.73 (0.23)	0.48 (0.27)	0.5 (0.24)

Table 2: The mean agreement rate for correct predictions (8 per condition) and incorrect predictions (4 per condition). The standard deviation across participants is in parentheses.

4.3 Qualitative Observations

After conducting the studies, we rewatched all the video recordings and pulled out relevant quotes or actions by participants. We then iteratively annotated and grouped these quotes by themes using a combined inductive and deductive approach [9]. We find that when using our tools, visualizations of neighboring signals allowed participants to reason about the model's output in terms of clinically-meaningful concepts, and examining variation in these signals helped participants to build intuition about prediction reliability. By inspecting the class histogram, ordering of neighbors, and neighboring signals, participants were able to relate the model's uncertainty to relevant challenges of the task. Finally, participants used the editor to confirm if the model's reasoning was sensible and to guide decision-making.

4.3.1 Nearest neighbors enable reasoning with clinically-relevant concepts. Visualizing nearest neighbors enabled participants to reason about the model in terms of clinically-relevant concepts by generalizing and comparing across neighbors. They would often notice a particular morphology present in the neighbors that helped them understand the model's behavior and whether it was clinically sensible. One participant, pointing to a pattern present in all the neighboring signals, said *"Yeah, ventricular. It's this elevation and this space that's making it think ventricular"* [P4]. Another described, *"The model is right — with ventricular ectopic, the QRS spike should be broad, which is present in all the similar examples"* [P13]. Overall, ten participants [P1, P3, P4-P5, P7-9, P12-14] reasoned about the model using high-level clinically-relevant concepts that they observed in the neighbors, such as *"depression in the signal"* [P13], *"slope right after the P-wave"* [P7], *"presence of a T-wave"* [P8], or *"P-R interval"* [P5].

In some cases, participants were unsure why neighbors were considered similar, or disagreed with their class labels. For example, one participant said, *"these [neighbors] are supposed to be ventricular ectopic... I think they're normal. I don't know what to make of this [output]"* [P2]. Such cases may be partly due to the fact that annotators had access to additional information about surrounding beats during annotation that is not available in the current dataset. Without this information, it can sometimes be unclear why a beat has the class label that it does. While the model's output was confusing in these cases, visualizing neighbors did prompt additional questions about the data and labeling process. For example, one participant asked, *"Some of these normal ones look like they could be abnormal, so I'd want to know why they were called normal and what that was based on"* [P6]. Another further hypothesized, *"Most likely this data was correctly annotated [...] but it's not using all that information here"* [P2].

In contrast, with the baseline condition, participants often had difficulty extracting higher-level, clinically-relevant concepts from the feature importance visualization. For example, echoing a sentiment shared by many, one participant said, *"I don't see how these blue [highlighted] areas are super helpful here... what are they trying to get at?"* [P7]. Another participant, who struggled trying to connect the explanation to the predicted class, said *"I don't understand how they go from this [pointing at highlighted areas] to saying that there's some aspect of a ventricular beat in there"* [P12]. Some others had difficulty figuring out what about the highlighted section was important — for example, one participant asked, *"Why is it highlighted here, is it looking at the height of this, is it looking at width? And why only this part?"* [P1]. In some cases, the highlighted areas *did* align with participants' expectations, though connecting these sections back to the prediction was not straightforward. One participant noted, for example, *"Sometimes it was highlighting things I would also consider, but I still thought its prediction was wrong. I don't have any intuition on that. I guess it's finding some features. I would want to know what those features are, see whether they're useful, if they have any intuitive correlation"* [P2].

4.3.2 Visualizing variation helps assess prediction reliability. All participants said that they did not place as much weight on the model's prediction when there was a lot of variance in the overlaid signals. Participants felt more confident in their answers when the

overlaid signals were very consistent and similar. They were also able to distinguish between variation that was acceptable given the task and domain (e.g., *"This input isn't as picture perfect, so it makes sense that the model shows some variation in the overlaid examples"* [P4]) compared to variation that was an indicator of unreliability (e.g., *"[The model's output] isn't giving me much information right now. If I was given this result I wouldn't just listen to the machine, I would want additional information"* [P4]).

When using the baseline condition, most participants only felt reassured when the predicted probability was very high and the prediction aligned with their own. When this was not the case, we observed that participants had trouble understanding how to incorporate the probability score. As a result, they often rationalized incorrect predictions — even when it went against their initial instincts. For example, one participant saw an abnormal beat, started to say it was abnormal, but then changed her mind after looking at the predicted class, which (incorrectly) was normal: *"I don't think this is normal... well actually seeing that the machine thinks normal... I guess it has a small QRS and the T-wave has a normal slope. Okay, I'll put this in the normal category"* [P7]. Seven participants [P2-4, P7, P9-11] went through similar processes of rationalizing an incorrect prediction after having expressed an inclination towards the correct class.

Even when they did not rationalize an incorrect prediction, participants often struggled with building intuition about the probability score or highlighted sections. For instance, one participant thought out loud, *"I don't know, it seems high probability for a weird looking one like this. And I don't know if it makes sense what it's looking at here and calling important. I'm not confident about this"* [P1]. Similarly, another said *"I'd say this is definitely supraventricular, but the model's not giving it a high probability. I'm really not sure why that would be"* [P11]. Eight participants [P1-2, P5-7, P11, P13-14] expressed similar difficulties in reasoning about the reliability of the prediction in the baseline interface.

4.3.3 Nearest neighbors help characterize uncertainty and incorporate it into decision-making. In the NN visualization, a wide distribution of nearest neighbors classes is one sign of model uncertainty. In such situations, participants consistently homed in on differences using the overlaid plot of waveforms and aligned these differences with clinical concepts. For example, one participant viewed a beat where neighbors were split between supraventricular ectopic and normal, noting *"For supraventricular ectopic one thing you look for is whether or not it has a P-wave. It's unclear in the input. These [brushing over supraventricular ectopic examples] are probably saying it isn't a P-wave. And these [brushing over normal examples] have the P-wave so they're probably saying that the input does also and that's why it should be normal"* [P5].

Similarly, participants often connected the distribution of nearest neighbors to natural ambiguities in the task. For example, one participant noticed some ventricular ectopic beats present in a fusion beat's neighbors — *"Given that fusion is itself a combination of ventricular ectopic and normal, it makes sense that there's uncertainty here, and that there are some yellow [ventricular ectopic] ones that look similar"* [P8]. Rather than distrusting the model, the ability to contextualize its uncertainty helped participants rationalize and move forward with its output. For instance, regarding neighbors

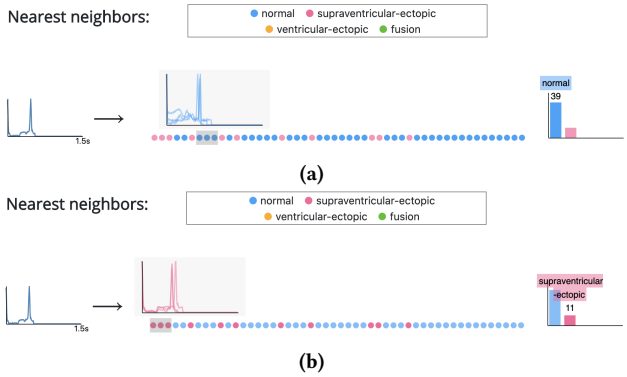


Figure 12: For this beat, one participant looked through some of the normal neighbors (a), comparing them to some of the supraventricular ectopic neighbors (b). They reasoned that the normal examples, though they made up the majority of neighbors, were not more similar in clinically-meaningful ways to the input than the supraventricular ectopic examples. As a result, they were able to arrive at the correct classification (supraventricular ectopic).

split across classes, another participant said *"I would be exactly split like the model is between supraventricular and ventricular ectopic. The fact that the model is also split between those two makes me feel better, and I would do further testing [in person] to differentiate which one it is"* [P4].

Beyond making sense of the presence of multiple classes in the nearest neighbors, participants were also able use this information along with their domain knowledge during decision-making. In many cases, upon viewing neighbors from the different classes, participants would realize that one of the classes was not actually similar to the input and, as a result, feel more confident in disregarding it. For example, for the beat shown in Figure 12, one participant said *"This is supraventricular ectopic. [The model] is calling it normal, but the normal ones don't look so similar. The pink ones [supraventricular ectopic] look more like it because they also don't contain a P-wave"* [P14]. In other words, they were able to relate variation in the neighbors to clinical concepts (normal neighbors with a P-wave, supraventricular ectopic neighbors without), hypothesize why the model is uncertain (it isn't sure whether the input example contains a P-wave), and use their own domain knowledge to determine how to proceed (the input does not actually have a P-wave, so go with supraventricular ectopic). Eight participants went through thought processes to better understand the model's uncertainty and reconcile it with their knowledge of the domain knowledge [P4-8, P10, P13-14].

In contrast, when the model appeared less certain to those using the baseline (i.e., a lower probability score), participants had difficulty reasoning about why. Many said they did not know why the probability was relatively low, or provided explanations based on their own knowledge as opposed to information from the feature importance visualization.

4.3.4 Editing inputs helps check model reasoning. Ten participants used the editor to formulate and test hypotheses about what would

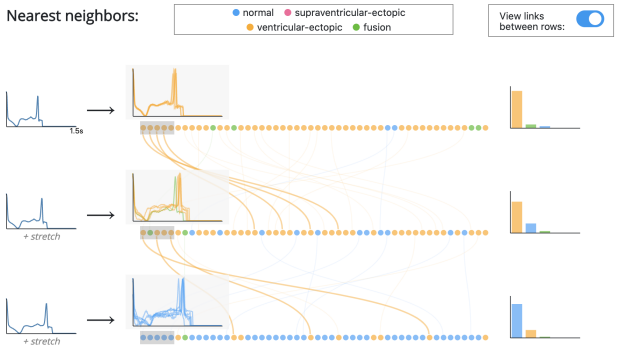


Figure 13: One participant hypothesized that the model was picking up on the narrowness of this beat in giving the prediction of ventricular ectopic, and thus stretching it would cause the neighbors to shift towards normal. After applying the stretching transformation, and seeing that the nearest neighbors did change to be more normal, they felt more confident in the model's reasoning for this beat and in classifying it as ventricular ectopic.

happen to the output after applying certain transformations [P4-9, P11-14]. They used this functionality as a way to "sense check" the model's reasoning, and were more confident if it aligned with their expectations (and vice versa). For example, one participant described using the editor to feel more confident in the model's prediction for a beat (shown in Figure 13), which had mostly ventricular ectopic neighbors: *"I'm not that confident with ventricular ectopic, and this looks almost normal. It's a little narrow, which is partly what ventricular means, so I think that's why this is saying ventricular and if I were to stretch it it would be normal. [Stretches the signal] And that's exactly what happened. That makes me more confident that this is more ventricular ectopic rather than normal. Just because that's exactly what my thought was and that's exactly what happened when I did it"* [P9]. The same participant mentioned later on, *"This is how I think of things. If I can predict what's going to happen I'm more likely to be confident in the decision."*

Sometimes, however, participants applied a transformation but were not able to understand why the nearest neighbors changed as they did, or how to incorporate the observed change into downstream decision-making [P2, P4-5, P8, P10]. This situation typically occurred when the participant applied a transformation that they expected would shift the neighbors towards one of the non-normal beat classes, but instead skewed the neighbors towards normal — a behavior that reflects the model having learned less granular representations of beat classes that were under-represented in the data. On one hand, this unexpected behavior prompted participants to rely less on the model's output in these cases — which, since the model is less accurate for these classes, is appropriate. At the same time, however, these instances were not able to offer participants useful insight into the model's reasoning.

In other cases, participants applied several transformations separately to try and gauge the sensitivity of the prediction to small changes, as a way of assessing model reliability [P1, P3, P6-7, P10-11]. Sometimes several small transformations provided positive reinforcement — *"Okay, this makes me more confident. When it's normal, and then you do all these [transformations], I think it should mostly stay normal, which it is. It's consistent so this all makes sense and I feel good with the machine"* [P1]. Other times, these transformations helped alert participants to the model's unreliability — *"Seeing it switch so quickly from supraventricular ectopic to normal does affect my perception of whether it [the model] is good at telling those apart"* [P3].

With respect to the model's behavior more generally, some participants expressed an increased understanding in how the model worked after using the editor and observing what transformations tended to lead to a large change in the output. One participant noted, *"Doing these transformations is making me think about how this program works... I can tell that the narrowness of a beat affects the decision a lot for example"* [P8]. Participants did not typically use the editor when the neighbors were consistent (both in terms of the shape of the signal and their class labels), because they did not feel the need to check the model's reasoning. Other times, they chose not to use the editor because they could not think of a specific hypothesis they wanted to test — this was particularly true for the participants who were medical students, who often expressed that they "didn't know enough" but that someone with more experience might know what to test.

4.4 Study Limitations

With this study, our focus is on evaluating the proposed interpretability and visualization techniques. Thus, our interface is simpler than something that would be used in a clinical setting — for example, in practice, a physician would typically view a strip of beats from multiple leads, rather than one beat in isolation, and often with a grid overlaid to better measure distances. For our purposes, however, these simplifications follow best practices of application-grounded evaluations [19] and would not materially change our qualitative observations about intuition-building and reasoning with high-level concepts. In some cases, these differences in displaying beats made participants more unsure about a beat's classification — however, this limitation applies equally to the baseline condition, so our quantitative observations about relative accuracies continue to hold.

5 DISCUSSION AND FUTURE WORK

In this paper, we present two interface modules that facilitate intuitive assessment of a machine learning model's reliability. Our work is motivated, in part, by interpretability needs elicited in prior work. For example, studies have found that communicating model limitations and uncertainty is important for building trust [14, 72], but that people have difficulty understanding the meaning of predicted probability scores and incorporating them into decision-making [11]. Other work has described the importance of users being able to "sense check" a model's decision as a way to build trust [7, 33, 45], but there have been few proposed methods or interfaces for doing so. In response, our interface modules are designed to allow users to interactively probe the model and to reason about its behavior through familiar examples grounded in domain knowledge. Users can explore a given input's nearest neighbors in the training data to better understand if and why the model is uncertain,

and what high-level features the model is learning. They can further manipulate the input using domain-specific transformations to test hypotheses about the model's behavior or its sensitivity.

Think-aloud studies with 14 medical practitioners suggest that our interfaces successfully achieve our design goals by helping participants reason about and interact with the model's output in ways that align with their existing conceptual models of the domain. The studies demonstrate how grounding interpretability in real examples, facilitating comparison across them, and visualizing class distributions can help users grasp the model's uncertainty and connect it to relevant challenges of the task. Moreover, by looking at and comparing real examples, users can discover or ask questions about limitations of the data — and doing so does not damage trust, but can play an important role in building it. We also find that our interactive input editor, which offers semantically-meaningful and domain-specific transformations with which to probe the model, provides an effective way for users to sense check the model's reasoning. Importantly, we find that participants in our study described the hypotheses they were testing in terms of higher-level features corresponding to their domain knowledge. In contrast, the baseline — which implemented a commonly-used feature importance method [62] — did not facilitate the same sorts of investigation. We found that this baseline interface demanded a large a mental leap from participants in order to understand how highlighted important sections of the waveform contributed to a high/low predicted probability.

At the same time, our results also point to limitations with the current design of our interface components and suggest opportunities for future work. We find that when the nearest neighbor waveforms looked significantly different than expected, participants had difficulty reasoning about why the model thought the neighbors were similar. We posit that part of participants' confusion was caused by the uneven distribution of beat classes in the training data, which affects the quality of nearest neighbors. For example, supraventricular ectopic beats comprise only 2.7% of training examples; thus, the model was neither able to precisely distinguish this beat from others, nor were there sufficient similar examples to fill the list of neighbors. However, this possibility of under-representation in the training data did not occur to participants when seeing low-quality neighbors. Aside from collecting sufficient data to compute better-quality neighbors, this result suggests the need for transparently communicating the model's training data distribution and its implications. If a user is then presented with output where the neighbors do not appear to make sense, they may be better equipped to understand why this might be the case. Indeed, we found that when we described this phenomena to participants after the conclusion of the study, they were able to understand why under-representation would affect the nearest neighbors — it had just not been on their radar previously. Cai et al. [14] similarly found the need for an *"AI Primer"* for users to explain, in part, *"AI-specific behavior that may be surprising."* Our observations suggest specific use cases of and types of information to include in such a primer.

In other cases, participants found it difficult to apply transformations using the input editor because the space of possible hypotheses was too open-ended. Here, methods that generate counterfactual examples (i.e., similar example(s) that are classified differently) [27, 54, 73] might provide useful inspiration. These methods automatically generate modified inputs by finding small transformations that yield different predictions, but because they do not require any user intervention, they can return unrealistic examples that cannot be probed further. However, such methods could usefully bootstrap our input editor. For example, automatically generated examples could help constrain the space of possible hypotheses to only those transformations that cause the greatest change in the model output. Users could then bring their domain knowledge to bear on selecting semantically-meaningful examples to either visualize directly or as a starting point for further transformation.

Overall, our interface modules and underlying design goals form a promising contribution to the growing body of research on designing ways for end users to contextualize and usefully engage with ML outputs. Our results suggest exciting directions for future work aiming to improve human-ML interaction.

ACKNOWLEDGMENTS

This research was sponsored by NSF Award #1900991, and by the United States Air Force Research Laboratory under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] Agnar Aamodt and Enric Plaza. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7, 1 (1994), 39–59. https://doi.org/10.3233/AIC-1994-7104

[2] Ajaya Adhikari, David M. J. Tax, Riccardo Satta, and Matthias Faeth. 2019. LEAFAGE: Example-based and Feature importance-based Explanations for Black-box ML models. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, New Orleans, LA, USA, 1–7. https://doi.org/10.1109/FUZZ-IEEE.2019.8858846

[3] Ifeoma Ajunwa. 2016. The Paradox of Automation as Anti-Bias Intervention. *Forthcoming in Cardozo Law Review* (2016).

[4] Solon Barocas, Andrew D. Selbst, and Manish Raghavan. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM, Barcelona Spain, 80–89. https://doi.org/10.1145/3351095.3372830

[5] Samyadeep Basu, Philip Pope, and Soheil Feizi. 2020. Influence Functions in Deep Learning Are Fragile. *arXiv:2006.14651 [cs, stat]* (June 2020). http://arxiv.org/abs/2006.14651 arXiv: 2006.14651.

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences* (2020). https://doi.org/10.1073/pnas.1907375117

[7] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. 2020. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*. Association for Computing Machinery, Barcelona, Spain, 648–657. https://doi.org/10.1145/3351095.3375624

[8] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. 2019. Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples. arXiv:1912.04853 [cs.HC]

[9] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[10] Zana Buçinca, Phoebe Lin, Krzysztof Z Gajos, and Elena L Glassman. 2020. Proxy tasks and subjective measures can be misleading in evaluating explainable ai systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 454–464.

[11] Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. 2015. The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems. In *2015 International Conference on Healthcare Informatics*. IEEE, Dallas, TX, USA, 160–169. https://doi.org/10.1109/ICHI.2015.26

[12] Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. 2019. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. ACM, Marina del Ray California, 258–262. https://doi.org/10.1145/3301275.3302289

[13] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. 2019. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[14] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. 2019. "Hello AI": Uncovering the Onboarding Needs of Medical Practitioners for Human-AI Collaborative Decision-Making. *Proceedings of the ACM on Human-computer Interaction* 3, CSCW (2019), 1–24.

[15] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. 2019. Exploring Neural Networks with Activation Atlases. *Distill* 4, 3 (March 2019), 10.23915/distill.00015. https://doi.org/10.23915/distill.00015

[16] Rich Caruana, Hooshang Kangarloo, JD Dionisio, Usha Sinha, and David Johnson. 1999. Case-based explanation of non-case-based learning methods.. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 212.

[17] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* 8, 8 (July 2019), 832. https://doi.org/10.3390/electronics8080832

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[19] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]* (March 2017). http://arxiv.org/abs/1702.08608 arXiv: 1702.08608.

[20] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Commun. ACM* 63, 1 (Dec. 2019), 68–77. https://doi.org/10.1145/3359786

[21] Dezhi Fang, Fred Hohman, Peter Polack, Hillol Sarker, Minsuk Kahng, Moushumi Sharmin, Mustafa al'Absi, and Duen Horng Chau. 2017. mHealth visual discovery dashboard. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. 237–240.

[22] Shayan Fazeli. [n. d.]. *ECG Heartbeat Categorization Dataset.* https://www.kaggle.com/shayanfazeli/heartbeat

[23] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075* (2021).

[24] Susanne Gaube, Harini Suresh, Martina Raue, Alexander Merritt, Seth J Berkowitz, Eva Lermer, Joseph F Coughlin, John V Guttag, Errol Colak, and Marzyeh Ghassemi. 2021. Do as AI say: susceptibility in deployment of clinical decision-aids. *NPJ digital medicine* 4, 1 (2021), 1–8.

[25] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Turin, Italy, 80–89. https://doi.org/10.1109/DSAA.2018.00018

[26] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. 2011. Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309.

[27] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual Visual Explanations. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. Long Beach, California, USA. http://proceedings.mlr.press/v97/goyal19a.html arXiv: 1904.07451.

[28] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–42.

[29] Florian Heimerl and Michael Gleicher. 2018. Interactive analysis of word vector embeddings. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 253–265.

[30] Tad Hirsch, Kritzia Merced, Shrikanth Narayanan, Zac E. Imel, and David C. Atkins. 2017. Designing Contestability: Interaction Design, Machine Learning, and Mental Health. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (Edinburgh, United Kingdom) *(DIS '17)*. Association for Computing Machinery, New York, NY, USA, 95–99. https://doi.org/10.1145/3064663.3064703

[31] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. 2019. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[32] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. 2020. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (May 2020), 1–26. https://doi.org/10.1145/3392878

[33] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. 2020. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–26.

[34] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human–computer interaction* 1, 4 (1985), 311–338.

[35] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 624–635.

[36] Sérgio Jesus, Catarina Belém, Vladimir Balayan, João Bento, Pedro Saleiro, Pedro Bizarro, and João Gama. 2021. How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 805–815.

[37] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. 2017. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology* 2, 4 (2017), 230–243.

[38] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. 2018. ECG Heartbeat Classification: A Deep Transferable Representation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, New York, NY, 443–444. https://doi.org/10.1109/ICHI.2018.00092

[39] Been Kim. 2015. *Interactive and Interpretable Machine Learning Models for Human Machine Collaboration.* Ph. D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA.

[40] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2280–2288. http://papers.nips.cc/paper/6300-examples-are-not-enough-learn-to-criticize-criticism-for-interpretability.pdf

[41] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. Sydney, Australia. http://arxiv.org/abs/1703.04730 arXiv: 1703.04730.

[42] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15*. ACM Press, Atlanta, Georgia, USA, 126–137. https://doi.org/10.1145/2678025.2701399

[43] Vivian Lai and Chenhao Tan. 2019. On Human Predictions with Explanations and Predictions of Machine Learning Models: A Case Study on Deception Detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*. ACM Press, Atlanta, GA, USA, 29–38. https://doi.org/10.1145/3287560.3287590

[44] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[45] Q Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–15.

[46] Peter Lipton. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplements* 27 (1990), 247–266.

[47] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. 2019. Latent space cartography: Visual analysis of vector space embeddings. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 67–78.

[48] Scott Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[49] Pablo Navarrete Michelini, Hanwen Liu, and Dan Zhu. 2019. Multigrid Back-projection Super–Resolution and Deep Filter Visualization. *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), 4642–4650. https://doi.org/10.1609/aaai.v33i01.33014642

[50] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.

[51] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[52] George B Moody and Roger G Mark. 2001. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50.

[53] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In

Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. ACM, Barcelona Spain, 607–617. https://doi.org/10.1145/3351095.3372850

[54] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[55] Sajad Mousavi, Fatemeh Afghah, and U Rajendra Acharya. 2020. HAN-ECG: An Interpretable Atrial Fibrillation Detection Model Using Hierarchical Attention Networks. *arXiv preprint arXiv:2002.05262* (2020).

[56] Deirdre K Mulligan, Daniel Kluttz, and Nitin Kohli. 2019. Shaping our tools: Contestability as a means to promote responsible algorithmic decision making in the professions. *Available at SSRN 3311894* (2019).

[57] Ingrid Nunes and Dietmar Jannach. 2017. A Systematic Review and Taxonomy of Explanations in Decision Support and Recommender Systems. *User Modeling and User-Adapted Interaction* 27, 3–5 (Dec. 2017), 393–444. https://doi.org/10.1007/s11257-017-9195-0

[58] Nicolas Papernot and Patrick McDaniel. 2018. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. *arXiv:1803.04765 [cs, stat]* (March 2018). http://arxiv.org/abs/1803.04765 arXiv: 1803.04765.

[59] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Manipulating and Measuring Model Interpretability. *arXiv:1802.07810 [cs]* (Nov. 2019). http://arxiv.org/abs/1802.07810 arXiv: 1802.07810.

[60] Alexander Renkl. 2014. Toward an Instructionally Oriented Theory of Example-Based Learning. *Cognitive Science* 38, 1 (Jan. 2014), 1–37. https://doi.org/10.1111/cogs.12086

[61] Alexander Renkl, Tatjana Hilbert, and Silke Schworm. 2009. Example-Based Learning in Heuristic Domains: A Cognitive Load Theory Account. *Educational Psychology Review* 21, 1 (March 2009), 67–78. https://doi.org/10.1007/s10648-008-9093-4

[62] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, San Francisco California USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[63] Giovanna Sannino and Giuseppe De Pietro. 2018. A deep learning approach for ECG-based heartbeat classification for arrhythmia detection. *Future Generation Computer Systems* 86 (2018), 446–455.

[64] Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 59–68. https://doi.org/10.1145/3287560.3287598

[65] Chung Kwan Shin and Sang Chan Park. 1999. Memory and neural network based expert system. *Expert Systems with Applications* 16, 2 (1999), 145–155.

[66] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.

[67] Kacper Sokol and Peter Flach. 2020. One explanation does not fit all. *KI-Künstliche Intelligenz* (2020), 1–16.

[68] Hendrik Strobelt, Daniela Oelke, Bum Chul Kwon, Tobias Schreck, and Hanspeter Pfister. 2015. Guidelines for effective usage of text highlighting techniques. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 489–498.

[69] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2020. Visualizing the Impact of Feature Attribution Baselines. *Distill* 5, 1 (Jan. 2020), 10.23915/distill.00022. https://doi.org/10.23915/distill.00022

[70] Harini Suresh, Natalie Lao, and Ilaria Liccardi. 2020. Misplaced Trust: Measuring the Interference of Machine Learning in Human Decision-Making. In *WebSci '20: 12th ACM Conference on Web Science, Southampton, UK, July 6-10, 2020*, Emilio Ferrara, Pauline Leonard, and Wendy Hall (Eds.). ACM, 315–324. https://doi.org/10.1145/3394231.3397922

[71] Geoffrey H Tison, Jeffrey Zhang, Francesca N Delling, and Rahul C Deo. 2019. Automated and interpretable patient ECG profiles for disease detection, tracking, and discovery. *Circulation: Cardiovascular Quality and Outcomes* 12, 9 (2019), e005289.

[72] Sana Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. 2019. What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use. In *Machine Learning for Healthcare Conference*. 359–380.

[73] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology* 31, 2 (March 2018), 841–887. http://arxiv.org/abs/1711.00399 arXiv: 1711.00399.

[74] Martin Wattenberg and Fernanda B Viégas. 2008. The word tree, an interactive visual concordance. *IEEE transactions on visualization and computer graphics* 14, 6 (2008), 1221–1228.

[75] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 56–65.

[76] Tongshuang Wu, Kanit Wongsuphasawat, Donghao Ren, Kayur Patel, and Chris DuBois. 2020. Tempura: Query analysis with structural templates. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[77] Yao Xie, Melody Chen, David Kao, Ge Gao, and Xiang'Anthony' Chen. 2020. CheXplain: Enabling Physicians to Explore and Understand Data-Driven, AI-Enabled Medical Imaging Analysis. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[78] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[79] Muhammad Zubair, Jinsul Kim, and Changwoo Yoon. 2016. An automated ECG beat classification system using convolutional neural networks. In *2016 6th international conference on IT convergence and security (ICITCS)*. IEEE, 1–5.