Explainable AI for Enhanced Medical Diagnostics

Explainable AI for Enhanced Medical Diagnostics

Shreyas Srinivasa*

University of Alabama, Birmingham, ssriniva@uab.edu

Agnivo Neogi

Visvesvaraya Technological University, agnivon@gmail.com

Sudarshan Acula Srinivasalu

Nitte Meenakshi Institute of Technology, iamsudarshan.as@gmail.com

In recent years, the integration of artificial intelligence (AI) into medical diagnostics has ushered in a new era of promise, where accuracy and efficiency in disease detection and prognosis have achieved remarkable heights. AI's capacity to process vast datasets and extract intricate patterns has showcased its potential to revolutionize healthcare outcomes. However, as the reliance on AI-enhanced diagnostic tools grows, an essential concern has emerged: the opacity of AI algorithms in decision-making. This is a pressing concern, particularly in critical medical scenarios where timely and accurate decisions hold the key to saving lives. Numerous studies have illuminated AI's proficiency in diagnosing a diverse range of medical conditions, from radiological image analysis to genomic profiling. Yet, the "black-box" nature of many AI models has impeded their seamless integration into clinical practice. This opacity, where models generate predictions without offering insights into the reasoning process, has led to a trust gap between AI recommendations and medical practitioners. The challenge lies in ensuring that AI's diagnostic prowess is augmented by transparency and interpretability, fostering a harmonious collaboration between machine intelligence and human expertise.

CCS CONCEPTS • Computing methodologies • Machine learning • Machine learning approaches • Neural networks

Introduction

The term "explaining a prediction" refers to the act of delivering textual or visual evidence that offers a qualitative comprehension of the connection between the many elements of an instance (such as words in text or patches in an image) and the prediction made by the model. We contend that the provision of explanations for predictions is a crucial element in fostering trust and promoting the effective utilization of machine learning by humans, provided that these explanations are both accurate and comprehensible. The process of elucidating individual predictions is depicted in Figure 1. It is evident that a physician is significantly more capable of making informed decisions when supplied with coherent explanations in conjunction with a model. In this particular scenario, an explanation refers to a concise compilation of symptoms

* Place the footnote text for the author (if applicable) here.

accompanied by their respective weights. These symptoms may either contribute to the forecast, denoted by the color green, or serve as evidence against it, denoted by the color red. Typically, individuals possess prior information pertaining to the specific field of application, which they can employ to either accept (trust) or reject a forecast, contingent upon their comprehension of the underlying rationale. Previous studies have indicated that the provision of explanations has the potential to enhance the acceptability of movie recommendations [11] and other automated systems [12]. Each machine learning application necessitates a certain level of faith in the model. The process of developing and assessing a classification model often involves the acquisition of annotated data, from which a portion is put aside for automated evaluation. While the pipeline described here is valuable for several applications, it is important to note that evaluating its performance on validation data may not accurately reflect its performance in real-world scenarios. This is because practitioners typically have a tendency to overestimate the correctness of their models [13]. Therefore, it is not advisable to simply rely on validation data for establishing trust in the model. Examining instances provides an alternate approach to evaluating the veracity of the model, particularly when the examples are accompanied by thorough explanations. Therefore, we propose elucidating a selection of exemplary individual predictions generated by a model as a means of offering a comprehensive comprehension. There exist multiple potential sources of error or shortcomings in both the construction of a model and its subsequent evaluation. Data leakage, also known as the inadvertent release of signal into the training (and validation) data that would not be present during deployment, has the potential to enhance accuracy [14]. Kaufman et al. [14] present a notable instance that poses a challenge, wherein the patient identification (ID) exhibits a strong correlation with the target class in both the training and validation datasets. Identifying this issue just through the observation of forecasts and raw data would pose a considerable challenge. However, the task becomes significantly more manageable with the provision of explanations, as exemplified in Figure 1, where patient ID is included as an explanatory factor for predictions. Another challenging issue that can be difficult to identify is known as dataset shift [15], which occurs when the training data differs from the test data (an example of this will be shown later using the well-known 20 newsgroups dataset). The elucidations provided by explanations are especially valuable in discerning the necessary steps to transform an unreliable model into a reliable one, such as eliminating compromised data or modifying the training data to mitigate dataset shift. Machine learning practitioners frequently encounter the task of model selection, which necessitates the evaluation of the comparative reliability of multiple models. In this particular scenario, it is observed that the algorithm exhibiting greater accuracy on the validation set is, in reality, significantly inferior. This observation becomes apparent when explanations are supplied, leveraging human previous knowledge, but remains challenging otherwise. Moreover, it is common to observe a discrepancy between the metrics that can be calculated and improved upon (e.g., accuracy) and the metrics that truly matter, such as user engagement and retention. Although the quantification of these indicators may provide challenges, our understanding of how specific model behaviors can impact them is well-established. Hence, a professional in the field may opt for a model with lower precision in content suggestion, deliberately disregarding

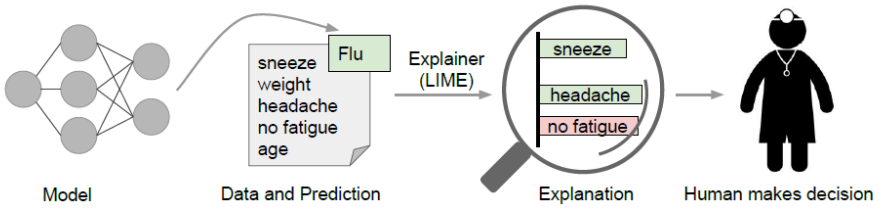


Figure 1: Explaining individual forecasts. The model predicts the patient has "u," and the LIME technique identifies the medical history symptoms that led to this prediction. Sneezing and headache support the theory, but tiredness does not. These criteria can help a doctor assess the model's prognosis. [10]

attributes associated with "clickbait" articles (which could negatively impact user retention). This decision may be made despite the potential improvement in model accuracy during cross-validation by utilizing those features. It is worth noting that explanations are especially valuable in these situations, as they allow for the comparison of various models when a method is capable of generating explanations for any given model.

1 DATA PREPROCESSING

The current scholarly articles demonstrate the methodology of enhancing interpretability and transparency in the use of models, as depicted in Figure 2. Despite the detailed specification of models, datasets, criteria, and outcomes in numerous medical domain publications, it remains necessary to provide explanations and justifications for each individual case. In the coming years, there will be an increased demand for interactive artificial intelligence (AI) systems that offer explainability and facilitate engagement with domain experts. This desire originates from the need to continually enhance outcomes in response to numerous circumstances, including changes in human behavior, weather patterns, and medical problems. Tables 1 to 4 represent potential strategies for managing the corresponding infections or diseases, and are considered suitable for predicting recovery outcomes in a hospital setting. We will examine the preprocessing techniques employed in the recent study, the algorithms implemented in their corresponding models, and the resulting outcomes in this section.

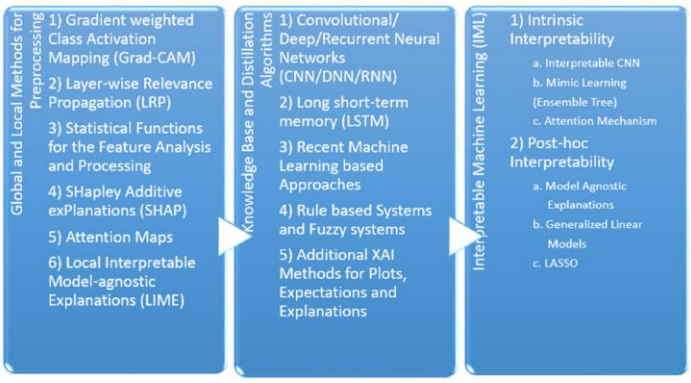


Figure 2: XAI Methods Classification [32].

1.1 Gradient Weighted Class Activation Mapping (Grad-CAM)

The Grad-CAM technique [19] is employed to forecast the corresponding notion by leveraging the target gradients, which are propagated to the final layer. The coarse localization mapping is utilized to identify and emphasize the significant places. The aforementioned technique is recognized as a variation of a heat map, commonly employed in image registration to discern various sizes and scales for predictive purposes. The Grad-CAM method is a propagation technique that offers a straightforward means of visualization and delivers explanations that are accessible to users. Object detection is a widely employed technique that has gained significant popularity, particularly in the medical field, for the purpose of identifying various diseases and affected regions in patients. The application is capable of effectively identifying chest X-rays (CXR), CT scans, brain tumors, and fractures in various human and animal anatomical regions. Given the potential limitations in accuracy when dealing with sensitive domains, several variants of CAM-supported analysis have been proposed. These

include Guided Grad-CAM [16], Respond-CAM [17], Multi-layer CAM [18], among others. The utilization of Guided Grad-CAM involves the assessment of model predictions through the identification of visually prominent characteristics. Therefore, the saliency maps emphasize the relevant properties of the interest class. The technique of combining Grad-CAM and guided backpropagation by pointwise multiplication is commonly referred to as saliency maps in academic literature. The Guided Grad-CAM technique is recognized for its ability to provide maps that are specific to each class. These maps are generated by taking the dot product of the feature map from the last convolutional layers and the neurons, which are then combined to form a projected class score using partial derivatives. The Respond CAM is employed for the manipulation of three-dimensional images characterized by intricate macromolecular structures obtained from cellular electron cryo-tomography (CECT). The Respond-CAM algorithm possesses a "sum to score" characteristic that yields superior outcomes compared to the Grad-CAM method. It is specifically designed to emphasize the class-discriminative regions of three-dimensional pictures by employing weighted feature maps. The sum-to-score property of the Respond-CAM model can be represented as $y^{(c)}$, where $y^{(c)}$ is the class score. Additionally, $b^{(c)}$ represents the last layer CNN parameter, and $\sum_{i,j,k}(L_A^{(c)})$ represents the sum of class c for Grad-CAM/Respond-CAM. Furthermore, C denotes the count of classes as provided in Equation 1. To calculate the conditional probability of the chosen feature, The Multi-layer Grad-CAM method is employed using a single maxout hidden layer. The architecture relies on the utilization of maxout units, which are employed in a solitary hidden layer alongside a softmax function that serves to normalize the output probability.

$$y^{(c)} = b^{(c)} + \sum_{i,j,k} \left(L_A^{(c)}\right)_{i,j,k} \quad (1)[32]$$

1.2 Layer-Wise Relevance Propagation (LRP)

Additionally, it is one of the often-employed techniques for propagating predictions in a neural network, utilizing propagation rules to propagate the prediction in a backward manner. The LRP exhibits versatility in its ability to effectively process various forms of input, including but not limited to photos, videos, and texts. The recording of relevance scores in each layer can be achieved through the use of distinct rules. The LRP methodology is founded upon and supported by the utilization of a deep Taylor decomposition (DTD). The neural network can be configured to operate on either a single layer or a group of layers. By incorporating high-quality explanations, the sophisticated deep neural network can be effectively scaled. Additionally, it is widely utilized within the medical field, encompassing many applications such as chest X-rays (CXR), axial brain slices, cerebral relevance maps, and the detection of anomalies, among others. The various variants for heatmap visualizations in Layer-wise Relevance Propagation (LRP) are LRP CNN, LRP DNN, LRP BiLRP, and LRP DeepLight. The relevance of the Local Relevance Propagation (LRP) method is comparatively higher when compared to other techniques used for visualization and sensitivity analysis. The input representations undergo forward propagation using a convolutional neural network (CNN) until the output is reached. Subsequently, the back propagation process is performed using Layer-wise Relevance Propagation (LRP) until the input is reached. The relevance scores for the categories are obtained by the use of LRP CNN [20]. In the context of the LRP DNN [21], the convolutional neural network (CNN) is optimized by initializing its weights for the purpose of activity recognition based on pixel intensity. The method described in LRP BiLRP [22] systematically decomposes the input feature pairs with similarity scores. The scaling and interpretation of high nonlinear functions is achieved by the utilization of the composition of Layer-wise Relevance Propagation (LRP). The BiLRP offers a similarity model that demonstrates verifiability and resilience for the given situation. The BiLRP is introduced as a composite method that combines different LRP techniques and is afterwards recombined at the input layer. In this context, x and x' represent input variables that are to be assessed for their degree of

similarity. $\emptyset_x$ refers to a collection of network layer components denoted as $\{\emptyset_1 to\ \emptyset_L\}$. Additionally, $y(x, x')$ represents the combined output as defined in Equation (2). The DeepLight LRP (23) utilizes decoding decision decomposition to examine the interdependencies among many parameters at multiple levels of granularity. This methodology is employed to examine the intricate temporal and spatial variations of structures characterized by a large number of dimensions and a limited number of samples.

$$BiLRP(y, x, x') = \sum_{m=1}^{h} LRP([\emptyset_L \circ \ldots \circ \emptyset_1]_m, x) \otimes LRP([\emptyset_L \circ \ldots \circ \emptyset_1]_m, x^x) \quad (2)[32]$$

1.3 Statistical Functions for the Feature Analysis and Processing

The comparison of survivors and non-survivors in terms of categorical variables was subjected to statistical analysis [24]. This analysis was conducted using the chi-square test or Fisher's exact test, and the results were presented in terms of interquartile range (IQR) and standard deviation or medians. The continuous variables were analyzed using either the Mann-Whitney U test or Student's t-test, and the results were reported as frequencies. The Kaplan-Meier method is commonly employed in academic research to visually analyze the association between two variables, accompanied by a log rank test to determine the statistical significance of this relationship. The multivariate Cox proportional hazards model is utilized to assess the impact of risk factors on the result. This model is further examined through the use of a log-log prediction plot. In instances of statistical analysis, a noteworthy p-value is considered to be less than 0.05 for univariate analysis and 0.10 for bivariate analysis. The utilization of the generalized estimating equation (GEE) [25] is employed to illustrate the associations among the sets of features that have been matched. The disparity in occurrence between feature inheritance with GEE matching lies in the changes made to the pre and post data. The Charlson comorbidity index score (Charlson et al., 1987) is employed to assess the impact of comorbidities on the one-year mortality risk of hospitalized patients. This scoring system assigns weights to different comorbid conditions in order to calculate an overall index score. The process of multivariate imputation involves utilizing multiple imputation for post-hoc sensitivity analysis on discrete and continuous data through the implementation of chained equations. The LMS approach, as described in reference [26], is employed for the computation of z-scores representing the usual lower limits of spirometric values. The kappa statistic is a measure of chance agreement, where a value of 1.0 indicates perfect agreement and a value of 0 indicates no agreement. The least absolute shrinkage and selection operator (LASSO) is a technique utilized in regression analysis to enhance prediction accuracy through variable selection and regularization [27]. The issue of imbalanced categorization is commonly addressed by the utilization of Synthetic Minority Oversampling Technique (SMOTE) [28]. Imbalance in datasets is commonly attributed to the presence of minority classes, which are subsequently replicated within the training set prior to model fitting. The act of duplicating class material serves to address the issue of class duplication, although it does not contribute any more knowledge.

1.4 Shapely Additive exPlanations (SHAP)

The SHAP method [29] use ranking-based algorithms to perform feature selection. The optimal characteristic is arranged in a descending order based on SHAP scores. The method is founded on the attributes of magnitude and operates as an additive feature attribution technique. The SHAP framework is a computational approach that utilizes Shapley values in order to provide an explanation for the output of any given model. This concept is a component of the game theoretic framework, renowned for its applicability in the optimization of credit allocation. The SHAP algorithm has effective

computational capabilities when used to both black box models and tree ensemble models. Calculating SHAP values on optimized model classes is an efficient approach, but it may encounter challenges in scenarios when model-agnostic parameters are similar. The additive feature of individual aggregated local SHAP values allows for their utilization in global explanations. SHAP can offer a more robust framework for conducting advanced machine learning analyses, such as fairness assessments, model monitoring, and cohort analysis.

1.5 Attention Maps

The LSTM RNN model is commonly utilized for its capacity to emphasize the precise instances in which predictions are primarily influenced by the input variables. This model also offers a high degree of interpretability for users [30]. In summary, the predicted accuracy, illness state analysis, performance breakdown, and interpretability of the RNN are enhanced. The attention vector is responsible for learning feature weights that establish a connection between the subsequent layer of the model and the most frequently utilized features. This vector is commonly employed in conjunction with LSTM to propagate attention weights towards the conclusion of the network. In this context, the weights obtained through the process of learning, denoted as W^k, are utilized to compute the value of a^k for each individual feature, denoted as x_k. In Equation (4), the value of y^k is determined by the learned attention vector, which assigns weights to the feature x_k at each time step.

$$a_k = softmax(W_k x_k) \tag{3}[32]$$

DeepSOFA [31] showcases the imperative nature of capturing individual physiological data in a time-sensitive manner inside an ICU setting. The utilization of the attention mechanism is employed to emphasize the factors within time series data that play a critical role in predicting mortality outcomes. Subsequently, the time step is allocated with increasing weights, believed to possess greater influence on the final result.

$$y_k = a_k \odot x_k \tag{4}[32]$$

2 MODEL SELECTION

2.1 Convolutional/Deep/Recurrent Neural Networks (CNN/DNN/RNN)

CNN, also known as Convolutional Neural Networks, is a prominent deep learning technique employed to model the intricate workings of the human brain. Its primary objective is to enhance performance and effectively address intricate problem-solving tasks. The process involves taking an input data or image and assigning weights and biases to its distinct elements, followed by differentiation between these factors. The filters employed in this context serve as a pertinent mechanism for transforming spatial and temporal interdependencies. Convolutional Neural Networks (CNNs) that have been specifically developed to generate structured output are commonly employed in the task of picture captioning [19]. The CNN + LSTM models have been observed to yield superior results in identifying local discriminative image regions, therefore enhancing the quality of captioning. The CNN scoring method (CNN stands for Convolutional Neural Network) offers accurate localization, as indicated by reference [16]. Subsequently, the scores are computed utilizing specific categories and predetermined criteria. The deep neural network (DNN) [33] is characterized by its architecture, which includes numerous hidden layers within the network. Once the deep neural network (DNN) has undergone training, it has

the capability to deliver enhanced performance in detecting suspicious picture findings. This improved performance may be effectively utilized for the purpose of fault identification and status determination. Recurrent Neural Networks (RNNs) are predominantly employed in the domain of natural language processing due to their ability to effectively handle sequential input. The internal memory structure of a system is typically favored for the purpose of retaining its input, making it particularly well-suited for machine learning techniques that include sequential data. The bi-directional recurrent neural network (RNN) [18] has been specifically constructed to serve as both an encoder and a decoder, effectively simulating the process of scanning through sequences during decoding. Hence, it is possible to obtain the sequences of forward and backward concealed states.

2.2 Long-Short-Term Memory (LSTM)

The utilization of Long Short-Term Memory (LSTM) has facilitated progress in the areas of processing, categorizing, and predicting time series data. The issue of the vanishing gradient is commonly addressed through the utilization of Long Short-Term Memory (LSTM) networks. The utilization of the bi-directional Long Short-Term Memory (LSTM) [23] is employed for the purpose of modeling both the within and across numerous structures, taking into account the spatial dependencies. The Deeplight model has a bi-directional Long Short-Term Memory (LSTM) architecture, consisting of two separate LSTM units that operate in opposite directions. The outputs of these LSTM units are subsequently fed into a fully-connected softmax output layer. The Long Short-Term Memory (LSTM) encoder processes embedded sequences of size n using a dual-layer architecture with n cells, and generates dense layers as output. The second Long Short-Term Memory (LSTM) model is designed with a reverse architecture, sometimes referred to as a decoder, which aims to recover the input data. The inclusion of a dropout layer between the encoder and decoder can be employed as a means to mitigate the issue of overfitting. This study employs the linear/non-linear classifier f to analyze the input variable a, which has a dimension of d. The classifier's positive prediction, $f(a) > 0$, is considered. Additionally, the relevance of the single dimension R_d is taken into account.

$$f(a) \approx \sum_{d=1}^{D} R_d \tag{5}[32]$$

In this context, $R_j^{(l)}$ represents a neural network layer with a single neuron at layer l. $R_{i\leftarrow j}^{(l-1,l)}$ refers to the deep light definition of the connection between neuron i at layer $l-1$ and neuron j at layer l. This connection is represented by Z_{ij}, which is calculated as the product of the input $a_i^{(l-1)}$ and the weight coefficient $w_{ij}^{(l-1,l)}$. Additionally, the stabilizer ϵ is included in Equation (6) to ensure stability.

$$\begin{aligned} R_j^{(l)} &= \sum_{i\in(l)} R_{i\leftarrow j}^{(l-1,l)} \\ R_{i\leftarrow j}^{(l-1,l)} &= \frac{Z_{ij}}{Z_j + \in .sign(Z_j)} R_j^{(l)} \end{aligned} \tag{6}[32]$$

2.3 Recent Machine Learning-Based Approaches

Support Vector Machines (SVMs) are a type of supervised learning algorithms that are utilized for regression, classification, and outlier detection tasks. High-dimensional spaces are commonly preferred for its usage, often exceeding

the size of the sample. The linear Support Vector Machine (SVM) [26] is commonly employed in the analysis of extremely large datasets to address multiclass classification tasks. Specifically, it utilizes the cutting plane technique as its underlying framework. The polynomial Support Vector Machine (SVM), also referred to as the polynomial kernel, is a mathematical model that represents polynomials in a feature space. This model is designed to analyze a training set by emphasizing the similarity between vectors. The degree parameter regulates the level of flexibility exhibited by the decision boundary. Therefore, the decision boundary has the potential to expand as a result of utilizing a kernel with a larger degree. The Support Vector Machine (SVM) also incorporates an additional kernel function referred to as the Gaussian Radial Basis Function (RBF). The RBF kernel is a value that is computed based on the distance from a certain point or origin. The term "deep belief network" (DBN) refers to a class or generative graphical model within the field of machine learning [34]. The construction of the model involves the incorporation of latent variables organized in several layers, wherein the layers are interrelated with the exception of the units within each layer. The deep rule forest (DRF) is a type of multilayer tree model that leverages rules to represent the combination of attributes and their interaction with outcomes [35]. The Discriminative Random Forest (DRF) is a method that utilizes techniques derived from random forest and deep learning to detect and analyze interactions. The reduction of validation errors can be achieved by the process of fine-tuning the hyperparameters of deep reinforcement learning frameworks (DRFs). The Dynamic Bayesian Network (DBN) is comprised of a sequence of transformations applied to a Restricted Boltzmann Machine (RBM), where each node in the RBM has a posterior probability that can take on values of either 1 or 0 [36].

$$P(h_i = 1|v) = f(b_i = W_i v) \tag{7}[32]$$

$$P(h_i = 1|h) = f(a_i = W_i h) \tag{8}[32]$$

Here, the $f(x) = 1 \,/\, (1 + e^{-x})$, which has energy and distribution function as:

$$E(v,h) = -\sum_{i \in v} a_i v_i - \sum_{j \in h} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{9}[32]$$

$$P(v,h) = -\frac{1}{z} e^{-E(v,h)} \tag{10}[32]$$

The Restricted Boltzmann Machine (RBM) employs unsupervised learning techniques, utilizing a probability density function pdf $p(v)$ and a likelihood function θ that is parameterized by W, a, and b. The input vector v is given as $p(v,\theta)$. The gradient method is used to optimize the likelihood function $logp(v,\theta)$, and improved learning can be obtained by updating the gradient parameters using the partial derivative of $p(v,\theta)$ with respect to θ, denoted as $\frac{\partial p(v,\theta)}{\partial \theta}$.

$$\begin{gathered}
\theta(n+1) = \theta(n) + a \times \left(-\frac{\partial p(v,\theta)}{\partial \theta}\right), \theta \in \{W, a, b\} \\
-\frac{\partial logp(v, w_{ij})}{\partial w_{ij}} = E_v\left[p(h_i|v) \times v_j\right] - v_j^{(i)} \times f(W_i \times v^{(i)} + b_i) \\
-\frac{\partial logp(v, b_i)}{\partial b_i} = E_v\left[p(h_i|v) \times v_j\right] - f(W_i \times v^{(i)}) \\
-\frac{\partial logp(v, a_j)}{\partial a_i} = E_v\left[p(h_i|v) \times v_j\right] - v_j
\end{gathered} \tag{11}[32]$$

2.4 Rule-Based Systems and Fuzzy Systems

A rule-based system utilizes knowledge representation rules to acquire the knowledge encoded inside systems. The reliance on expert systems is absolute, as these systems employ reasoning methods akin to those used by human experts to address knowledge-intensive problems. Interpretable classifiers employing Bayesian analysis have been utilized in stroke prediction models [37]. The process of interpreting decision statements is made easier by discretizing if-then conditions in a high-dimensional and multivariate feature space. The posterior distribution of the decision list is obtained through the application of the Bayesian rule. The employed framework in this context, which is designed to promote sparsity, incorporates a medical grading system that exhibits a high level of accuracy. The utilization of gradient boosting trees in interpretable mimic learning has been found to yield strong prediction performance, making it an effective knowledge distillation strategy [38]. The approach of mimic learning involves the utilization of a model consisting of both a teacher and a student. In this framework, the teacher model serves the purpose of reducing noise and error present in the training data. Additionally, soft labels are employed as a kind of regularization to prevent overfitting in the student model. The application of this approach is observed within the medical field, namely in the context of acute lung injury, where it has demonstrated notable efficacy in generating accurate predictions. Furthermore, it has been observed that this approach can be effectively utilized in the domains of voice processing, multitask learning, and reinforcement learning. Fuzzy rules can be characterized as a type of conditional statement, specifically if-then sentences, which provide a degree of truth rather than a binary true/false outcome. The prediction of ICU patient mortality is facilitated by a sophisticated rule-based fuzzy system that incorporates a diverse dataset comprising both categorical and numeric attributes organized in a hierarchical structure [39]. The model contains interpretable fuzzy rules that are located in each unit of the hidden layer. In order to enhance interpretability, a guided random attribute shift is incorporated into the stack technique. Supervised clustering involves the utilization of a fuzzy partition matrix and cluster centers. In Equation (12), the output weight vectors, denoted as β_{dp}, correspond to a building unit indexed by dp. The partition matrix is represented by U_{dp}, and the output set is denoted as T.

$$\beta_{dp} = \left(\frac{1}{Const} I + U_{dp}^{T} U_{dp}\right)^{-1} U_{dp} T \quad (12)[32]$$

The interpretability of the layer's prediction can be enhanced by using random projections to increase linear separability. In this context, α' represents the sub constants, Z_{dp} represents the random projection matrix, and Y_{dp} represents the output vector of the last unit.

$$X_{dp} = X + \alpha' Y_{dp} Z_{dp}$$

$$Y_{dp} = U_{dp} \beta_{dp} \quad (13)[32]$$

2.5 Additional XAI Methods for Plots, Expectations, and Explanations

The partial dependence plot (PDP) in the field of machine learning illustrates the marginal impact of one or several input features on the ultimate prediction. Typically, this relationship exhibits a partial dependency. The PDP algorithm calculates the mean value of all input variables, excluding the PDP computed variable n [40]. The variable n is thereafter examined

in relation to the alteration in the target variable for the intention of documenting and graphing. When comparing the PDP to individual conditional expectancies, the latter specifically examine particular cases that reveal differences in the recovery of subgroups within the patient population [41]. The optimal approach for explaining the classifier prediction using eXplainable Artificial Intelligence (XAI) is through the utilization of Local Interpretable Model-agnostic Explanations (LIME). LIME serves as an interpretable model that approximates the behavior of a black box model specifically for the instance being analyzed [42]. The artifacts refer to modules that are created by the user and can be interpreted. These modules are subsequently utilized to create local black boxes, specifically for neighboring instances. Semantic LIME (S-LIME) effectively addresses the constraints imposed by user intervention and artifact limitations. This is achieved through the utilization of independently generated semantic characteristics, which are obtained utilizing unsupervised learning techniques. The fidelity function is defined as follows: it involves a model g, an instance x and y, and a feature that measures agreement. The function π is used, which employs an exponential kernel with weighted σ and a distance D.

$$\mathcal{F}(x, f, g, \pi) = \sum_{y \epsilon X} \pi(x, y). \left(f(y) - g(y)\right)^2 \quad (14)[32]$$

$$D(x, y) = \sum_{x_1=1} |x_i - y_i| \quad (15)[32]$$

LIME is a widely utilized method for emphasizing significant aspects and offering explanations based on its coefficient. However, its utility is hindered by the presence of randomness in the sample step, rendering it unsuitable for implementation in medical contexts. In order to establish confidence, protect interests, and mitigate legal concerns, a proposed method called optimized LIME explanations (OptiLIME) is recommended for diagnostic purposes [43]. The mathematical aspects of OptiLIME are prominently emphasized and maintained consistently during multiple iterations to effectively explore the optimal kernel width in an automated manner. According to the formula provided in Equation (16), the diminishing R^2 is transformed into $l(kw, \tilde{R}^2)$, which represents a global maximum, in order to determine the optimal width. The $\widetilde{R}^2$ represents the anticipated level of adherence when random kw values are considered.

$$l\left(kw, \tilde{R}^2\right) = \begin{cases} R^2(kw), if\ R^2(kw) \leq \tilde{R}^2 \\ 2\tilde{R}^2 - \tilde{R}^2(kw), if R^2(kw) > \tilde{R}^2 \end{cases} \quad (16)[32]$$

The conventional receiver operating characteristic (ROC) plot and area under the curve (AUC) are influenced by the adjustable threshold, which in turn affects the occurrence of false positive and false negative errors [44]. The utilization of partial ROC and AUC measures in the context of unbalanced data is valuable. Additional approaches, such as partial AUC and the area under the precision-recall (PR) curve, have been proposed as optional solutions. However, it is important to note that these methods alone may not provide a comprehensive solution and should be used with caution. Hence, a novel approach referred to as partial area under the curve $pAUC$ and c statistics of receiver operating characteristic (ROC) have been introduced, preserving the continuous and discrete properties of the area under the curve (AUC), respectively. In the context of evaluating the performance of a binary classification model, the horizontal partial Area Under the Curve (AUC) is computed by considering $x = 1$ as the integration border for the AUC calculation, while designating the other regions as true negatives. When considering the integration with the baseline as the x-axis, it is important to note that the baseline

x-value is set to 0 when swapping the x and y axes. Therefore, by converting the variable x (false positive rate) to $1 - x$ (true negative rate), the desired true negative rate (TNR) may be obtained, and when $x = 0$, it becomes 1.

$$pAUC_x \triangleq \int_{y_1}^{y_2} 1 - r^{-1}(y)dy \quad (17)[32]$$

The normalized form of the partial c statistic (c_Δ) for ROC data is expressed in Equation (18). The c_Δ can be represented as a ratio of J from the set of positive elements P, while k can be considered as a subset of negative elements N.

$$\hat{C} \triangleq \frac{2PN.c_\Delta}{J.N + K.P} \quad (18)[32]$$

The partial c statistic can by summed up as shown by the whole curve having q disjoint partial curves.

$$c = \sum_{i=1}^{q} (c_\Delta)_i \quad (19)[32]$$

3 ATTENTION MECHANISMS

Attention mechanisms are integrated to highlight regions in medical images that contribute significantly to the model's decision. Self-attention mechanisms, inspired by the transformer architecture, have shown promise in medical image analysis [1]. These mechanisms allow the model to focus on relevant areas of an image.

3.1 Background

The objective of minimizing sequential processing is also the fundamental principle of the Extended Neural GPU [50], ByteNet [51], and ConvS2S [52]. These models employ convolutional neural networks as its fundamental components, enabling the simultaneous computation of hidden representations for all input and output positions. In the aforementioned models, the computational complexity associated with establishing connections between signals originating from any two input or output places increases proportionally with the spatial separation of these positions. Specifically, ConvS2S exhibits a linear growth pattern, whereas ByteNet has a logarithmic growth pattern. This phenomenon introduces additional challenges in acquiring knowledge of the relationships between sites that are far apart [53]. In the Transformer model, the number of operations is reduced to a constant value. However, this reduction comes at the expense of decreased effective resolution, which occurs because attention-weighted positions are averaged. To mitigate this effect, we employ Multi-Head Attention, as explained in section 4.3. Self-attention, also known as intra-attention, refers to an attention process that establishes connections between various positions within a singular sequence, with the purpose of generating a representation of said sequence. The utilization of self-attention has proven to be effective in a range of tasks such as reading comprehension, abstractive summarization, textual entailment, and the acquisition of phrase representations that are independent of specific tasks [54, 55, 56, 57]. The utilization of end-to-end memory networks is founded on a recurrent attention mechanism, as opposed to sequence aligned recurrence. These networks have demonstrated strong performance in tasks such as simple-language question answering and language modeling [58]. Based on current understanding, it can be asserted that the Transformer represents a novel transduction model that exclusively utilizes self-attention for the

computation of input and output representations, hence eliminating the need for sequence aligned RNNs or convolution. In the subsequent parts, we shall elucidate the Transformer, provide rationale for self-attention, and deliberate on its merits in comparison to models referenced as [59, 60] and [52].

3.2 Model Architecture

Most competitive neural sequence transduction models have an encoder-decoder structure [61, 62, 63]. Here, the encoder maps an input sequence of symbol representations $(x_1, \dots, x_n)$ to a sequence of continuous representations $z = (z_1, \dots, z_n)$. Given z, the decoder then generates an output sequence $(y_1, \dots, y_m)$ of symbols one element at a time. At each step the model is auto-regressive [64], consuming the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 3, respectively.

3.3 Encoder and Decoder Stacks

Encoder: The encoder is composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. We employ a residual connection [65] around each of the two sub-layers, followed by layer normalization [66]. That is, the output of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension $d_{model} = 512$.

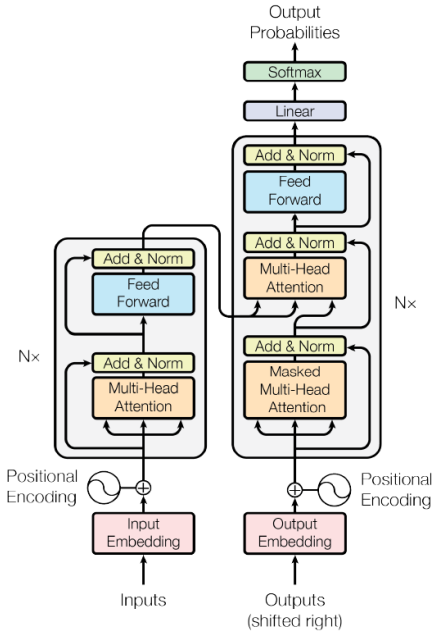


Figure 3: The Transformer – model architecture [1].

Decoder: The decoder is comprised of a stack consisting of $N = 6$ identical layers. Furthermore, the decoder incorporates an additional sub-layer in addition to the two existing sub-layers in each encoder layer. This additional sub-layer is responsible for conducting multi-head attention on the output of the encoder stack. In a manner akin to the encoder, we

utilize residual connections surrounding each of the sub-layers, which are subsequently followed by layer normalization. In order to prevent positions inside the decoder stack from attending to following positions, we make modifications to the self-attention sub-layer. The utilization of masking, in conjunction with the adjustment of output embeddings by one position, guarantees that the predictions for a given position i are solely influenced by the known outputs at positions preceding i.

3.4 Attention

The concept of an attention function involves the process of mapping a query and a collection of key-value pairs to generate an output. In this context, the query, keys, values, and output are all represented as vectors. The resulting value is calculated by taking a weighted total of the values, with each value being assigned a weight determined by a compatibility function that compares the query to the relevant key.

3.5 Scaled Dot-Product Attention

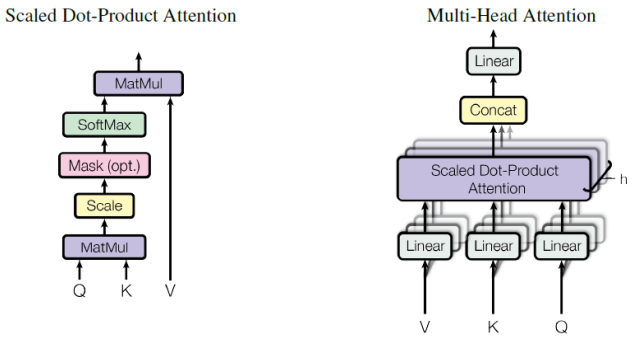


Figure 4: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel [1].

The attention mechanism we focus on in this study is referred to as "Scaled Dot-Product Attention" (see Figure 4). The input comprises queries and keys with a dimension of d_k, as well as values with a dimension of d_v. The dot products between the query and all keys are calculated, then each dot product is divided by the value of $\sqrt{d_k}$. Finally, a softmax function is applied to determine the weights assigned to the values.

In practical implementation, the attention function is computed on a collective collection of inquiries, which are organized and processed as a matrix denoted as Q. The keys and values are further consolidated into matrices K and V. The matrix of outputs is computed as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (20)[1]$$

The two attention functions that are frequently employed in various contexts are additive attention [2] and dot-product (multiplicative) attention. The dot-product attention mechanism shares similarities with our approach, with the exception of the scaling factor of $\frac{1}{\sqrt{d_k}}$. The compatibility function in additive attention is computed by employing a feed-forward network that consists of a solitary hidden layer. Although both dot-product attention and its counterpart have similarities

in terms of theoretical difficulty, the former exhibits superior practical performance due to its expedient execution and effective utilization of memory. This advantage stems from the fact that dot-product attention may be implemented through the utilization of meticulously designed matrix multiplication code. In the case of small values of d_k, the performance of the two processes is comparable. However, for larger values of d_k, additive attention demonstrates superior performance compared to dot product attention without scaling, as indicated by previous research [3]. It is hypothesized that as the values of d_k increase significantly, the dot products exhibit substantial magnitudes, hence causing the softmax function to operate inside regions characterized by exceedingly small gradients. In order to mitigate this effect, we adjust the dot products by multiplying them by the factor $\frac{1}{\sqrt{d_k}}$.

3.6 Multi-Head Attention

Instead of implementing a singular attention function with keys, values, and queries of d_{model} dimensions, we discovered that it is advantageous to employ h separate linear projections to transform the queries, keys, and values h times. These linear projections are learned and result in dimensions of d_k, d_k and d_v for the queries, keys, and values, respectively. The attention function is applied in parallel to each projected version of queries, keys, and values, resulting in output values of dimension d_v. The values represented in Figure 4 are obtained by concatenating and subsequently projecting the given data. The utilization of multi-head attention enables the model to collectively focus on input from distinct representation subspaces at various places. The process of averaging decreases the effectiveness of a single attention head.

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_h)W^O \; where \; head_i \\ = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

The parameter matrices for the projections are denoted as $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. In this study, we utilize eight parallel attention layers, also referred to as heads. In each of these cases, the values of d_k, d_v, $\frac{d_{model}}{h} = 64$ are set to 64. The computational cost of multi-head attention is comparable to that of single-head attention with full dimensionality, owing to the decreased dimension of each head.

3.7 Applications of Attention in our Model

The Transformer uses multi-head attention in three different ways:

- In "encoder-decoder attention" layers, the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder. This allows every position in the decoder to attend over all positions in the input sequence. This mimics the typical encoder-decoder attention mechanisms in sequence-to-sequence models such as [67, 62, 52].
- The encoder contains self-attention layers. In a self-attention layer all of the keys, values and queries come from the same place, in this case, the output of the previous layer in the encoder. Each position in the encoder can attend to all positions in the previous layer of the encoder.
- Similarly, self-attention layers in the decoder allow each position in the decoder to attend to all positions in the decoder up to and including that position. We need to prevent leftward information flow in the decoder to preserve the auto-regressive property. We implement this inside of scaled dot-product attention by masking out (setting to $-\infty$) all values in the input of the softmax which correspond to illegal connections. See Figure 4.

3.8 Position-wise Feed-Forward Networks

Furthermore, within both the encoder and decoder, every layer is equipped with a fully connected feed-forward network that operates independently and uniformly on each location. The composition of two linear transformations is performed, with a Rectified Linear Unit (ReLU) activation function applied in between.

$$FFN(x) = \max(0, xW_1 + b_1)\, W_2 + b_2 \quad (21)[1]$$

Although the linear transformations remain consistent across various places, they employ distinct parameters as one moves from one layer to another. An alternative manner of articulating this concept is characterizing it as the combination of two convolutions with a kernel size of 1. The input and output dimensions are both equal to d_{model} = 512. Additionally, the inner-layer has a dimensionality of d_{ff} = 2048.

3.9 Positional Encoding

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n^2 \cdot d)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention(restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Table 1: This study investigates the maximum path lengths, per-layer complexity, and minimum number of sequential operations associated with various layer types. The length of the sequence is denoted as *n*, the representation dimension is denoted as *d*, the kernel size of convolutions is denoted as *k*, and the size of the neighborhood in restricted self-attention is denoted as *r* [1].

Given that our model lacks recurrence and convolution, it becomes necessary to incorporate details regarding the relative or absolute position of tokens in the sequence. This is crucial for the model to effectively utilize the sequence's order. In order to achieve this objective, we incorporate "positional encodings" into the input embeddings located at the lowermost layers of both the encoder and decoder stacks. The positional encodings and embeddings share a common dimension, denoted as d_{model}, allowing for their summation. There exists a wide range of positional encodings, both learned and fixed [52].

In this work, we use sine and cosine functions of different frequencies:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

In other words, each dimension of the positional encoding is associated with a sinusoidal function. The wavelengths exhibit a geometric development ranging from 2π to $10000 \cdot 2\pi$.. The selection of this particular function was based on our hypothesis that it would facilitate the model's ability to train attention based on relative positions. This is due to the fact that, for any constant offset k, PE_{pos+k}, the positional encoding at position p can be expressed as a linear function PE_{pos}.

Additionally, we conducted experiments involving the utilization of learned positional embeddings [52]. It was observed that the two variations yielded almost indistinguishable outcomes, as depicted in Table 1, row (E). The sinusoidal variant was selected due to its potential to enable the model to extrapolate to sequence lengths that beyond those experienced during the training phase.

4 FEATURE IMPORTANCE ANALYTICS

SHAP (SHapley Additive exPlanations)[2] values attribute the contribution of each feature to the prediction, offering insights into the decision-making process. This can be especially useful in cases where features are not visually

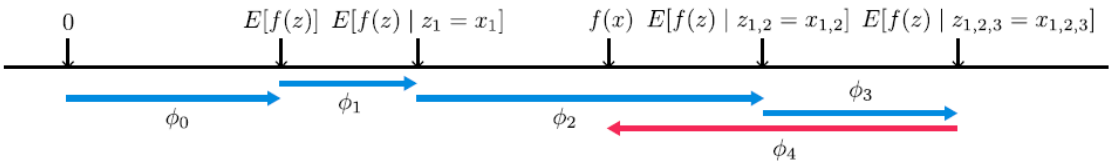


Figure 5: The SHAP (SHapley Additive exPlanation) values measure the change in anticipated model prediction when each feature is considered. The authors explain how to get from the initial anticipated value E[f(z)] to the current output f(x) without feature knowledge. The SHAP values are calculated by averaging individual values (ϕ_i) across all possible orderings. [2]

interpretable, such as lab values or genomic data. In this study, we put forth the utilization of SHAP values as a comprehensive metric for assessing the significance of features. The Shapley values of the conditional expectation function in the original model can be determined as the solution to Equation 8. In this equation, $f_x(z')$ represents the conditional expectation function $f\left(h_x(z')\right) = E|f(z)z_S|$, where S is the set of non-zero indexes in z' (Figure 5). The utilization of SHAP values, offers a distinctive approach to quantifying the relevance of features in an additive manner. These values possess the desirable attributes of adhering to Properties 1-3 and rely on conditional expectations to establish simpler representations of inputs. The definition of SHAP values assumes a simplified input mapping, denoted as $h_x(z') = z_S$, where z_S represents the input with missing values for features not included in the set S. Due to the limited capability of most models in accommodating random patterns of missing input values, it is necessary to approximate the function $f(z_S)$ by employing the expected value of $f(z)$ given z_S. The provided definition of SHAP values aims to establish a strong correspondence with Shapley regression, Shapley sampling, and quantitative input influence feature attributions. Additionally, it enables the establishment of associations with LIME, DeepLIFT, and layer-wise relevance propagation.

Property 1 (Local Accuracy) [2]

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i' \quad (22)[2]$$

The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$

Property 2 (Missingness) [2]

$$x_i' = 0 \Longrightarrow \phi_i = 0 \quad (23)[2]$$

Missingness constrains features where $x_i' = 0$ to have no attributed impact

Property 3 (Consistency) [2] *Let $f_x(z') = f(h_x(z'))$ and $z'\backslash i$ denote setting $z_i' = 0$. For any two models f and f', if*

$$f_x'(z') - f_x'(z'\backslash i) \geq f_x(z') - f_x(z'\backslash i) \quad (24)[2]$$

For all inputs $z' \in \{0,1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$

The precise calculation of SHAP values presents significant difficulties. Nevertheless, it is possible to approximate these methods by integrating the insights obtained from existing additive feature attribution techniques. In this study, we provide two model-agnostic approximation techniques, namely the well-established Shapley sampling values method and a unique approach called Kernel SHAP. In addition, we present a description of four approximation approaches that are particular to model types, two of which are considered innovative (Max SHAP and Deep SHAP). When employing these techniques, the assumptions of feature independence and model linearity can be made, which serve to simplify the computation of predicted values. It should be noted that the set of features not included in S is denoted as $\bar{S}$.

$$f(h_x(z')) = E[f(z)|z_S] \quad \text{SHAP explanation model simplified input mapping} \quad (25)[2]$$

$$= E_{z_{\bar{S}}|z_S}[f(z)] \quad \text{expectation over } z_{\bar{S}}|z_S \quad (26)[2]$$

$$= E_{z_{\bar{S}}}[f(z)] \quad \text{assume feature independence ([46],[10],[47],[48])} \quad (27)[2]$$

$$\approx f(|z_S, E[z_{\bar{S}}]|) \quad \text{assume model linearity} \quad (28)[2]$$

4.1 Model-Agnostic Approximations

If the assumption of feature independence is made when estimating conditional expectancies (as stated in Equation 28), it is possible to estimate SHAP values directly using either the Shapley sampling values method [46] or the Quantitative Input Influence method [48]. This approach is supported by previous studies [10, 47]. The aforementioned methods employ a sampling approximation technique to estimate the permutation-based variant of the well-known Shapley value equations (Equation 29). Individual sampling estimations are conducted for each feature attribution. The Kernel SHAP method, as explained in the following section, exhibits a reduced requirement for evaluations of the original model in order to achieve comparable approximation accuracy, which is feasible for a limited number of inputs.

$$\Phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!\,(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z'\backslash i)] \quad (29)[2]$$

4.1.1Kernel SHAP (Linear LIME + Shapely values)

The Linear LIME approach utilizes a linear explanation model to locally approximate the function f. The notion of locality is quantified within the simplified binary input space. Upon initial examination, it is evident that the regression formulation of LIME, as depicted in Equation 30, has notable dissimilarities when compared to the standard Shapley value formulation represented by Equation 8. Nevertheless, given that linear LIME operates as an additive approach for feature attribution, it is established that the Shapley values represent the sole feasible solution for Equation 29, while also adhering to Properties 1-3, namely local accuracy, missingness, and consistency. An inquiry that arises naturally is whether the solution to Equation 30 is able to retrieve these values. The answer is contingent upon the selection of the loss function L, weighting kernel $\pi_{x'}$ and regularization term Ω. The selection of parameters in LIME is based on heuristics. However, it should be noted that Equation 30 fails to accurately estimate the Shapley values when these parameters are employed. One potential

outcome is the violation of local accuracy and/or consistency, resulting in counterintuitive behavior under specific conditions.

In the following section, we present a method to circumvent the heuristic selection of parameters in Equation 30. Additionally, we outline the process of determining the loss function L, weighting kernel $\pi_{x\prime}$, and regularization term Ω that effectively restore the Shapley values.

Definition 1 Additive feature attribution methods have an explanation model that is a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i', \tag{30}[2]$$

Theorem 1 (Shapley kernel) *Under Definition 1, the specific forms of $\pi_{x\prime}$, L and Ω that make the solutions of Equation 2 consistent with Properties 1 through 3 are:*

$$\Omega(g) = 0,$$
$$\pi_{x'}(z') = \frac{(M-1)}{(M\ choose\ |z'|)|z'|(M-|z'|)}$$
$$L(f, g, \pi_{x'}) = \sum_{x\prime \epsilon Z} \left[f\left(h_x^{-1}(z')\right) - g(z')\right]^2 \pi_{x'}(z')$$

Where $|z'|$ is the number of non-zero elements in z'

It is of significance to acknowledge that the value of $\pi_{x'}(z')$ is equal to ∞ when the absolute value of z' is within the range of 0 to M. This condition establishes that ϕ_o is a function of $f_x(\emptyset)$ and $f(x) = \sum_{i=0}^{M} \phi_i$. In practical applications, the issue of infinite weights can be circumvented through the process of analytically reducing two variables by incorporating these limitations into the optimization procedure. Given the assumption that $g(z')$ in Theorem 1 adheres to a linear form, and considering that L represents a squared loss, it is possible to solve Equation 30 by employing linear regression. Therefore, the computation of Shapley values in game theory can be achieved by the utilization of weighted linear regression. The user's text is too short to be rewritten academically. The simplified input mapping employed by LIME is comparable to the approximation of the SHAP mapping described in Equation 29. This characteristic facilitates the regression-based, model-agnostic estimation of SHAP values. The utilization of regression for the joint estimation of all SHAP values is found to offer improved sample efficiency compared to the direct application of classical Shapley equations. The inherent relationship between linear regression and Shapley values can be understood by recognizing that Equation 29 represents a disparity in means. Given that the mean serves as the optimal least squares point estimate for a given set of data points, it is logical to seek a weighting kernel that enables linear least squares regression to replicate the Shapley values. This results in a kernel that exhibits a clear distinction from previously selected kernels based on heuristics (Figure 6A).

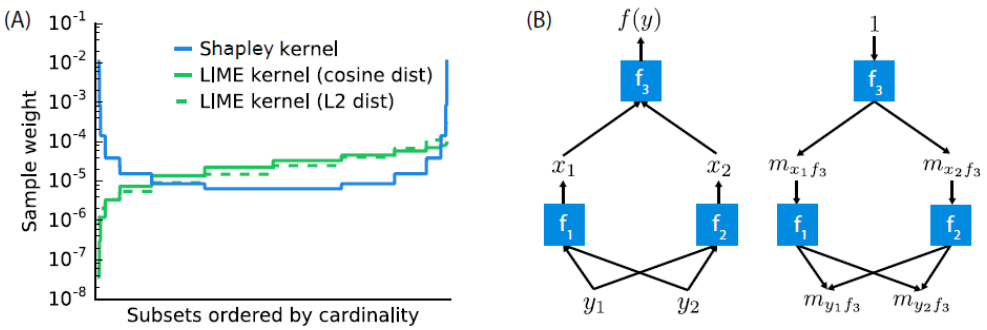


Figure 6: (A) When all potential z0 vectors are sorted in ascending cardinality order, the Shapley kernel weighting is symmetric. This sample has 215 vectors. This differs from heuristic-selected kernels. Compositional models like deep neural networks have many basic parts. DeepLIFT's back-propagation technique can efficiently approximate the full model using analytic answers for component Shapley values. [2]

4.2 Model Specific Approximations

Kernel SHAP enhances the effectiveness of model-agnostic estimations of SHAP values by improving sample efficiency. However, by focusing on particular model types, we can devise quicker approximation approaches that are specific to those models.

Linear SHAP

In the context of linear models, it is possible to approximate SHAP values directly from the weight coefficients of the model, provided that we assume independence among the input features (as stated in Equation 28).

Corollary 1 (Linear SHAP) *Given a linear model* $f(x) = \sum_{j=1}^{M} w_j x_j b;\ \phi_0(f, x) = b$ *and* $\phi_i(f, x) = w_j(x_j - E|x_j|)$

This conclusion can be derived from Theorem 1 and Equation 28, as previously observed by Štrumbelj and Kononenko [46].

Low-Order SHAP

Linear regression using Theorem 2 exhibits a computational complexity of $O(2^M + M^3)$, rendering it efficient for cases when M is small, particularly when employing an approximation of the conditional expectations (Equation 28 or 29).

Max SHAP

Using a permutation formulation of Shapley values, we can calculate the probability that each input will increase the maximum value over every other input. Doing this on a sorted order of input values lets us compute the Shapley values of a max function with M inputs in $O(M^2)$ time instead of $O(M2^M)$.

Deep SHAP (DeepLIFT + Shapley values)

Although Kernel SHAP has the capability to be applied to many models, including deep models, it is vital to inquire whether it is possible to exploit additional knowledge regarding the compositional characteristics of deep networks in order to enhance computational efficiency. The resolution to this inquiry is attained by means of an overlooked correlation between Shapley values and DeepLIFT [49]. If the reference value in Equation 31 is interpreted as representing the expected value of $E[x]$ in Equation 29, then DeepLIFT provides an approximation of SHAP values under the assumption that the input features are independent of each other and the deep model is linear. DeepLIFT employs a linear composition rule that effectively linearizes the non-linear elements within a neural network. The back-propagation rules, which determine the linearization of each component, possess an intuitive nature yet were selected by heuristic methods. DeepLIFT is an additive technique for attributing features, which ensures both local correctness and missingness. It is important to note that consistency is exclusively satisfied by Shapley values as attribution values. The motivation behind our endeavor is to modify DeepLIFT in order to serve as a compositional approximation of SHAP values, resulting in the development of Deep SHAP.

The Deep SHAP methodology integrates SHAP values that are calculated for individual components of the network in order to derive SHAP values for the entire network. The process is accomplished by iteratively transmitting DeepLIFT's multipliers, which are currently expressed in terms of SHAP values, in a reverse manner via the network, as illustrated in Figure 6B.

$$\sum_{i=1}^{n} C_{\Delta xi\Delta o} = \Delta \mathrm{o} \tag{31}[2]$$

Since the SHAP values for the simple network components can be efficiently solved analytically if they are linear, max pooling, or an activation function with just one input, this composition rule enables a fast approximation of values for the whole model. Deep SHAP avoids the need to heuristically choose ways to linearize components. Instead, it derives an effective linearization from the SHAP values computed for each component. The max function offers one example where this leads to improved attributions

5 LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS (LIME)

Employ LIME to generate locally faithful explanations for individual predictions [3]. LIME creates surrogate interpretable models that approximate the behavior of the complex AI model in the vicinity of a specific prediction.

5.1 Interpretable Data Representations

It is crucial to make a distinction between features and interpretable data representations before introducing the explanation system. Regardless of the actual features that the model employs, interpretable explanations must use a representation that is understandable to people. While the classifier may employ more intricate (and incomprehensible) features like word embeddings, one potential interpretable form for text classification is a binary vector denoting the presence or absence of a word. The interpretable representation for image classification may be a binary vector that indicates the "presence" or "absence" of a contiguous patch of similar pixels (a super-pixel), whereas the classifier may represent the image as a tensor with three color channels per pixel. We denote $x \in R^d$ be the original representation of an instance being explained, and we use $x' \in \{0,1\}^{d'}$ to denote a binary vector for its interpretable representation.

5.2 Fidelity-Interpretability Trade-off

In a formal manner, an explanation is defined as a model g that belongs to the class G. The class G consists of models that have the potential to be interpreted, such as linear models, decision trees, or falling rule lists [7]. In other words, a model g that belongs to G can be easily given to the user using visual or textual means. The domain of function g is defined as the set $\{0,1\}^{d'}$, indicating that g operates based on the existence or absence of interpretable components. Not all elements $g \in G$ may possess a level of simplicity that allows for easy interpretation. Therefore, we define $\Omega(g)$ as a metric of complexity, rather than interpretability, for the explanation $g \in G$. As an illustration, in the case of decision trees, $\Omega(g)$ might represent the depth of the tree, whereas in the context of linear models, $\Omega(g)$ could denote the count of non-zero weights.

Let us represent the model being explained as $f: R^d \rightarrow R$. In the context of classification, the function $f(x)$ represents the probability or binary indicator denoting the membership of x within a specific class.

The proximity measure $\pi_x(z)$ is employed to determine the distance between an instance z and x, hence establishing the concept of locality surrounding x. Let us denote $L(f, g, \pi_x)$ as a metric that quantifies the degree of discrepancy between the approximation of function g and the true representation of function x inside the specified locality indicated by π_x. To achieve both interpretability and local faithfulness, it is necessary to minimize the function $L(f, g, \pi_x)$ while ensuring that the complexity measure $\Omega(g)$ remains sufficiently low for human interpretability. The explanation generated by LIME is acquired using the following process:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}}\, L(f, g, \pi_x) + \Omega(g) \qquad (32)[10]$$

This formulation has the potential to be applied with various families G, fidelity functions L, and complexity measures Ω. This study primarily centers on sparse linear models as a means of providing explanations, with a particular emphasis on conducting the search process through perturbations.

5.3 Sampling for Local Exploration

The objective is to reduce the locality-aware loss, denoted as $L(f, g, \pi_x)$, without imposing any assumptions on the function f. This is desired in order to ensure that the explanation remains independent of the specific model being used. Therefore, in order to understand the regional characteristics of function f when the comprehensible inputs change, we estimate the value of $L(f, g, \pi_x)$ by randomly selecting samples, with their weights determined by π_x. Instances around x' are sampled by randomly selecting nonzero items from x' in a uniform manner. The number of such selections is similarly uniformly sampled. In this study, we are provided with a disturbed sample z', where z' belongs to the set $\{0,1\}$ and represents a percentage of the nonzero components of x'. Our objective is to restore the sample to its original representation z in R^d. Once we have obtained z, we calculate $f(z)$, which serves as the label for the explanation model. The dataset Z consists of altered samples together with their corresponding labels. We aim to optimize Equation 32 in order to obtain an explanation $\xi(x)$. The fundamental concept underlying LIME is illustrated in Figure 7, where we select examples that are

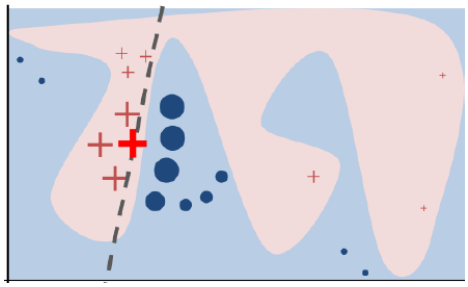


Figure 7: Illustration of LIME. The black-box model's complex decision function f is unknown to LIME. A linear model cannot approximate the blue/pink background. The red cross shows the example. LIME samples instance, predicts with f, and weights instances by size and closeness. The dashed line shows local but not worldwide accurate acquired explanation. [10]

both close to x (which have a higher weight due to π_x) and far from x (which have a lower weight from π_x). While the original model may include excessive complexity for a comprehensive explanation, LIME offers a locally faithful explanation that adheres to linearity in this particular instance. The concept of locality is effectively encapsulated by the variable π_x. It is important to acknowledge that our method demonstrates a considerable level of resilience to sampling noise due to the incorporation of sample weights based on π_x as described in Equation 32. In this study, we give a specific example that exemplifies the broader concept discussed.

5.4 Sparse Linear Explanations

In the subsequent sections of this study, we shall denote G as the set of linear models, where $g(z') = w_g * z'$. The locally weighted square loss, denoted as L, is employed in our study, as specified in Equation (33). In this context, we define $\pi_x(z)$ as an exponential kernel, which is mathematically represented as $exp(-D(x,z)^2 / \sigma^2)$. This kernel is defined based on a distance function D, such as cosine distance for text or $L2$ distance for images, and it is characterized by a width parameter σ.

$$L(f, g, \pi_x) = \sum_{x,x' \in z} \pi_x(z)\big(f(z) - g(z')\big)^2 \qquad (33)[10]$$

In the context of text classification, it is imperative to guarantee that the provided explanation is easily understandable. This is achieved by employing an interpretable representation known as a bag of words. Additionally, a constraint is imposed by placing a limit, denoted as K, on the number of words. Mathematically, this constraint can be expressed as $\Omega(g) = \infty 1[|w_g|_0 > K]$. It is possible to change the value of K to accommodate the user's capacity, or alternatively, employ varying values of K for different cases. In this study, a fixed value for the parameter K is employed, deferring the investigation of alternative values to subsequent research endeavors. In the context of image classification, a common approach involves utilizing "super-pixels" instead of words, which are obtained by the application of a standard algorithm. Consequently, the interpretable representation of an image is represented by a binary vector, where the value of 1 denotes the presence of the original super-pixel, while 0 signifies a grayed out super-pixel. The specific selection of Ω in Eq. 32 poses challenges for direct solution. However, we address this issue by employing an approximation method. Firstly, we employ Lasso with the regularization route [8] to pick K features. Subsequently, we estimate the weights through the least squares method. This technique is referred to as K-LASSO and is outlined in Algorithm 1. The difficulty of Algorithm 1 is independent of the dataset size, but rather relies on the computational time required to compute $f(x)$ and the number of

samples N. In practical applications, the process of elucidating the concept of random forests, consisting of 1000 trees, is efficiently executed using the scikit-learn library (http://scikit-learn.org) on a laptop computer. Specifically, when the dataset size, denoted as N, is set to 5000, the aforementioned task may be completed in less than 3 seconds. It is important to note that this timeframe does not incorporate any optimization techniques such as utilizing graphics processing units (GPUs) or parallelization methods. The process of elucidating every forecast made by the Inception network [9] in the context of image categorization necessitates approximately 10 minutes. Every selection of interpretable representations and G will inevitably possess certain intrinsic limitations. Initially, it should be noted that although the fundamental model can be regarded as an opaque entity, there exist certain interpretable representations that may lack the capacity to elucidate specific behaviors. An instance can be illustrated by a model that predicts the retro nature of sepia-toned photographs, which cannot be elucidated only by the existence or non-existence of super pixels. Furthermore, the selection of G, specifically sparse linear models, may result in the absence of a reliable explanation if the underlying model exhibits

ALGORITHM 1: Sparse Linear Explanations using LIME[10]

Require: Classifier f, Number of samples N

Require: Instance x, and its interpretable version x^2

Require: Similarity kernel π_x, Length of explanation K

$Z \leftarrow \{\}$

for $i \in \{1, 2, 3, \dots, N$ do

$z_i' \leftarrow sample\ around(x')$

$Z \leftarrow Z \cup (z_i', f(z_i), \pi_x(z_i))$

end for

$w \leftarrow K - Lasso(Z, K)$ with z_i' as features, $f(z)$ as target return w

significant non-linearity, even within the vicinity of the prediction. Nevertheless, it is possible to make an approximation of the accuracy of the explanation regarding Z and thereafter provide this data to the user. The measure of faithfulness described here can also be employed to choose a suitable set of explanations from a collection of interpretable model classes, thereby accommodating the specific dataset and classifier being utilized. The examination of this topic will be deferred to future research, as our trials have demonstrated that linear explanations are effective for many black-box models.

5.5 Example 1: Text classification with SVMs

In the right side of Figure 8, we provide an explanation of the predictions made by a support vector machine with a radial basis function (RBF) kernel that was trained on unigrams. The purpose of this training was to distinguish between the topics of "Christianity" and "Atheism" using a subset of the 20-newsgroup dataset. Despite achieving a held-out accuracy of 94%, it is important to approach the classifier's results with caution. The explanation for an instance reveals that predictions are generated based on seemingly arbitrary factors, as phrases such as "Posting," "Host," and "Re" have no discernible link to either Christianity or Atheism. The term "Posting" is observed in 22% of instances within the training dataset, with 99% of these occurrences belonging to the category labeled as "Atheism". The classifier is capable of

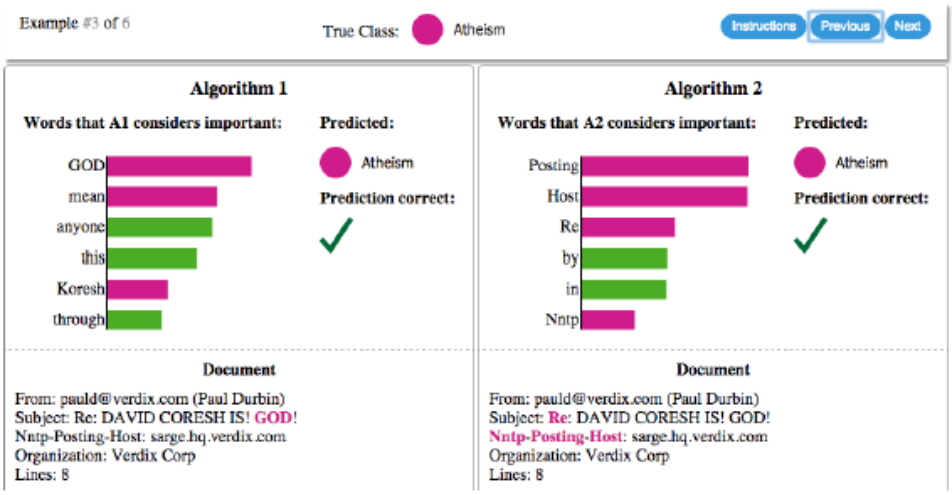
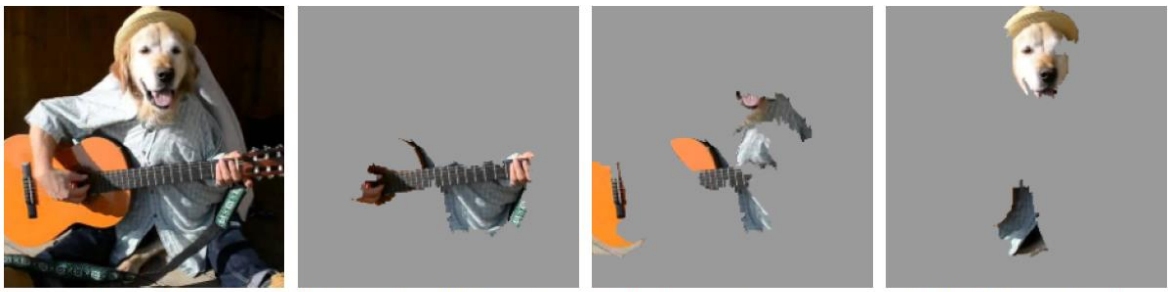


Figure 8: This image shows why rival classifiers anticipate "Christianity" or "Atheism" as a document's theme. The text highlights the bar chart's key terms. This color scheme shows word class. Choosing green for "Christianity," magenta for "Atheism." [10]

identifying the appropriate names of individuals who frequently contribute to the original newsgroups, even in the absence of headers. However, it should be noted that this ability does not extend to generalization. Upon gaining profound insights from the provided explanations, it becomes evident that the dataset in question exhibits significant flaws that are not readily apparent through a mere examination of the raw data or predictions. Consequently, it is imperative to exercise caution when relying on the classifier or held-out evaluation in this context. The identification of the problems and the formulation of appropriate measures to address these issues and enhance the reliability of the classifier are evident.

5.6 Example 2: Deep networks for images



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 9: This response aims to elucidate the prediction process of Google's Inception neural network for picture classification. The three classes that have been predicted with the highest probabilities are "Electric Guitar" (p = 0.32), "Acoustic Guitar" (p = 0.24), and "Labrador" (p = 0.21) [10].

When employing sparse linear explanations for image classifiers, it may be desirable to solely emphasize the super-pixels that exhibit positive weight towards a particular class. This approach provides insight into the reasoning behind the model's belief that the given class is likely to be present.

In this manner, we elucidate the process of predicting using Google's pre-trained Inception neural network [25] on a randomly selected image (Figure 9a). Figures 9b, 9c, and 9d depict the superpixels explanations for the three highest predicted classes, with the remaining portions of the image rendered in gray. The value of K was set to 10. The neural network demonstrates a natural ability to identify distinguishing features for each class, which aligns with human perception. Specifically, Figure 4b sheds light on the prediction of an acoustic guitar as electric, attributing it to the presence of the fretboard. This type of explanation serves to increase confidence in the classifier, even in cases when the highest predicted class is incorrect, as it demonstrates that the classifier is not behaving in an irrational manner.

6 ETHICAL CONSIDERATIONS

Modules that analyze potential ethical considerations in the diagnosis process should be integrated into the system. We must ensure the AI system adheres to fairness, transparency, and privacy guidelines to avoid biased or harmful decisions [4].

THE TEN COMMANDMENTS [68]

1. It is imperative to acknowledge and delineate the specific component of a decision or action that is executed and implemented by artificial intelligence (AI).
2. It is imperative to ensure that there is clear distinction between the segments of communication that are executed by an artificial intelligence agent.
3. The accountability for an artificial intelligence (AI) decision, action, or communicative process must be assumed by a capable individual or entity, whether it be a physical or legal person.
4. It is imperative that decisions, actions, and communicative processes carried out by artificial intelligence (AI) systems adhere to principles of transparency and explainability.
5. In order for an AI decision to be considered valid, it is important that it possesses the qualities of comprehensibility and repeatability.
6. The elucidation of an artificial intelligence (AI) choice necessitates a foundation rooted on contemporary and scientifically advanced theories.
7. It is imperative that each choice, action, or communication undertaken by an artificial intelligence system refrain from engaging in manipulative behavior by feigning accuracy.
8. An AI decision, action, or communication must adhere to all relevant legal regulations and must not result in any harm to individuals.
9. AI decisions, actions, or communications must adhere to the principle of non-discrimination. This is especially relevant in the context of algorithmic training.
10. The responsibility for defining targets, exercising control, and monitoring the decisions, activities, and communications of AI systems should not be delegated to algorithms.

Following the assertive declaration in the first commandment, which emphasizes the imperative for an AI algorithm to refrain from concealing its existence, thereby facilitating open dialogue, examination, and evaluation of its pertinence,

hazards, efficiency, and efficacy, the second commandment pertains to the interaction between humans and AI agents, underscoring the necessity for humans to be shielded from deception by said agents. The number provided by the user is 20. In the context of healthcare, the customary procedure of obtaining informed consent necessitates that patients are provided with the necessary information to make an informed choice, particularly before any treatment alternatives are pursued. This phenomenon holds particular relevance in contemporary times, as the more intricate process of shared decision-making between healthcare providers and patients is gaining widespread acceptance as a customary approach. Therefore, in the event that automated processing is incorporated into the decision-making procedure, it is imperative to ensure that the patient is well informed. The objective is not accomplished by the use of the term "machine decision," but rather through the utilization of a specification, which entails an AI-supported decision, diagnostic finding, or therapy recommendation. The topic of accountability, as discussed by Floridi and Sanders in their scholarly article on autonomous actors, is examined in Commandments 3 and 4. The number provided by the user is 18. Grodzinsky et al. expanded upon Floridi's conceptual framework by introducing two additional levels of abstractions, namely LoA1 (the user view) and LoA2 (the designer view). The purpose of this extension was to explore the following inquiry: "Is it possible for an artificial agent, which possesses the capability to modify its own programming, to attain such a high level of autonomy that the original designer can no longer be held accountable for the agent's behavior?" The number being referred to is 19. Commandments 5–7 encompass fundamental ideas derived from the realms of scientific practice and medical ethics, which are applicable to AI agents. The eighth commandment is technically valid. Nonetheless, the practical implementation of this concept poses challenges and is contingent upon the specific legal framework governing the accessibility of medical bots on the internet. Furthermore, the effectiveness of enforcement is intricately linked to navigating the complex landscape of medical laws, regulations, and judicial rulings. According to Article 22 of the General Data Protection Regulation (GDPR), the utilization of automated processing is permissible solely upon obtaining explicit consent from the data subject, where it is deemed essential for the execution of a contractual agreement, or when it is approved by the legislation of the European Union or a member state. In order to protect people' rights, it is imperative to implement sustainable measures, which include, but are not limited to, the right to contest a decision. Commandment 9 pertains to the concepts of algorithmic fairness and bias. The composition of training data and its corresponding categories must be given special attention when considering that outputs generated by AI systems are influenced by the data sets, they are trained on. Race and skin color serve as prominent illustrations of algorithmic prejudice, as evidenced by the resources provided in the "Helpful Links" section. Nevertheless, this principle can also be extended to encompass several aspects, including but not limited to gender, age, income, origin, and education. It is imperative to acknowledge that customized medicine is inherently discriminatory in its nature, as it aims to categorize patient populations in order to enhance the provision of healthcare. This phenomenon gives rise to many pertinent differentiations among subsets of individuals. It is imperative to subject any algorithm to rigorous testing in order to identify and mitigate biases to the greatest extent feasible. Furthermore, anybody utilizing the algorithm must possess a comprehensive understanding of its inherent limits. The utilization of training data sets has significant importance in the advancement of AI solutions, necessitating its representation of the broader community to ensure equitable benefits for all individuals. In general, minority groups tend to have lower levels of representation, and the health concerns faced by these populations may be less apparent to developers of artificial intelligence (AI) solutions. Commandment 10 asserts that the creation and evaluation of machines should not occur without the involvement of human participation. In a web-based poll, a total of 121 experts were solicited for their opinions. Among the participants, 50.4% identified as female. The survey sample consisted of 47% computer experts and 33% medical doctors. The experts were queried regarding their perspectives on the significance and relevance of a set of 10 commandments. The noteworthy finding is that there is a general consensus among computer specialists and medical practitioners regarding the significance

and relevance of almost all commandments, save for commandment 6. This particular commandment states that an elucidation of an AI judgment should be grounded in contemporary scientific theories. The primary contention made herein is that the determination of explanations should be grounded in evidence-based and theory-based methodologies.

7 USER INTERFACE

We need a user-friendly interface that displays both diagnostic outcomes and the generated explanations. The interface should be intuitive for medical professionals to interact with, fostering trust and understanding [5]. The implementation of electronic health record systems, exemplified by the NHS Care Records Service [45], has revealed critical insights into the dynamics of user interaction within medical environments. A central revelation is that while these systems were initially conceived with a clinical-centric perspective, their primary users in the early stages often included allied health professionals and administrative staff. Their interests and concerns, however, were frequently overlooked in the implementation process, leading to usability challenges and inefficiencies.

7.1 User-Centered Approach

An ideal user interface for a medical system must adopt a user-centered design approach. This entails active engagement with all stakeholders, including clinical, administrative, and allied health professionals, from the inception of the system's development. Understanding the unique workflow and information needs of each user group is paramount to creating an interface that seamlessly integrates into their daily routines.

7.2 Flexibility and Adaptability

As observed in the case of the NHS Care Records Service, the ability to adapt and reconfigure the system to align with local practices of care delivery proved crucial in mitigating early frustrations. This underscores the importance of building flexibility into the interface, allowing users to customize workflows and adapt the system to their specific clinical contexts. Such adaptability empowers users to overcome usability challenges and optimize their interaction with the system.

7.3 Streamlined Data Entry

Efficient data entry is a cornerstone of any effective medical system interface. It is imperative that the interface facilitates the swift and accurate recording of patient information. This includes intuitive input mechanisms, intelligent auto-fill features, and structured templates that align with the natural progression of clinical encounters. Furthermore, concurrent data entry during patient interactions should be supported to enhance real-time documentation.

7.4 Minimized Administrative Burden

The experience of clinicians being compelled to take on additional administrative tasks due to system limitations highlights the need for interfaces that alleviate, rather than exacerbate, administrative burdens. An ideal interface should automate routine tasks, such as data entry and retrieval, to allow healthcare professionals to focus their time and expertise on patient care.

7.5 Seamless Integration with Clinical Workflows

To achieve widespread acceptance and adoption, a medical system's interface should seamlessly integrate with existing clinical workflows. This includes interoperability with ancillary systems and devices, as well as the provision of clear pathways for information exchange between different healthcare providers and specialties.

7.6 Usability Testing and Continuous Improvement

Usability testing and ongoing user feedback mechanisms are indispensable in refining the interface of a medical system. Regular evaluations, conducted with representative end-users, can identify pain points, uncover latent needs, and drive iterative improvements. This iterative process ensures that the interface evolves in tandem with the dynamic demands of clinical practice.

In conclusion, an ideal user interface for a medical system should be rooted in a user-centered design philosophy, characterized by flexibility, streamlined data entry, reduced administrative burden, seamless workflow integration, and a commitment to continuous improvement through user feedback. By prioritizing these design considerations, medical systems can enhance user satisfaction, optimize clinical workflows, and ultimately improve the quality of patient care.

8 EVALUATION MEASURES

The evaluation metrics for Explainable Artificial Intelligence (XAI) systems are a crucial consideration throughout the design phase of such systems. Explanations serve the purpose of addressing various objectives related to interpretability, and hence require diverse criteria to assess the accuracy and reliability of the explanations for their intended use. One typical method for evaluating AI newbie end-users is through experimental design including human-subject investigations. XAI evaluation has utilized a range of controlled in-lab and online crowdsourced research. Case studies seek to get input from domain expert users while they engage in complex cognitive activities using analytics tools. In contrast, computational measurements are specifically developed to assess the precision and comprehensiveness of explanations generated by interpretable algorithms. This section provides a comprehensive analysis and classification of the primary assessment metrics used for evaluating XAI systems and algorithms. We offer condensed and readily applicable XAI evaluation metrics and techniques derived from the literature, presented in Tables 2–6.

8.1 M1: Mental Model

According to cognitive psychology theories, a mental model refers to a conceptual representation of how users comprehend a system. Human-Computer Interaction (HCI) researchers analyze users' cognitive frameworks to assess their comprehension of intelligent systems across different applications. Costanza et al. [69] examined the comprehension of users about a smart grid system, whereas Kay et al. [70] investigated the comprehension and adjustment of consumers towards uncertainty in machine learning predictions of bus arrival times. Within the realm of Explainable Artificial Intelligence (XAI), explanations serve the purpose of enabling users to construct a cognitive representation of the underlying mechanisms of the AI system. Machine learning explanation serves as a means to assist consumers in constructing a more precise cognitive representation. Examining users' cognitive frameworks of Explainable Artificial Intelligence (XAI) systems can assist in validating the efficacy of explanations in elucidating the decision-making process of an algorithm. Table 2 provides a concise overview of various evaluation approaches employed to assess users' cognitive representation of machine learning models. Research in psychology on human-AI interactions has examined the structure, types, and functions of explanations to identify the key components of an ideal explanation that promotes greater user comprehension and more precise mental models [71, 72]. Lombrozo [73] examined the impact of various explanation styles on the organization of conceptual representation. To determine the appropriate method for an intelligent system to clarify its actions to individuals without expertise, investigations on the explanations provided by machine learning have

examined how users comprehend intelligent agents [74, 75] and algorithms [76] to ascertain the expectations users have for machine explanations. In relation to this, Lim and Dey [77] investigate the various forms of explanations that users may anticipate in four practical applications. Their research focuses on analyzing the precise forms of explanations that users require in various contexts, such as system recommendation, crucial events, and unexpected system behavior. Bansal et al. [78] developed a game to assess user mental model by predicting model failure. Participants in the game are rewarded with monetary incentives based on their final performance score. The trials conducted on a basic three-dimensional assignment demonstrate that as the data and model become more complex, users' capacity to anticipate model failure diminishes. An effective method for assessing users' understanding of intelligent systems is to directly inquire about the decision-making process employed by the system. Examining users' interviews, think-alouds, and self-explanations yields useful insights on their cognitive processes and conceptual frameworks [79]. Kulesza et al. [80] conducted a study on user understanding, specifically focusing on how the accuracy and thoroughness of explanations affect the accuracy of end-users' mental model in a music recommendation interface. Their findings indicated that the extent of explanation completeness had a greater impact on user comprehension of the agent, in comparison to the level of explanation soundness. Binns et al. [81] examined how different types of machine explanations affect consumers' perception of fairness in algorithmic decision-making. During the design cycles of interpretable interfaces for intelligent systems, it is important to take into account the user's attention and expectations [82]. The desire to create and assess explanations that can be easily understood by humans has also resulted in the development of interpretable models and specialized tools for assessing mental models. As an illustration, Ribeiro et al. [10] assessed users' comprehension of the machine learning method using visual explanations. The researchers demonstrated how explanations might reduce human overestimation of the effectiveness of an image classifier and assist users in selecting a more effective classifier by relying on the provided explanations. In a subsequent study, the researchers conducted a comparison between the global explanations and instance explanations of a classifier model. They discovered that global explanations were more proficient in identifying the shortcomings of the model [83]. Kim et al. [84] conducted a crowdsourced survey to assess the comprehensibility of feature-based explanations for end-users in a separate publication. In their study, Lakkaraju et al. [85] used interpretable decision sets as a model representation to enhance comprehension. They evaluated users' mental models using various metrics, including user accuracy in predicting machine output and the length of users' self-explanations.

Mental Model Measures	Evaluation Methods
User Understanding of Model	Interview ([69]) and Self-explanation ([81,74,75])
	Likert-scale Questionnaire ([84, 80, 85, 77, 73, 76])
Model Output Prediction	User Prediction of Model Output ([70, 10, 83])
Model Failure Prediction	User Prediction of Model Failure ([78, 86])

Table 2: Assessment Metrics and Techniques Employed in Analyzing User Cognitive Frameworks in Explainable Artificial Intelligence (XAI) Systems.

8.2 M2: Explanation Usefulness and Satisfaction

End-user satisfaction and usefulness of machine explanation are also of importance when evaluating explanations in intelligent systems [87]. Researchers use different subjective and objective measures for understandability, usefulness, and sufficiency of details to assess explanatory value for users [88]. Although there are implicit methods to measure user satisfaction [89], a considerable part of the literature follows qualitative evaluation of satisfaction in explanations, such as

questionnaires and interviews. For example, Gedikli et al. [90] evaluated 10 different explanation types with user ratings of explanation satisfaction and transparency. Their results showed a strong relationship between user satisfaction and perceived transparency. Similarly, Lim et al. [91] explore explanation usefulness and efficiency in their interpretable context-aware system by presenting different types of explanations such as "why," "why not," and "what if" explanation types and measuring users' response time. Another line of research studies whether intelligible systems are always appreciated by the users or it has a conditional value. An early work from Lim and Dey [77] studied user understanding and satisfaction of different explanation types in four real-world context-aware applications. Their findings show that, when considering scenarios involved with criticality, users want more information explaining the decision-making process and experience higher levels of satisfaction after receiving these explanations. Similarly, Bunt et al. [92] considered whether explanations are always necessary for users in every intelligent system. Their results show that, in some cases, the cost of viewing explanations in diary entries like Amazon and YouTube recommendations could outweigh their benefits. To study the impact of explanation complexity on users' comprehension, Lage et al. [93] studied how explanation length and complexity affect users' response time, accuracy, and subjective satisfaction. They also observed that increasing explanation complexity resulted in lowered subjective user satisfaction. In a recent study, Coppers et al. [94] also show that adding intelligibility does not necessarily improve user experience in a study with expert translators. Their experiment suggests that an intelligible system is preferred by experts when the additional explanations are not part of the translators readily available knowledge. In another work, Curran et al. [96] measured users' understanding and preference of explanations in an image recognition task by ranking and coding user transcripts. They provide three types of instance explanations for participants and show that although all explanations were coming from the same model, participants had different levels of trust in explanations' correctness, according to explanations clarity and understandability. Table 3 summarizes the study methods used to measure user satisfaction and usefulness of machine learning explanations. Note that the primary goal of XAI system evaluations for domain and AI experts is through direct evaluation of user satisfaction of explanation design during the design cycle. For example, case studies and participatory design are common approaches for directly including expert users as part of the system design and evaluation processes.

Satisfaction Measures	Evaluation Methods
User Satisfaction	Interview and Self-report ([92, 90, 77, 91])
	Likert-scale Questionnaire ([94, 90, 93, 77, 91])
	Expert Case Study ([96, 97, 98, 99, 100])
Explanation Usefulness	Engagement with Explanations ([94])
	Task Duration and Cognitive Load ([90, 93, 91])

Table 3: Measures of user satisfaction and study methods employed in assessing user satisfaction and the effectiveness of explanations in XAI research.

8.3 M3: User Trust and Reliance

User trust in an intelligent system is a factor that affects both emotions and thoughts, and it can influence whether someone has a favorable or negative opinion of the system [101, 102]. The concepts of initial user trust and the evolution of trust over time have been examined and described using various terminology, including rapid trust [103], default trust [104], and suspicious trust [105]. The initial state of trust is influenced by prior information and beliefs. However, trust and confidence can be altered by investigating and pushing the system with edge situations [106]. Consequently, the user's

level of trust and distrust may vary at different stages of their engagement with a particular technology. Trust is defined and quantified by researchers using various methodologies. When examining trust, it is typical to consider factors such as the user's knowledge, technical skills, familiarity, confidence, beliefs, faith, emotions, and personal connections [107, 102]. To assess user trust and dependence, one can employ direct methods such as conducting interviews and administering questionnaires to get user perspectives during and after their interaction with a system. As an illustration, Yin et al. [108] examined the significance of model correctness in relation to user trust. Their research demonstrates that user confidence in the system was influenced by both the system's explicitly stated correctness and users' subjective perception of accuracy as time progressed. In a similar vein, Nourani et al. [109] investigated the impact of incorporating explanations and the degree of meaningfulness on the user's sense of correctness. Their empirical study demonstrates that the comprehensibility of explanations can have a substantial impact on the perceived accuracy of a system, regardless of the actual accuracy seen during system operation. Furthermore, trust evaluation scales could be tailored to the particular application context of the system and the design objectives of explainable artificial intelligence (XAI). For example, many scales would evaluate user perception of system reliability, predictability, and safety as distinct factors. In relation to this, the study by Cahour and Forzy [110] presents a comprehensive trust measuring setup. This setup utilizes various trust scales (trust constructs), video recording, and self-confrontation interviews to assess user trust. The aim is to analyze three different ways of system presentation. In order to gain a deeper comprehension of the aspects that impact confidence in adaptive agents, Glass et al. [111] conducted a study to determine the specific types of questions that users would desire to ask an adaptive assistant. Prior studies have examined the evolution of user awareness by presenting the system's level of confidence and uncertainty in the outputs of machine learning in applications of varying levels of importance [112, 113]. Several studies have examined the influence of XAI on cultivating justifiable trust among users in various fields. For example, Pu and Chen [169] introduced an organizational framework to produce explanations and assessed perceived competence and user's intention to return as indicators of user trust. Another instance involved a comparison of user trust with explanations for various objectives, such as transparency and justification explanation [114]. The researchers utilized perceived understandability as a metric to gauge user trust and demonstrated that explicit explanations can mitigate the adverse consequences of trust erosion under unforeseen circumstances. Berkovsky et al. [115] assessed user trust in real-world apps by examining different recommendation interfaces and content selection methodologies. The researchers assessed user dependence on a movie recommendation system using six separate trust factors. In addition, Eiband et al. [118] replicate the experiment conducted by Langer et al. [116] on the impact of "placebic" explanations (explanations that do not communicate any information) on the thoughtless behavior of users. The researchers investigated if offering placebo explanations would enhance user dependence on the recommender system. Their findings indicate that future research on explanations for intelligent systems should explore the use of placebo explanations as a reference point for evaluating machine learning generated explanations. Bussone et al. [117] conducted a study to assess the trust of expert users in a clinical decision-support system. They measured trust using Likert-scale and think-alouds. The study indicated that providing explanations of facts resulted in increased user trust and reliance on the system. Table 4 presents a compilation of subjective and objective assessment techniques for quantifying user trust in machine learning systems and their accompanying explanations. User trust is often assessed as a fixed characteristic in numerous research. Nevertheless, it is crucial to consider the user's expertise and accumulated knowledge when dealing with intricate AI systems. Gathering repeated measurements throughout time can aid in comprehending and analyzing the pattern of consumers' trust development as their experience progresses. In their study, Holliday et al. [89] assessed trust and reliance at various phases of collaborating with an explainable text-mining technology. They demonstrated that the degree of user confidence in the system fluctuated over time as the user acquired additional expertise and familiarity with the system. It is worth mentioning

that our literature review did not uncover a prevalent emphasis on directly measuring trust in analysis tools for data and machine learning experts. However, the evaluation process in case studies often takes into account users' dependence on tools and their inclination to continue using them. To clarify, our summarization does not assert that data specialists disregard trust. Rather, we have not discovered it to be a fundamental result that is directly assessed in the literature for this particular user group.

Trust Measures	Evaluation Methods
Subjective Measures	Self-explanation and Interview ([117, 110])
	Likert-scale Questionnaire ([115, 117, 110, 109])
Objective Measures	User Perceived System Competence ([109, 113, 108])
	User Compliance with System ([118])
	User Perceived Understandability ([114, 108])

Table 4: Assessment Metrics and Techniques Employed for Evaluating User Confidence in XAI Research.

8.4 M4: Human-AI Task Performance

A key goal of XAI is to help end-users to be more successful in their tasks involving machine learning systems [90]. Thus, human-AI task performance is a measure relevant to all three groups of user types. For example, Lim et al. [91] measured users' performance in terms of success rate and task completion time to evaluate the impact of different types of explanations. They use a generic interface that can be applied to various types of sensor-based context-aware systems, such as weather prediction. Further, explanations can assist users in adjusting the intelligent system to their needs. Kulesza et al. [119] study of explanations for a music recommender agent found a positive effect of explanations on users' satisfaction with the agent's output, as well as on users' confidence in the system and their overall experience. Another use case for machine learning explanations is to help users judge the correctness of system output [120, 121, 122]. Explanations also assist users in debugging interactive machine learning programs for their needs [123, 79]. In a study of end-users interacting with an email classifier system, Kulesza et al. [123] measured classifier performance to show that explanatory debugging benefits user and machine performance. Similarly, Ribeiro et al. [10] found users could detect and remove wrong explanations in text classification, resulting in training better classifiers with higher performance and explanations quality. To support these goals, Myers et al. [139] designed a framework that users can ask why and why not questions and expect explanations from the intelligent interfaces. Table 5 summarizes a list of evaluation methods to measure task performance in human-AI collaboration and model tuning scenarios. Visual analytics tools also help domain experts to better perform their tasks by providing model interpretations. Visualizing model structure, details, and uncertainty in machine outputs can allow domain experts to diagnose models and adjust hyper-parameters to their specific data for better analysis. Visual analytics research has explored the need for model interpretation in text [124, 98, 125] and multimedia [126, 127] analysis tasks. This body of work demonstrates the importance of integrating user feedback to improve model results. An example of a visual analytics tool for text analysis is TopicPanaroma [128], which models a textual corpus as a topic graph and incorporates machine learning and feature selection to allow users to modify the graph interactively. In their evaluation procedure, they ran case studies with two domain experts: a public relations manager used the tool to find a set of tech-related patterns in news media, and a professor analyzed the impact of news media on the public during a health crisis. In analysis of streaming data, automated approaches are error-prone and require expert users to review model details and uncertainty for better decision making [129, 130]. For example, Goodall et al. [131] presented Situ, a visual analytics

system for discovering suspicious behavior in cyber network data. The goal was to make anomaly detection results understandable for analysts, so they performed multiple case studies with cybersecurity experts to evaluate how the system could help users to improve their task performance. Ahn and Lin [138] present a framework and visual analytic design to aid fair data-driven decision making. They proposed FairSight, a visual analytic system to achieve different notions of fairness in ranking decisions through visualizing, measuring, diagnosing, and mitigating biases. Other than domain experts using visual analytics tools, machine learning experts also use visual analytics to find shortcomings in the model architecture or training flaws in DNNs to improve the classification and prediction performance [99, 132]. For instance, Kahng et al. [96] designed a system to visualize instance-level and subset-level of neuron activation in a long-term investigation and development with machine learning engineers. In their case studies, they interviewed three machine learning engineers and data scientists who used the tool and reported the key observations. Similarly, Hohman et al. [137] present an interactive system that scalably summarizes and visualizes what features a DNN model has learned and how those features interact in instance predictions. Their visual analytic system presents activation aggregation to discover important neurons and neuron-influence aggregation to identify interactions between important neurons. In the case of recurrent neural networks (RNNs), LSTMVis [100] and RNNVis [133] are tools to interpret RNN models for natural language processing tasks. In another recent paper, Wang et al. [206] presented DNN Genealogy, an interactive visualization tool that offers a visual summary of DNN representations. Another critical role of visual analytics for machine learning experts is to visualize model training processes [135]. An example of a visual analytics tool for diagnosing the training process of a deep generative model is DGMTracker [136], which helps experts understand the training process by visually representing training dynamics. An evaluation of DGMTracker was conducted in two case studies with experts to validate efficiency of the tool in supporting understanding of the training process and diagnosing a failed training process.

Performance Measures	Evaluation Methods
User Performance	Task Performance ([120, 96, 79, 91])
	Task Throughput([79, 85, 91])
	Model Failure Prediction ([120, 121, 122])
Model Performance	Model Accuracy ([123, 99, 132, 10, 122])
	Model Tuning and Selection ([128])

Table 5: Assessment Metrics and Techniques Employed in Evaluating Human-machine Task Performance in XAI Research

8.5 M5: Computational Measures

Computational metrics are widely used in the field of machine learning to assess the accuracy and comprehensiveness of interpretability strategies in conveying the knowledge acquired by the model. Herman [140] argues that depending on human assessment of explanations can result in compelling explanations instead of transparent systems, as users tend to prefer simplified explanations. Hence, this issue raises the contention that the accuracy of explanations in adhering to the black-box paradigm should be assessed using computational techniques rather than human-subject investigations. Fidelity in the context of an adhoc explainer pertains to the accuracy of the ad-hoc technique in producing accurate explanations, such as the accuracy of a saliency map, for the predictions made by the model. This results in a range of computational strategies for assessing the accuracy of generated explanations, the coherence of explanation outcomes, and the faithfulness of ad-hoc interpretability methods to the original black-box model [141]. Machine learning researchers frequently regard consistency in explanation outcomes, computational interpretability, and qualitative self-interpretation of results as

indicators of explanation accuracy [142, 143, 144, 145]. Zeiler and Fergus [146] examine the integrity of the visualization for a CNN network by assessing its validity in identifying model shortcomings, which leads to improved prediction outcomes. Alternatively, in other instances, the evaluation of explanation quality involves the comparison of a novel explanation technique with established state-of-the-art techniques [147, 148, 47]. For example, Ross et al. [149] conducted a series of empirical assessments and compared the consistency and computational cost of their explanations with the LIME technique [10]. Samek et al. [160] introduced a complete framework for assessing saliency explanations of picture data. This framework measures the significance of features in relation to the classifier's prediction. The researchers conducted a comparison of three distinct methods for explaining saliency in image data: sensitivity-based [150], deconvolution [146], and layerwise relevance propagation [151]. They also examined the relationship between the quality of saliency maps and the performance of neural networks on various image datasets when subjected to input perturbation. In contrast, Kindermans et al. [152] demonstrate that interpretability techniques exhibit inconsistencies when applied to basic picture alterations, thereby resulting to potentially deceptive saliency maps. An input invariance property is defined to ensure the reliability of explanations generated by saliency algorithms. In a related concept, Adebayo et al. [161] suggest three assessments to gauge the sufficiency of interpretability methods for tasks that are influenced by either the data or the model itself. Additional evaluation techniques involve measuring the accuracy of explanations in relation to models that are intrinsically interpretable, such as linear regression and decision trees. As an illustration, Ribeiro et al. [112] conducted a comparison between explanations produced by the LIME ad-hoc explainer and explanations derived from an interpretable model. In their work, they generated explanations of the highest quality straight from interpretable models such as sparse logistic regression and decision trees, which were then utilized for comparisons. One drawback of this approach is that the evaluation is restricted to producing a gold standard using an interpretable model. User simulated evaluation is an alternative approach for doing computational examination of model explanations. Ribeiro et al. [93] conducted a simulation to assess user trust in explanations and models by establishing criteria for identifying "untrustworthy" explanations and models. An experiment was conducted to examine whether actual users would have a preference for more dependable explanations and select superior models. The authors subsequently conducted comparable user-simulated assessments in the Anchors explanation approach [162] to document the simulated users' precision and coverage in identifying the superior classifier only based on explanations. Schmidt and Biessmann [163] have adopted a distinct method for measuring the quality of explanations using human intuition. They have devised a metric for explanation quality that is based on the time it takes for users to complete a task and the level of agreement in predictions. Another illustration involves the research conducted by Lundberg and Lee [148], in which they contrasted the SHAP ad-hoc explainer model with LIME and DeepLIFT [47]. Their underlying assumption was that effective model explanations should align with the explanations provided by individuals who comprehend the model. Lertvittayakumjorn and Toni [118] propose three user tasks for assessing local explanation methods in text classification. These activities involve exposing model behavior to users, providing justifications for predictions, and assisting users in exploring uncertain predictions. In [154], a comparable concept was executed using the featurewise comparison of a ground-truth and model explanation. They offer a standard that is annotated by users to assess the explanations of machine learning instances. Subsequently, Poerner et al. [155] employ this benchmark as a human-annotated reference point for evaluating explanations at both the word level (small-context) and the phrase level (large-context). When evaluating the meaningfulness of explanations, benchmarks that are annotated by humans might provide significant insights. However, the discussion by Das et al. [157] suggests that machine learning models, specifically visual question answering attention models in their study, do not focus on the same regions as people. The researchers present a dataset focused on human attention, which includes mouse-tracking data. They next assess the accuracy of attention maps produced by advanced algorithms by comparing them to human-generated maps.

Interpretability techniques facilitate the assessment of model trustworthiness by providing quantitative measures, such as model fairness, reliability, and safety, through explanations. The trustworthiness of a model encompasses domain-specific objectives, such as achieving fairness through fair feature learning, ensuring reliability, and promoting safety through robust feature learning. As an illustration, Zhang et al. [156] demonstrate the application of machine learning explanations to identify problems in representation learning that arise from potential biases in the training dataset. Their methodology extracts the connections between pairs of attributes based on their patterns of inference. In addition, Kim et al. [84] conducted quantitative testing of machine learning models based on their explanations. The idea activation vector technique allows for testing the model's bias towards specific concepts, such as picture patterns, by generating a vector score. Later on, they expanded their idea of using a concept-based global explanation to learn how models represent information. This approach helps to systematically uncover ideas that are both meaningful to humans and significant for the model's predictions [158]. The researchers conducted human-subject experiments to assess acquired notions. Table 6 presents a compilation of assessment approaches for quantifying the accuracy of interpretability techniques and the reliability of computational methods.

Computational Measures	Evaluation Methods
Explainer Fidelity	Simulated Experiments ([172, 173])
	Sanity Check ([101, 162, 177, 215, 217, 226]
	Comparative Evaluation ([178, 183])
Model Trustworthiness	Debugging Model and Training ([219])
	Human-Grounded Evaluation ([43, 134, 149, 187])

Table 6: Evaluation measures and methods are employed to assess the fidelity of interpretability techniques and the reliability of trained models. Machine learning and data specialists utilize this set of evaluation tools to assess the accuracy of interpretability techniques or to evaluate the training quality of models beyond traditional performance metrics.

9 XAI ARCHITECTURE

Figure 10 depicts our XAI architecture that is not specific to any particular area. The work by Gomboc et al. (2005) [164] primarily addressed the left side of the diagram and the challenge of importing data from the simulation. In order to accommodate the substantial log files generated by tactical military simulations, which capture data at a high level of detail (e.g., OOS records variable values approximately every second), we employ a relational database for information storage. The Virtual Humans presently store a limited amount of data, but the complete capabilities of the relational database could prove valuable in the future if the data logging in Virtual Humans gets more detailed. We concentrate on a certain set of behaviors exhibited by the virtual human. If we were to include the complete spectrum, the resulting log files would be of similar size to those found in tactical simulations. Once XAI has successfully imported the required information from the simulation, the subsequent task involves scrutinizing the data to identify noteworthy and significant facts. The meaning of "interesting" will vary depending on the context in which it is used. In the context of OOS, we employed a temporary description that emphasized the occurrence of an entity discharging its weapon or commencing, reaching the midpoint, or concluding a task. Engaging in collaboration with specialists in the field will assist us in further refining this definition. As creators of an automated tutor for the Virtual Humans, our understanding of "interesting" pertains to the occurrences that the tutor should engage the student in discussing. It is the tutor's duty to identify these events. Currently, we manually label these teaching points. However, we are in the midst of automating this task by employing a heuristic-based technique to

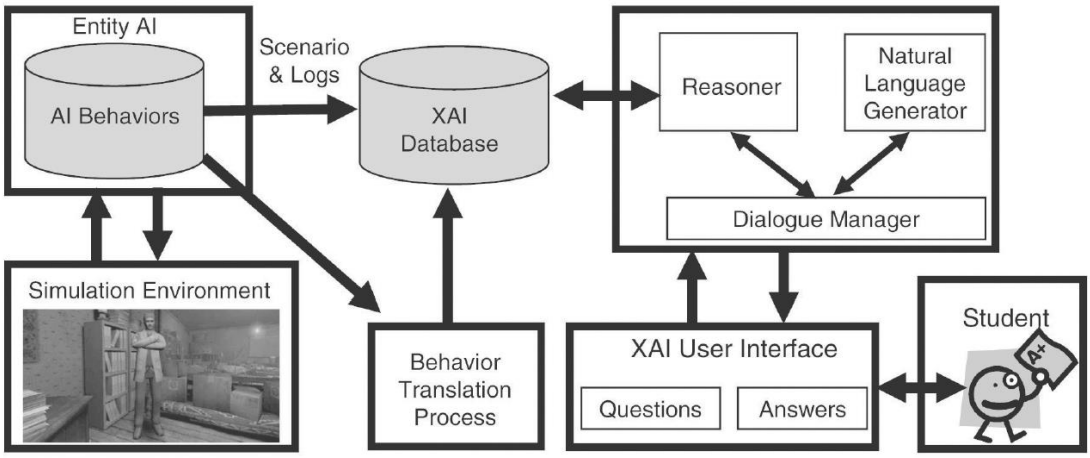


Figure 10: XAI Architecture

identify instructional moments in each activity. After initializing the XAI system, users choose an entity to communicate with and choose the initial time point in the simulation they want to discuss. Users inquire the entity about the present moment by choosing inquiries from a menu. The conversation manager coordinates the system's answer by utilizing the reasoner to obtain pertinent information and subsequently generating English responses through the natural language generator (NLG). Natural Language Generation (NLG) populates placeholders in linguistic templates with data retrieved from the database. Natural Language Generation (NLG) is implemented using XSL templates and utilizes XSL capabilities such as iteration and procedure calls. This allows for the sharing of common tasks, such as state descriptions, among templates. According to (Gomboc et al. 2005) [164], this refers to an ideal situation in which the simulation provides a comprehensive depiction of behavior, including entity goals, action preconditions, and effects. Nevertheless, the absence of a corresponding representation in the simulation does not preclude our ability to analyze and elucidate the behavior of entities. Prior research on elucidating the behavior of expert systems [165] addressed a comparable issue. In cases where the inference engine of the expert system possesses unique attributes that are not reflected in the system's knowledge base, the results produced by these attributes can still be explicated through pre-programmed explanation procedures. Nevertheless, if modifications are implemented to the distinctive attributes without concurrently revising the explanation procedures, there exists a possibility of providing erroneous explanations. Therefore, while explanations can still be provided, doing so will negatively impact the system's maintainability and robustness. When it comes to expressing simulation behaviors for Explainable Artificial Intelligence (XAI), there are three alternatives, each with its own costs and advantages:

1. Import the behaviors automatically. Cost: necessitates a depiction of objectives as well as the prerequisites and consequences of actions. Advantage: The system has a high level of maintainability and robustness. Target: representations based on plans

2. Import the behaviors in a semi-automated manner. Cost: The ability to identify goals, preconditions, and effects in behavior representation is required. Advantage: It is more manageable and resilient compared to option 3, and it relies on less assumptions compared to option 1. Target: Rule-based representations.

3. Construct the XAI representation of the behaviors manually. Cost: The maintainability is low as any modification in the behavior needs to be replicated in the XAI representation. Require a subject matter expert to compose the absent objectives, preconditions, and outcomes. Advantage: Does not make any assumptions regarding the representation of the simulation's activity. Target: Procedural representations

We choose to focus on rule-based representations with option 2 because certain components on the left side of the rules serve as prerequisites (for example, having ammunition to fire a weapon), while other components may act as termination conditions (for instance, refraining from firing a weapon when a friendly entity is in the trajectory) or internal administrative tasks (such as adjusting internal variables and preventing a rule from firing twice). Likewise, not all components located on the right side of rules are consequences and can also serve as internal record-keeping. By utilizing this particular model, we have the ability to manually annotate the preconditions and effects within these rules, and subsequently import the behavior automatically. While there is no assurance that annotations will be updated alongside changes in entity behaviors, at the very least this meta-data is located in close proximity to the original representation of behavior. Gathering log files and scenario information may be challenging, but the available choices are straightforward: if the simulation does not provide data for export, the XAI system will be unable to respond to inquiries regarding it. For our Explainable Artificial Intelligence (XAI) system designed for Out-of-Specification (OOS) analysis, we manually inputted scenario and log information to construct a preliminary system. However, this data is tailored to a single simulation run, and certain manual data-entry tasks need to be replicated for every subsequent simulation run. Hand-authoring behavior representations is a more practical approach because they remain consistent between simulation runs.

9.1 Building a new XAI System

To integrate our existing XAI system with a new simulation, a series of actions must be followed: 1. Analyze the depiction of behavior and select one of the three methods for including behaviors as explained in the preceding section. 2. Integrate data import functionality for behaviors and log files. 3. Define the question list for the specified domain. 3a. Provide the logical form (LF) for each question. 3b. Compose the LF query. 4. Enhance the capabilities of the natural language generator to accommodate the additional inquiries and their corresponding potential responses. 5. Develop a Graphical User Interface (GUI) To establish the question list for a new simulation, two procedures must be taken. To begin, one must compose the logical structure of the question, which serves as the basis for generating the English version of the query. In the case of the Virtual Humans, we had a total of 110 unique questions. By employing the natural language generator, we were able to modify the phrasing of these questions without having to rewrite all 110 of them. The next stage involves composing the query to extract the desired information from the database. This is accomplished by utilizing a specialized language known as the query logical form, which is used to encode inquiries (see to the information below for further clarification). To integrate XAI with a new simulation, the final task involves constructing a new graphical user interface (GUI) or repurposing an existing GUI from a prior XAI system. While all XAI systems share fundamental graphical user interface (GUI) components such as entity selection, time selection, question selection, and dialogue display between the user, XAI, and tutor, the ability to replay the simulation relies on the support of the target simulation. If XAI is integrated into the simulation as a feature, it will utilize the simulation's GUI. Due to these limitations, we developed the graphical user interfaces (GUIs) for XAI for OOS and XAI for Virtual Humans as distinct elements that interact with the rest of the system via XML messages. The play-and-plug capability is facilitated by our abstract message format. The messages transmit the information included in menus, such as the list of questions and the list of time points, as well as the choices made by the user from these menus. The messages additionally modify the current state of the conversation between the student and tutor, as well as the conversation between the student and XAI. The graphical user interface (GUI) has the

ability to provide a variety of menu options and text using several types of widgets, such as radio buttons and drop-down menus.

9.2 XAI for Virtual Humans

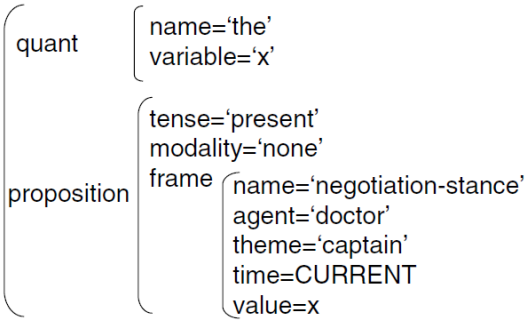


Figure 11: LF for "What is your negotiation stance?"

To connect XAI to the Virtual Humans, the initial stage involved studying the behavior representation, as outlined in the previous section. In this instance, the developers of the simulation were required to accurately represent not only the physical actions involved in treating patients, but also the underlying behaviors expressed through the student and doctor's speech (e.g., committing, insisting), as well as the doctor's cognitive processes (e.g., making the decision to assist the captain). The physical activities model incorporates preconditions and effects to elucidate the connections between actions (e.g., the requirement of supplies for patient treatment). Upon importing this model, we encountered several flaws in its design, thus it would be more precise to state that we imported the physical behaviors in a semi-automated manner. With the resolution of the software glitches, we should now have the capability to completely automate this procedure. The implementation of non-physical behaviors involved the utilization of numerous rules that were defined inside the Soar cognitive architecture [166]. With sufficient time, it should be feasible to manually annotate the objectives, preconditions, and outcomes in all of these rules. Initially, our implementation prioritized the regulations that regulate trust, as teaching trust development is one of the educational objectives of the 2005 Virtual Humans system. The Virtual Humans paradigm, as outlined in (Traum et al. 2005) [167], considers three characteristics - familiarity, credibility, and solidarity - that influence the establishment of trust. There is a direct correlation between trust and three factors: the doctor's familiarity with you, their perception of your honesty, and their belief in shared objectives. The doctor's level of confidence in you increases as these factors strengthen, and vice versa. Our current prototype represents individual stages of the doctor's thought process by connecting rules to English paraphrases, such as "the negotiation failed due to the loss of trust from the doctor." After formulating the database structure for the desired simulation and implementing the code to import the data, the subsequent stage entailed encoding the inquiries to be addressed by XAI in the specific field (such as defining the question itself, encoding the pertinent database queries, and making any required modifications to the natural language generation templates). Queries are encoded in a logical form (LF), which is an abstract representation of their content. Figure 11 displays a simplified graphical depiction of our XML format for the question, "What is your negotiation stance?" The logical form was created with the intention of facilitating future endeavors to construct syntactic attributes of the character's language, such as tense and modality, instead of embedding them directly into templates. Another notable

feature is the variable, CURRENT, which is replaced during runtime with the line now being discussed. It is evident that we would not create distinct questions like "what is the negotiation stance at line 1" and "what is the negotiation stance at line 2". Nevertheless, this same technique enables us to possess a single coherent structure for both inquiries: "what caused the failure of the negotiation?" and "what is the reason behind your avoidance of the negotiation?". In this context, the runtime variable represents the position taken throughout the negotiation. The question's logical form is supplemented by an abstract representation of the inquiry, referred to as the query LF, in order to extract the response. Additionally, it employs runtime variables, allowing developers to write a single query LF for both the questions "why did the negotiation fail?" and "why are you avoiding the negotiation?". The XAI reasoner converts the LF query into a SQL query, which is then transmitted to the database. One potential area for future research is to automatically generate the query logical form (LF) based on the LF of the question. Subsequently, the XSL templates need to be altered in order to generate the English version of the question and the many possible answers. Templates can be recycled to facilitate the inclusion of novel inquiries and their corresponding potential responses. For instance, there is a specific collection of templates that produces English explanations of states and tasks. These terms are employed to depict conditions and activities in inquiries as well as responses. Our future work aims to enhance the domain-independence of our natural language production by reducing the amount of hard-coded English and using templates that encode domain-independent features of language, such as syntax and morphology.

10 EXPLAINABLE AI REGULATIONS AND COMPLIANCE

The European Union has exerted substantial endeavors in policy communication and law concerning data and AI. However, both the GDPR and the proposed AI Act lack explicit provisions for the interpretability and explainability of these technologies for end users. While there is a certain focus on the requirement for explainability in order to facilitate oversight, the absence of technical reports and standards in this domain clearly indicates the necessity for additional progress and advancement. Within the United States, there exists a substantial allocation of resources towards research and development in the field of artificial intelligence (AI). While there are some discussions on the need of explainability in AI systems with regards to civil rights, the prevailing legislative framework primarily emphasizes the promotion of innovation rather than safeguarding human rights. The draft of the Algorithmic Accountability Act includes certain sections that address the rights of end-users to get explanations. However, the primary emphasis of the policy is still on fostering innovation. Within the United Kingdom, the legislative framework pertaining to artificial intelligence (AI) is primarily focused on data protection. However, there is a notable absence of well-defined enforcement mechanisms or uniformity when it comes to ensuring transparency and comprehensibility in AI systems. Presently, there is a predominant emphasis on offering direction to the sector and fostering innovation, while giving minimal consideration to safeguarding end-user rights and ensuring their protection. These priorities necessitate a more equitable strategy that harmonizes business interests with human rights and the imperative for reliable AI.

10.1 Key themes arising from our analysis

The government's approach to AI explainability has increasingly acknowledged the significance of interpretability and transparency in AI systems. There is a strong commitment to developing explainable AI systems that can be audited and held accountable within the broader governance framework of managing AI risks. The methodologies taken by NIST or ICO in explaining research on explainability and implementing it in organizations have not yet provided concrete regulatory requirements that may be acted upon. However, regulations like the EU GDPR, AI Act draft, AI Liability Directive proposal, and the US Accountability Act draft can be seen as initial legal foundations for implementing

explainability. Nevertheless, technical requirements would benefit from more precise specifications that encompass not only oversight but also end-user services. Challenges may arise when trying to establish a legal framework for explanations, considering factors such as casuistry and feasibility within the EU GDPR. These challenges involve finding a balance between criteria for explainability methods, the inherent complexity of models, the interests and expertise of stakeholders, and contextual organizational factors that may introduce additional tensions related to increased transparency. For instance, the use of interpretability as a design criterion may encounter opposition due to the EU rule concerning the protection of trade secrets and intellectual property rights. Although transparency is seen as favorable in terms of adhering to EU user rights, it might pose challenges when attempting to only maximize AI system openness without providing oversight and legal recourse for the same consumers. The determination of end-user responsibility can be established by considering the requirement, as stated in Article 29(1) of the AI Act, for providers to offer instructions to end-users regarding the appropriate usage of high-risk AI systems. Furthermore, Article 13(1) fails to consider the potential for third parties harmed by the AI system to utilize interpretability as a means of providing evidence. These organizational aspects, such as maintaining the confidentiality of trade secrets, protecting intellectual property rights, and ensuring the privacy of third parties, can serve as safeguards and provide motivation for firms to engage in research and development activities, without being constantly burdened by the need to provide explanations. Consequently, a company may perceive less risk if they are not required to provide explanations regarding the process and output of their AI system to users. This is because such explanations might potentially be used against them as evidence in legal disputes [178]. Alternatively, they may favor being subjected to transparent regulatory measures to indirectly evaluate the safety, fairness, and compliance with legal standards of AI models, notwithstanding their inherent complexity. Prior to that, laws appear to provide for the evaluation of transparency and the circumstances in which explanations are provided to end users, at the discretion of the provider. This can be seen in Article 13 of the EU AI Act, where it is only applicable to high-risk systems as determined by the provider. The criterion of "appropriate" in relation to transparency type implies a focus on legal sufficiency for supervision rather than a general end-user perspective. The primary obstacle lies in achieving a harmonious equilibrium between the intricacy of the model, the proficiency of the end-user, and the many legal and commercial limitations. The ambiguity may also arise from a broader sense of ambiguity in terminology, which was prevalent in the creation of communications and reports by governments and standardizing agencies that were commissioned for this purpose. For instance, we observed that NIST's publications on explainability offered a comprehensive study examination. However, it appeared that this did not support their Risk Management Framework or the White House's Blueprint. This ambiguity can be attributed to the absence of official coordination between agencies and their internal working groups. For instance, within IEEE, there are two separate working groups: one is responsible for developing a guide on AI explainability (C/AISC/XAI with P2894 [183]), while the other is focused on creating a standard (CIS/SC/XAI WG with P2976 [184]). In the subsequent discourse, we expound upon the tensions and constraints that we have identified in policies and norms, drawing on insights from academic research.

10.2 Considering the recent emergence of the concept of explainability in research

An important drawback of current documents is that, although they acknowledge the significance of explainability for the improvement of civil rights, their brief references in policy communications suggest a lack of a well-informed understanding of explainability as a research topic that is still in its early stages, innovative, intricate, and far from being a completely resolved issue. Currently, AI developers have challenges in implementing "explainability" in their systems due to the lack of clear definitions and established methodologies for achieving it. If an organization aims to establish criteria for the comprehensibility of a system, it is currently not feasible to do this task completely due to the specific reasons and

conflicts we outline below. Comprehension of the theoretical aspects of explainability. As the concept of explainability is still in its early stages, it is not yet apparent what criteria should be used to determine what constitutes a good explanation [175, 174, 176]. Prior research in Human-Computer Interaction (HCI) has demonstrated that the need for explanations varies depending on the context. Different stakeholders may require different types of explanations based on their specific objectives. This requirement is influenced by factors such as the domain of application and various human factors, including AI literacy and cultural background, which still require further investigation. Conversely, policy documents frequently reference the concept of explainability in AI policies, but fail to provide particular details regarding stakeholders, purpose, nature of explanations, and other relevant factors. Failure to define these ideas in policy, especially when their understanding from research is still incomplete, creates ambiguity that can have both positive and negative effects on the development and responsible use of new methods. Practical implementation capability of explainability. In practice, the designers and developers of an AI system may face challenges while constructing a system with the goal of explainability. Policy guidelines delineate three categories of explainability for an AI system, however the predominant focus in research publications has been solely on technical explainability. In addition, explainability approaches are specific to the data, task, and algorithm being used. now, not all AI systems being used in production have been linked to methodologies for explainability. This is because a significant portion of research is now centered in deep learning (DL) technology, while organizations still heavily depend on classic machine learning (ML) models. Therefore, one may not necessarily receive assistance in cultivating the appropriate forms of comprehensibility. The research community's narrow focus on algorithmic research, particularly in the field of deep learning (DL), is likely due to the way it is organized. This prioritization is evident in the rewards and publication opportunities offered, the methodologies that are emphasized and taught, the recognition from peers, and the incentives provided by organizations [177]. Moreover, the current explainability methodologies are widely recognized to be plagued by many problems that impede their practical application. These models are frequently low-fidelity [161], susceptible to several forms of disturbances like adversarial attacks [178, 179, 180], and inconsistent [89], sometimes failing to reveal all the problems (such as spurious correlations) that an ML model may experience [182]. However, researchers still face difficulties in creating improved explanations due to the unresolved difficulty of establishing suitable objectives and benchmarks for explainability [168]. The usability of explainability for stakeholders. Given the assumption that methods for achieving explainability in an AI system exist, further problems emerge. AI designers and developers may lack awareness of these methods, as there is a well-known disconnect between research and practice, not limited to AI. Consequently, they may struggle to effectively utilize these methods for their systems, as they may require varying levels of algorithmic knowledge, coding abilities, and so on [169, 170, 171]. Furthermore, it has been extensively researched that individuals who utilize the explanations generated by the explainability methods incorporated in the AI system may not receive adequate support to effectively utilize them [171, 173]. Furthermore, individuals may have insufficient expertise to comprehend these systems [80] and may succumb to pitfalls arising from diverse cognitive biases [174, 173, 168], resulting in unquestioning reliance on AI systems. Research has shown that individuals who receive explanations can be misled by customized explanations that appear to be based on facts and align with their existing opinions. This is achieved by taking advantage of confirmation bias or automation prejudice, as well as using illusions of explanatory depth. If an end-user were to utilize a biased and unfactual explanation as evidence during litigation, this may potentially result in significant financial consequences for the business. This scenario is mentioned in the EU AI Liability Directive.

10.3 Allocation of explanations accountability

In addition to the aforementioned theoretical and practical issues associated with the development and utilization of explainable AI systems, there are also concerns related to organizational barriers in achieving explainable AI. Multiple tensions arise when attempting to create an explainable system. According to NIST, it is now widely recognized that there is a deep and convoluted relationship between explainability and other desirable features of an AI system, such as accuracy or privacy-preservation, at the algorithmic level [185] [186]. Organizations prioritize accuracy in order to enhance the effectiveness and efficiency of their business processes. However, this emphasis on correctness might hinder the ability to make models easily understandable. Service providers' discretion. In a broader sense, various publications have suggested that there are organizational factors [190, 191] that could hinder developers who want to make models more explainable. These factors include lack of incentives, limited time availability, and the high computational costs associated with generating explanations [192, 193]. The articles outline these characteristics as essential for achieving justice goals, but they can be readily transferred or applied to other contexts. Conversely, legislative papers such as Article 13(1) of the EU AI Act seem to give service providers the freedom to create AI systems that are explainable, particularly with regards to providing detailed, easily understandable, and practical explanations. Parallel to the ongoing discussions on the rules of AI with regards to fairness [187], disinterested providers may opt for the simplest solutions, which may not adequately assist the achievement of explainability in practice. Opportunities for gaming and the practice of ethical washing. This technique can be attributed to the concept of ethics blue washing, which aims to ensure legal compliance by implementing the simplest option. The examined policies provide companies with the ability to exercise discretion. In this context, AI explanation typologies and targets can focus on specific aspects of data and system architecture that are regarded desirable and compliant. Explanations may be used to provide a general description of the attributes of an AI system, rather than to provide specific guidance on the actions that the decision-maker should take to address their situation, or on the underlying design principles that influenced the construction and upkeep of the AI system [199, 200]. Furthermore, the act of ethics dumping or selectively seeking justifications for organizational behavior based on local practices or laws, rather than internationally recognized standards set by reputable bodies, can be supported by stakeholders who possess more attractive codes of ethics and standards [188, 142]. Explanations should be implemented in accordance with the legal principles that impact the individuals receiving the explanations. This involves setting specific criteria to enforce measures of accountability, such as those outlined in the US Accountability Act and the EU AI liability drafts. However, it is important to note that complying with legal standards should not be equated with wholeheartedly adopting "ethics best practices" [196, 197, 198]. Future implementations of AI regulations, along with data and service regulations (such as the EU's enforcement of the Digital Services Act [189] in 2023), will serve as a valuable standard for assessing the accuracy of algorithmic explanations. This will hopefully help assign accountability for non-factual information.

11 CONCLUSION

This research article examined ways to improve model interpretability and transparency, with an emphasis on medical applications. This emphasizes the need to explain and justify particular cases, especially under changing conditions. The paper stressed the importance of prediction models in hospital infection management. The paper extensively covered the Extended Neural GPU, ByteNet, ConvS2S, and Transformer models, stressing self-attention for input and output representations. These models' pros and cons were examined, revealing their computational complexity and knowledge acquisition effects. SHAP values were suggested as a complete metric for measuring feature relevance in interpretable models. The paper described SHAP values' local correctness, missingness, and consistency. Shapley sampling values and Kernel SHAP were discussed to assess feature importance additively. AI ethics, especially in medical diagnosis, were

discussed. Fairness, transparency, and privacy were stressed to eliminate bias. A user-friendly design for showing diagnostic outcomes and explanations was also stressed, based on electronic health record system deployment lessons. The paper also examined Explainable Artificial Intelligence (XAI) system evaluation metrics, emphasizing the need for varied criteria to measure explanation accuracy and reliability. A complete analysis and classification of main assessment indicators laid the groundwork for XAI system and algorithm evaluation. Discussion of the XAI architecture presented a flexible framework not domain-specific. Using rule-based representations with a semi-automated approach, the article discussed XAI simulation behavior expression problems and options. Hand-authoring behavior representations for consistency between simulation runs was demonstrated by manually entering scenario and log data. The article finished with a critical analysis of EU, US, and UK AI laws. The research showed that these frameworks lack interpretability and explainability, underlining the need for a fairer strategy that combines commercial interests with human rights and ensures AI technology implementation. This paper contributes to the discussion on improving AI system transparency and interpretability, especially in healthcare.

ACKNOWLEDGMENTS

The successful execution of our project gives us an opportunity to convey our gratitude to each one who has been instrumental in paving the path to our continuation of this project. Whatever we have done is due to such guidance and help and we would not forget to thank them all. We would like to thank and seek the blessings from our Gurus – Bhagavan Shree Hari Chetana Narayana and Kala Bhaireshvara Bhagavan for their guidance on project-based learning and constructivist principles. We wholeheartedly thank our guide Professor Dr. Amber Wagner, Department of Computer Science, UAB for her support and guidance anytime we required. We also thank and share this moment of happiness with our parents who rendered us enormous support during the whole tenure of our research. Finally, may Dharma always prevail. Jai Shree Ram!

REFERENCES

[1] Vaswani, A. et al. (2017). Attention Is All You Need. In Proceedings of NeurIPS.

[2] Lundberg, S. M. and Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. In Proceedings of NeurIPS.

[3] Ribeiro, M. T. et al. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In Proceedings of KDD.

[4] Obermeyer, Z. and Emanuel, E. J. (2016). Predicting the Future - Big Data, Machine Learning, and Clinical Medicine. New England Journal of Medicine.

[5] Car, J. and Sheikh, A. (2003). Integrating health informatics and medical education. BMJ.

[6] General Data Protection Regulation (GDPR). (2018). Official Journal of the European Union.

[7] F. Wang and C. Rudin. Falling rule lists. In Artificial Intelligence and Statistics (AISTATS), 2015.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. Annals of Statistics, 32:407-499, 2004.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Computer Vision and Pattern Recognition (CVPR), 2015.

[10] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. 2016. “Why Should I Trust You?” Explaining the Predictions of Any Classifier arXiv:1602.04938v3. University of Washington.

[11] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative altering recommendations. In Conference on Computer Supported Cooperative Work (CSCW), 2000.

[12] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. The role of trust in automation reliance. Int. J. Hum.-Comput. Stud., 58(6), 2003.

[13] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison. Investigating statistical machine learning as a tool for software development. In Human Factors in Computing Systems (CHI), 2008.

[14] S. Kaufman, S. Rosset, and C. Perlich. Leakage in data mining: Formulation, detection, and avoidance. In Knowledge Discovery and Data Mining (KDD), 2011.

[15] J. Q. Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. Dataset Shift in Machine Learning. MIT, 2009.

[16] Tang, Z.; Chuang, K.V.; DeCarli, C.; Jin, L.W.; Beckett, L.; Keiser, M.J.; Dugger, B.N. Interpretable classification of Alzheimer's disease pathologies with a convolutional neural network pipeline. Nat. Commun. 2019, 10, 2173.

[17] Zhao, G.; Zhou, B.; Wang, K.; Jiang, R.; Xu, M. RespondCAM: Analyzing deep models for 3D imaging data by visualizations. In Medical Image Computing and Computer Assisted Intervention—MICCAI 2018; Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G., Eds.; Springer: Cham, Switzerland, 2018; pp. 485–492.

[18] Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv 2014, arXiv:1409.0473.

[19] Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

[20] Arras, L.; Horn, F.; Montavon, G.; Müller, K.; Samek,W. 'What is relevant in a text document?': An interpretable machine learning approach. arXiv 2016, arXiv:1612.07843.

[21] Hiley, L.; Preece, A.; Hicks, Y.; Chakraborty, S.; Gurram, P.; Tomsett, R. Explaining motion relevance for activity recognition in video deep learning models. arXiv 2020, arXiv:2003.14285.

[22] Eberle, O.; Buttner, J.; Krautli, F.; Mueller, K.-R.; Valleriani, M.; Montavon, G. Building and interpreting deep similarity models. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 44, 1149–1161.

[23] Thomas, A.W.; Heekeren, H.R.; Müller, K.-R.; Samek, W. Analyzing neuroimaging data through recurrent deep learning models. Front. Neurosci. 2019, 13, 1321.

[24] Burnham, J.P.; Rojek, R.P.; Kollef, M.H. Catheter removal and outcomes of multidrug-resistant central-line-associated bloodstream infection. Medicine 2018, 97, e12782.

[25] Fiala, J.; Palraj, B.R.; Sohail, M.R.; Lahr, B.; Baddour, L.M. Is a single set of negative blood cultures sufcient to ensure clearance of bloodstream infection in patients with Staphylococcus aureus bacteremia? The skip phenomenon. Infection 2019, 47, 1047–1053.

[26] Oonsivilai, M.; Mo, Y.; Luangasanatip, N.; Lubell, Y.; Miliya, T.; Tan, P.; Loeuk, L.; Turner, P.; Cooper, B.S. Using machine learning to guide targeted and locally-tailored empiric antibiotic prescribing in a children's hospital in Cambodia. Open Res. 2018, 3, 131.

[27] Hsu, C.N.; Liu, C.L.; Tain, Y.L.; Kuo, C.Y.; Lin, Y.C. Machine Learning Model for Risk Prediction of Community-Acquired Acute Kidney Injury Hospitalization From Electronic Health Records: Development and Validation Study. J. Med. Internet Res. 2020, 22, e16903.

[28] Greco, M.; Angelotti, G.; Caruso, P.F.; Zanella, A.; Stomeo, N.; Costantini, E.; Protti, A.; Pesenti, A.; Grasselli, G.; Cecconi, M. Artificial Intelligence to Predict Mortality in Critically ill COVID-19 Patients Using Data from the First 24h: A Case Study from Lombardy Outbreak. Res. Sq. 2021.

[29] Kim, K.; Yang, H.; Yi, J.; Son, H.E.; Ryu, J.Y.; Kim, Y.C.; Jeong, J.C.; Chin, H.J.; Na, K.Y.; Chae, D.W.; et al. Real-Time Clinical Decision Support Based on Recurrent Neural Networks for In-Hospital Acute Kidney Injury: External Validation and Model Interpretation. J. Med. Internet Res. 2021, 23, e24120.

[30] Kaji, D.A.; Zech, J.R.; Kim, J.S.; Cho, S.K.; Dangayach, N.S.; Costa, A.B.; Oermann, E.K. An attention based deep learning model of clinical events in the intensive care unit. PLoS ONE 2019, 14, e0211057.

[31] Shickel, B.; Loftus, T.J.; Adhikari, L.; Ozrazgat-Baslanti, T.; Bihorac, A.; Rashidi, P. DeepSOFA: A Continuous Acuity Score for Critically Ill Patients using Clinically Interpretable Deep Learning. Sci. Rep. 2019, 9, 1–12.

[32] Ruey-Kai Sheu and Mayuresh Sunil Pardeshi. 2022. A Survey on Medical Explainable AI (XAI): Recent Progress, Explainability Approach, Human Interaction and Scoring System, Sensors 2022, 22(20), 8068, 21 October 2022, https://doi.org/10.3390/s22208068.

[33] Rueckel, J.; Kunz, W.G.; Hoppe, B.F.; Patzig, M.; Notohamiprodjo, M.; Meinel, F.G.; Cyran, C.C.; Ingrisch, M.; Ricke, J.; Sabel, B.O. Artificial intelligence algorithm detecting lung infection in supine chest radiographs of critically ill patients with a diagnostic accuracy similar to board-certified radiologists. Crit. Care Med. 2020, 48, e574–e583.

[34] Lee, H.-C.; Yoon, S.B.; Yang, S.-M.; Kim,W.H.; Ryu, H.-G.; Jung, C.-W.; Suh, K.-S.; Lee, K.H. Prediction of Acute Kidney Injury after Liver Transplantation: Machine Learning Approaches vs. Logistic Regression Model. J. Clin. Med. 2018, 7, 428.

[35] Kang, Y.; Huang, S.T.;Wu, P.H. Detection of Drug–Drug and Drug–Disease Interactions Inducing Acute Kidney Injury Using Deep Rule Forests. SN Comput. Sci. 2021, 2, 1–14.

[36] Hua, Y.; Guo, J.; Zhao, H. Deep Belief Networks and deep learning. In Proceedings of the 2015 International Conference on Intelligent Computing and Internet of Things, Harbin, China, 17–18 January 2015; pp. 1–4.

[37] Letham, B.; Rudin, C.; McCormick, T.H.; Madigan, D. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. Ann. Appl. Stat. 2015, 9, 1350–1371.

[38] Che, Z.; Purushotham, S.; Khemani, R.; Liu, Y. Interpretable Deep Models for ICU Outcome Prediction. AMIA Annu. Symp. Proc. 2017, 2016, 371–380.

[39] Davoodi, R.; Moradi, M.H. Mortality prediction in intensive care units (ICUs) using a deep rule-based fuzzy classifier. J. Biomed. Inform. 2018, 79, 48–59.

[40] Johnson, M.; Albizri, A.; Harfouche, A. Responsible artificial intelligence in healthcare: Predicting and preventing insurance claim denials for economic and social wellbeing. Inf. Syst. Front. 2021, 1–17.

[41] Xu, Z.; Tang, Y.; Huang, Q.; Fu, S.; Li, X.; Lin, B.; Xu, A.; Chen, J. Systematic review and subgroup analysis of the incidence of acute kidney injury (AKI) in patients with COVID-19. BMC Nephrol. 2021, 22, 52.

[42] Angiulli, F.; Fassetti, F.; Nisticò, S. Local Interpretable Classifier Explanations with Self-generated Semantic Features. In Proceedings of the International Conference on Discovery Science, Halifax, NS, Canada, 11–13 October 2021; Springer: Cham, Switzerland, 2021; pp. 401–410.

[43] Visani, G.; Bagli, E.; Chesani, F. OptiLIME: Optimized LIME explanations for diagnostic computer algorithms. arXiv 2020, arXiv:2006.05714.

[44] Carrington, A.M.; Fieguth, P.W.; Qazi, H.; Holzinger, A.; Chen, H.H.; Mayr, F.; Manuel, D.G. A new concordant partial AUC and partial c statistic for imbalanced data in the evaluation of machine learning algorithms. BMC Med. Inform. Decis. Mak. 2020, 20, 1–12.

[45] Aziz Sheikh, Tony Cornford. Implementation and adoption of nationwide electronic health records in secondary care in England: final qualitative results from prospective national evaluation in “early adopter” hospitals. BMJ. 17 October 2011. 10.1136/bmj.d6054.

[46] Erik Štrumbelj and Igor Kononenko. “Explaining prediction models and individual predictions with feature contributions”. In: Knowledge and information systems 41.3 (2014), pp. 647–665.

[47] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning Important Features Through Propagating Activation Differences”. In: arXiv preprint arXiv:1704.02685 (2017).

[48] Anupam Datta, Shayak Sen, and Yair Zick. “Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems”. In: Security and Privacy (SP), 2016 IEEE Symposium on. IEEE. 2016, pp. 598–617.

[49] Avanti Shrikumar et al. “Not Just a Black Box: Learning Important Features Through Propagating Activation Differences”. In: arXiv preprint arXiv:1605.01713 (2016).

[50] Samy Bengio Łukasz Kaiser. Can active memory replace attention? In Advances in Neural Information Processing Systems, (NIPS), 2016.

[51] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. arXiv preprint arXiv:1610.10099v2, 2017.

[52] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122v2, 2017.

[53] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[54] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733, 2016.

[55] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model. In Empirical Methods in Natural Language Processing, 2016.

[56] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304, 2017.

[57] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130, 2017.

[58] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2440–2448. Curran Associates, Inc., 2015.

[59] Łukasz Kaiser and Ilya Sutskever. Neural GPUs learn algorithms. In International Conference on Learning Representations (ICLR), 2016.

[60] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. arXiv preprint arXiv:1610.10099v2, 2017.

[61] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs/1406.1078, 2014.

[62] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.

[63] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014.

[64] Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[66] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[67] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

[68] Müller, Heimo & Mayrhofer, Michaela & Veen, Evert-Ben & Holzinger, Andreas. (2021). The Ten Commandments of Ethical Medical AI. Computer. 54. 119-123. 10.1109/MC.2021.3074263.

[69] Enrico Costanza, Joel E. Fischer, James A. Colley, Tom Rodden, Sarvapali D. Ramchurn, and Nicholas R. Jennings. 2014. Doing the laundry with agents: A field trial of a future smart energy system in the home. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 813–822.

[70] Matthew Kay, Tara Kola, Jessica R. Hullman, and Sean A. Munson. 2016. When (ish) is my bus?: User-centered visualizations of uncertainty in everyday, mobile predictive systems. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM, 5092–5103.

[71] Frank C. Keil. 2006. Explanation and understanding. Annual Review of Psychology 57 (2006), 227–254.

[72] Tania Lombrozo. 2006. The structure and function of explanations. Trends in Cognitive Sciences 10, 10 (2006), 464–470.

[73] Tania Lombrozo. 2009. Explanation and categorization: How "why?" informs "what?". Cognition 110, 2 (2009), 248– 253.

[74] Jonathan Dodge, Sean Penney, Andrew Anderson, and Margaret M. Burnett. 2018. What should be in an XAI explanation? What IFT reveals. In IUI Workshops.

[75] Sean Penney, Jonathan Dodge, Claudia Hilderbrand, Andrew Anderson, Logan Simpson, and Margaret Burnett. 2018. Toward foraging for understanding of StarCraft agents: An empirical study. In 23rd International Conference on Intelligent User Interfaces (IUI'18). ACM, New York, NY, 225–237. https://doi.org/10.1145/3172944.3172946.

[76] Emilee Rader and Rebecca Gray. 2015. Understanding user beliefs about algorithmic curation in the Facebook news feed. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, 173–182.

[77] Brian Y. Lim and Anind K. Dey. 2009. Assessing demand for intelligibility in context-aware applications. In Proceedings of the 11th International Conference on Ubiquitous Computing. ACM, 195–204.

[78] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S. Lasecki, Daniel S. Weld, and Eric Horvitz. 2019. Beyond accuracy: The role of mental models in Human-AI team performance. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, Vol. 7. 2–11.

[79] Todd Kulesza, Simone Stumpf, Margaret Burnett, Weng-Keen Wong, Yann Riche, Travis Moore, Ian Oberst, Amber Shinsel, and Kevin McIntosh. 2010. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In ,2010 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'10). IEEE, 41–48.

[80] Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just, right? Ways explanations impact end users' mental models. In 2013 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'13). IEEE, 3–10.

[81] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. "It's reducing a human being to a percentage": Perceptions of justice in algorithmic decisions. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 377.

[82] Simone Stumpf, Simonas Skrebe, Graeme Aymer, and Julie Hobson. 2018. Explaining smart heating systems to discourage fiddling with optimized behavior.

[83] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In AAAI Conference on Artificial Intelligence.

[84] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In International Conference on Machine Learning. 2673–2682.

[85] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1675–1684.

[86] Besmira Nushi, Ece Kamar, and Eric Horvitz. 2018. Towards accountable AI: Hybrid human-machine analyses for characterizing system failure. In 6th AAAI Conference on Human Computation and Crowdsourcing.

[87] Mustafa Bilgic and Raymond J. Mooney. 2005. Explaining recommendations: Satisfaction vs. promotion. In Beyond Personalization Workshop, IUI, Vol. 5. 153.

[88] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. "It's reducing a human being to a percentage": Perceptions of justice in algorithmic decisions. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 377.

[89] Daniel Holliday, Stephanie Wilson, and Simone Stumpf. 2016. User trust in intelligent systems: A journey over time. In Proceedings of the 21st International Conference on Intelligent User Interfaces. ACM, 164–168.

[90] Kristina Höök. 2000. Steps to take before intelligent user interfaces become real. Interacting with Computers 12, 4 (2000), 409–426.

[91] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2119–2128.

[92] Andrea Bunt, Matthew Lount, and Catherine Lauzon. 2012. Are explanations always important? A study of deployed, low-cost intelligent interactive systems. In Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces. ACM, 169–178

[93] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Samuel J. Gershman, and Finale Doshi-Velez. 2019. Human evaluation of models built for interpretability. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, Vol. 7. 59–67.

[94] Sven Coppers, Jan Van den Bergh, Kris Luyten, Karin Coninx, Iulianna Van der Lek-Ciudin, Tom Vanallemeersch, and Vincent Vandeghinste. 2018. Intellingo: An intelligible translation environment. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 524.

[95] William Curran, Travis Moore, Todd Kulesza, Weng-Keen Wong, Sinisa Todorovic, Simone Stumpf, Rachel White, and Margaret Burnett. 2012. Towards recognizing cool: Can end users help computer vision recognize subjective attributes of objects in images. In Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces. ACM, 285–288.

[96] Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng Polo Chau. 2018. ActiVis: Visual exploration of industry-scale deep neural network models. IEEE Transactions on Visualization and Computer Graphics 24, 1 (2018), 88–97.

[97] Josua Krause, Adam Perer, and Enrico Bertini. 2014. INFUSE: Interactive feature selection for predictive modeling of high dimensional data. IEEE Transactions on Visualization and Computer Graphics 20, 12 (2014), 1614–1623.

[98] Mengchen Liu, Shixia Liu, Xizhou Zhu, Qinying Liao, Furu Wei, and Shimei Pan. 2016. An uncertainty-aware approach for exploratory microblog retrieval. IEEE Transactions on Visualization and Computer Graphics 22, 1 (2016), 250–259.

[99] Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. 2017. Towards better analysis of deep convolutional neural networks. IEEE Transactions on Visualization and Computer Graphics 23, 1 (2017), 91–100.

[100] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M. Rush. 2018. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. IEEE Transactions on Visualization and Computer Graphics 24, 1 (2018), 667–676.

[101] Robert R. Hoffman, Matthew Johnson, Jeffrey M. Bradshaw, and Al Underbrink. 2013. Trust in automation. IEEE Intelligent Systems 28, 1 (2013), 84–88.

[102] Maria Madsen and Shirley Gregor. 2000. Measuring human-computer trust. In 11th Australasian Conference on Information Systems, Vol. 53. Citeseer, 6–8.

[103] Debra Meyerson, Karl E. Weick, and Roderick M. Kramer. 1996. Swift trust and temporary groups. Trust in Organizations: Frontiers of Theory and Research 166 (1996), 195.

[104] Stephanie M. Merritt, Heather Heimbaugh, Jennifer LaChapell, and Deborah Lee. 2013. I trust it, but I don't know why: Effects of implicit attitudes toward automation on trust in an automated system. Human Factors 55, 3 (2013), 520–534.

[105] Philip Bobko, Alex J. Barelka, and Leanne M. Hirshfield. 2014. The construct of state-level suspicion: A model and research agenda for automated and information technology (IT) contexts. Human Factors 56, 3 (2014), 489–508.

[106] Robert R. Hoffman, John K. Hawley, and Jeffrey M. Bradshaw. 2014. Myths of automation, part 2: Some very human consequences. IEEE Intelligent Systems 29, 2 (2014), 82–85.

[107] Jiun-Yin Jian, Ann M. Bisantz, and Colin G. Drury. 2000. Foundations for an empirically determined scale of trust in automated systems. International Journal of Cognitive Ergonomics 4, 1 (2000), 53–71.

[108] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Understanding the effect of accuracy on trust in machine learning models. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. 1–12.

[109] Mahsan Nourani, Samia Kabir, Sina Mohseni, and Eric D. Ragan. 2019. The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, Vol. 7. 97–105.

[110] Béatrice Cahour and Jean-François Forzy. 2009. Does projection into use improve trust and exploration? An example with a cruise control system. Safety Science 47, 9 (2009), 1260–1270.

[111] Alyssa Glass, Deborah L. McGuinness, and Michael Wolverton. 2008. Toward establishing trust in adaptive agents. In Proceedings of the 13th International Conference on Intelligent User Interfaces. ACM, 227–236.

[112] Stavros Antifakos, Nicky Kern, Bernt Schiele, and Adrian Schwaninger. 2005. Towards improving trust in context aware systems by displaying system confidence. In Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services. ACM, 9–14.

[113] Pearl Pu and Li Chen. 2006. Trust building with explanation interfaces. In Proceedings of the 11th International Conference on Intelligent User Interfaces. ACM, 93–100.

[114] Florian Nothdurft, Felix Richter, and Wolfgang Minker. 2014. Probabilistic human-computer trust handling. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'14). 51–59.

[115] Shlomo Berkovsky, Ronnie Taib, and Dan Conway. 2017. How to recommend? User trust factors in movie recommender systems. In Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI'17). ACM, New York, NY, 287–300. https://doi.org/10.1145/3025171.3025209.

[116] Ellen J. Langer, Arthur Blank, and Benzion Chanowitz. 1978. The mindlessness of ostensibly thoughtful action: The role of “placebic” information in interpersonal interaction. Journal of Personality and Social Psychology 36, 6 (1978), 635.

[117] Adrian Bussone, Simone Stumpf, and Dympna O’Sullivan. 2015. The role of explanations on trust and reliance in clinical decision support systems. In International Conference on Healthcare Informatics (ICHI’15). IEEE, 160–169.

[118] Malin Eiband, Daniel Buschek, Alexander Kremer, and Heinrich Hussmann. 2019. The impact of placebic explanations on trust in intelligent systems. In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, LBW0243.

[119] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. 2012. Tell me more?: The effects of mental model soundness on personalizing an intelligent agent. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI’12). ACM, NewYork, NY, 1–10.

[120] Alex Groce, Todd Kulesza, Chaoqiang Zhang, Shalini Shamasunder, Margaret Burnett, Weng-Keen Wong, Simone Stumpf, Shubhomoy Das, Amber Shinsel, Forrest Bice, et al. 2014. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. IEEE Transactions on Software Engineering 40, 3 (2014), 307–323.

[121] Josua Krause, Aritra Dasgupta, Jordan Swartz, Yindalon Aphinyanaphongs, and Enrico Bertini. 2017. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In 2017 IEEE Conference on Visual Analytics Science and Technology (VAST’17). IEEE, 162–172.

[122] Simone Stumpf, Vidya Rajaram, Lida Li,Weng-KeenWong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. International Journal of Human-Computer Studies 67, 8 (2009), 639–662.

[123] 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. ACM, 126–137.

[124] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. Interactive topic modeling. Machine Learning 95, 3 (2014), 423–469.

[125] James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc Pottier, Anne Schur, and Vern Crow. 1995. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In Proceedings of Information Visualization, 1995. IEEE, 51–58.

[126] Nicholas Bryan and Gautham Mysore. 2013. An efficient posterior regularized latent variable model for interactive sound source separation. In International Conference on Machine Learning. 208–216.

[127] Jaegul Choo, Hanseung Lee, Jaeyeon Kihm, and Haesun Park. 2010. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In 2010 IEEE Symposium on Visual Analytics Science and Technology (VAST’10). IEEE, 27–34.

[128] Shixia Liu, Xiting Wang, Jianfei Chen, Jim Zhu, and Baining Guo. 2014. TopicPanorama: A full picture of relevant topics. In 2014 IEEE Conference on Visual Analytics Science and Technology (VAST’14). IEEE, 183–192.

[129] Daniel M. Best, Alex Endert, and Daniel Kidwell. 2014. 7 key challenges for visualization in cyber network defense. In Proceedings of the 11th Workshop on Visualization for Cyber Security. ACM, 33–40.

[130] Stephen Rudolph, Anya Savikhin, and David S. Ebert. 2009. FinVis: Applied visual analytics for personal financial planning. In IEEE Symposium on Visual Analytics Science and Technology, 2009. Citeseer, 195–202.

[131] John Goodall, Eric D. Ragan, Chad A. Steed, JoelW. Reed, G. David Richardson, Kelly M. T. Huffer, Robert A. Bridges, and Jason A. Laska. 2018. Situ: Identifying and explaining suspicious behavior in networks. IEEE Transactions on Visualization and Computer Graphics 25, 1 (2018), 204–214.

[132] Nicola Pezzotti, Thomas Höllt, Jan Van Gemert, Boudewijn P. F. Lelieveldt, Elmar Eisemann, and Anna Vilanova. 2018. DeepEyes: Progressive visual analytics for designing deep neural networks. IEEE Transactions on Visualization and Computer Graphics 24, 1 (2018), 98–108.

[133] Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. 2017. Understanding hidden memories of recurrent neural networks. In 2017 IEEE Conference on Visual Analytics Science and Technology (VAST’17). IEEE, 13–24.

[134] Qianwen Wang, Jun Yuan, Shuxin Chen, Hang Su, Huamin Qu, and Shixia Liu. 2019. Visual genealogy of deep neural networks. IEEE Transactions on Visualization and Computer Graphics 26, 11 (2020), 3340–3352.

[135] Wen Zhong, Cong Xie, Yuan Zhong, Yang Wang, Wei Xu, Shenghui Cheng, and Klaus Mueller. 2017. Evolutionary visual analysis of deep neural networks. In ICML Workshop on Visualization for Deep Learning.

[136] Mengchen Liu, Jiaxin Shi, Kelei Cao, Jun Zhu, and Shixia Liu. 2018. Analyzing the training processes of deep generative models. IEEE Transactions on Visualization and Computer Graphics 24, 1 (2018), 77–87.

[137] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng Polo Chau. 2019. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. IEEE Transactions on Visualization and Computer Graphics 26, 1 (2019), 1096–1106.

[138] Yongsu Ahn and Yu-Ru Lin. 2019. FairSight: Visual analytics for fairness in decision making. IEEE Transactions on Visualization and Computer Graphics 26, 1 (2019), 1086–1095.

[139] Brad A. Myers, David A.Weitzman, Andrew J. Ko, and Duen H. Chau. 2006. Answering why and why not questions in user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 397–406.

[140] Bernease Herman. 2017. The promise and peril of human evaluation for model interpretability. arXiv:1711.07414. https://arxiv.org/abs/1711.07414.

[141] Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. In Human and Machine Learning. Springer, 159–175.

[142] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. 2018. The building blocks of interpretability. Distill (2018). https://doi.org/10.23915/distill.00010 https://distill. pub/2018/building-blocks.

[143] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. In ICML Deep Learning Workshop 2015.

[144] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. 2016. Graying the black box: Understanding DQNs. In International Conference on Machine Learning. 1899–1908.

[145] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and MaxWelling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. arXiv:1702.04595. http://arxiv.org/abs/1702.04595.

[146] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In European Conference on Computer Vision. Springer, 818–833.

[147] Lingyang Chu, Xia Hu, Juhua Hu, Lanjun Wang, and Jian Pei. 2018. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1244–1253.

[148] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems. 4765–4774.

[149] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17). 2662–2670. https://doi.org/10.24963/ijcai.2017/371.

[150] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv:1312.6034. https://arxiv.org/abs/1312.6034.

[151] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS One 10, 7 (2015), e0130140.

[152] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un) reliability of saliency methods. In Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer, 267–280.

[153] Philipp Schmidt and Felix Biessmann. 2019. Quantifying interpretability and trust in machine learning systems. arXiv:1901.08558. https://arxiv.org/abs/1901.08558.

[154] Sina Mohseni and Eric D. Ragan. 2018. A human-grounded evaluation benchmark for local explanations of machine learning. arXiv:1801.05075. https://arxiv.org/abs/1801.05075.

[155] Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphological prediction. In 56th Annual Meeting of the Association for Computational Linguistics (ACL'18).

[156] Quanshi Zhang, Wenguan Wang, and Song-Chun Zhu. 2018. Examining CNN representations with respect to dataset bias. In 32nd AAAI Conference on Artificial Intelligence.

[157] Abhishek Das, Harsh Agrawal, Larry Zitnick,Devi Parikh, and Dhruv Batra. 2017. Human attention in visual question answering: Do humans and deep networks look at the same regions? Computer Vision and Image Understanding 163 (2017), 90–100.

[158] Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim. 2019. Towards automatic concept-based explanations. In Advances in Neural Information Processing Systems. 9273–9282.

[159] Andrew Slavin Ross and Finale Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In 32nd AAAI Conference on Artificial Intelligence.

[160] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems 28, 11 (2017), 2660–2673.

[161] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In Advances in Neural Information Processing Systems. 9505–9515.

[162] Robert R. Hoffman and Gary Klein. 2017. Explaining explanation, part 1: Theoretical foundations. IEEE Intelligent Systems 32, 3 (2017), 68–73.

[163] Shane T. Mueller and Gary Klein. 2011. Improving users' mental models of intelligent software tools. IEEE Intelligent Systems 26, 2 (2011), 77–83.

[164] Gomboc, D.; Solomon, S.; Core, M. G.; Lane, H. C.; and van Lent, M. 2005. Design recommendations to support automated explanation and tutoring. In Proc. of the Fourteenth Conference on Behavior Representation in Modeling and Simulation.

[165] Swartout, W. R., and Moore, J. D. 1993. Explanation in second generation expert systems. In David, J.; Krivine, J. P.; and Simmons, R., eds., Second Generation Expert Systems. Springer-Verlag.

[166] Laird, J. E.; Newell, A.; and Rosenbloom, P. 1987. Soar: An architecture for general intelligence. Artificial Intelligence 33:1–64.

[167] Traum, D.; Swartout, W.; Marsella, S.; and Gratch, J. 2005. Fight, flight or negotiate: Believable strategies for conversing under crisis. In Proc. of the 5th International Working Conference on Intelligent Virtual Agents.

[168] Sina Mohseni, Jeremy E. Block, and Eric D. Ragan. 2021. Quantitative Evaluation of Machine Learning Explanations: A Human-Grounded Benchmark. In IUI '21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021, Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O'Donovan, and Paul Teale (Eds.). ACM, 22–31. https: //doi.org/10.1145/3397481.3450689.

[169] Agathe Balayn, Natasa Rikalo, Christoph Lofi, Jie Yang, and Alessandro Bozzon. 2022. How can Explainability Methods be Used to Support Bug Identification in Computer Vision Models?. In CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022, Simone D. J. Barbosa, Cliff Lampe, Caroline Appert, David A. Shamma, Steven Mark Drucker, Julie R. Williamson, and Koji Yatani (Eds.). ACM, 184:1–184:16. https: //doi.org/10.1145/3491102.3517474.

[170] Vaishak Belle and Ioannis Papantonis. 2021. Principles and Practice of Explainable Machine Learning. Frontiers Big Data 4 (2021), 688969. https: //doi.org/10.3389/fdata.2021.688969.

[171] Umang Bhatt, Alice Xiang, Shubham Sharma, AdrianWeller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. 2020. Explainable machine learning in deployment. In FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, Mireille Hildebrandt, Carlos Castillo, L. Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna (Eds.). ACM, 648–657. https://doi.org/10.1145/ 3351095.3375624.

[172] Sérgio M. 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 FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021, Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 805–815. https://doi.org/10.1145/3442188.3445941.

[173] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. 2022. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. CoRR abs/2202.01602 (2022). arXiv:2202.01602 https://arxiv.org/abs/2202.01602.

[174] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behaviour?. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. 5540–5552. https://doi.org/10.18653/v1/2020.acl-main.491.

[175] Shipi Dhanorkar, Christine T. Wolf, Kun Qian, Anbang Xu, Lucian Popa, and Yunyao Li. 2021. Who needs to know what, when? Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle. In DIS '21: Designing Interactive Systems Conference 2021, Virtual Event, USA, 28 June, July 2, 2021, Wendy Ju, Lora Oehlberg, Sean Follmer, Sarah E. Fox, and Stacey Kuznetsov (Eds.). ACM, 1591–1602. https://doi.org/10.1145/3461778.3462131.

[176] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2019. An Evaluation of the Human-Interpretability of Explanation. CoRR abs/1902.00006 (2019). arXiv:1902.00006 http://arxiv.org/abs/1902.00006.

[177] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen K. Paritosh, and Lora Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021, Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker (Eds.). ACM, 39:1–39:15. https://doi.org/10.1145/3411764.3445518.

[178] Sebastian Bordt, Michèle Finck, Eric Raidl, and Ulrike von Luxburg. 2022. Post- Hoc Explanations Fail to Achieve their Purpose in Adversarial Contexts. In FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022. ACM, 891–905. https://doi.org/10. 1145/3531146.3533153.

[179] Amirata Ghorbani, Abubakar Abid, and James Y. Zou. 2019. Interpretation of Neural Networks Is Fragile. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press, 3681–3688. https://doi.org/10.1609/aaai.v33i01. 33013681.

[180] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020, Annette N. Markham, Julia Powles, Toby Walsh, and Anne L.Washington (Eds.). ACM, 180–186. https://doi.org/10.1145/3375627. 3375830.

[181] Eunjin Lee, David Braines, Mitchell Stiffler, Adam Hudler, and Daniel Harborne. 2019. Developing the sensitivity of LIME for better machine learning explanation. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Tien Pham (Ed.), Vol. 11006. International Society for Optics and Photonics, SPIE, 1100610. https://doi.org/10.1117/12.2520149.

[182] Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. 2022. Post hoc Explanations may be Ineffective for Detecting Unknown Spurious Correlation. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net. https://openreview.net/forum? id=xNOVfCCvDpM.

[183] XAI Explainable Artificial Intelligence IEEE Computer Society (IEEE C/AISC/XAI) Artificial Intelligence Standards Committee. 2020. IEEE P2894 - Guide for an Architectural Framework for Explainable Artificial Intelligence. https://standards.ieee.org/ieee/2894/10284/.

[184] Standard for XAI eXplainable AI Working Group IEEE Computational Intelligence Society/ Standards Committee (IEEE CIS/SC/XAI WG). 2024. IEEE CIS/SC/XAI WG P2976 - Standard for XAI – eXplainable Artificial Intelligence - for Achieving Clarity and Interoperability of AI Systems Design. https://standards.ieee.org/ieee/2976/10522/.

[185] Andrew Bell, Ian Solano-Kamaiko, Oded Nov, and Julia Stoyanovich. 2022. It's Just Not That Simple: An Empirical Study of the Accuracy-Explainability Tradeoff in Machine Learning for Public Policy. In FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022. ACM, 248–266. https://doi.org/10.1145/3531146.3533090.

[186] Tobias Budig, Selina Herrmann, and Alexander Dietz. 2020. Trade-offs between privacy-preserving and explainable machine learning in healthcare. In Seminar Paper, Inst. Appl. Informat. Formal Description Methods (AIFB), KIT Dept. Econom. Manage., Karlsruhe, Germany. https://doi.org/10.5445/IR/1000138902.

[187] Agathe Balayn and Seda Gürses. 2021. Beyond Debiasing: Regulating AI and its inequalities. Technical Report. https://edri.org/wp-content/uploads/2021/09/ EDRi_Beyond-Debiasing-Report_Online.pdf.

[188] Luciano Floridi. 2019. Translating Principles Into Practices of Digital Ethics: Five Risks of Being Unethical. Philosophy and Technology 32, 2 (2019), 185–193. https://doi.org/10.1007/s13347-019-00354-x.

[189] European Parliament and Council. 2020-12-15. Proposal for a regulation of the European Parliament and of the Council on a Single Market For Digital Services (Digital Services Act) and amending Directive 2000/31/EC.

[190] Michael A. Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna M. Wallach. 2020. Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI. In CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1– 14. https://doi.org/10.1145/3313831.3376445.

[191] Bogdana Rakova, Jingying Yang, Henriette Cramer, and Rumman Chowdhury. 2021. Where Responsible AI meets Reality: Practitioner Perspectives on Enablers for Shifting Organizational Practices. Proc. ACM Hum. Comput. Interact. 5, CSCW1 (2021), 7:1–7:23. https://doi.org/10.1145/3449081.

[192] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2019. LShapley and C-Shapley: Efficient Model Interpretation for Structured Data. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net. https://openreview.net/forum?id= S1E3Ko09F7.

[193] Hans de Bruijn, Martijn Warnier, and Marijn Janssen. 2022. The perils and pitfalls of explainable AI: Strategies for explaining algorithmic decision-making. Government Information Quarterly 39, 2 (2022), 101666. https://doi.org/10.1016/ j.giq.2021.101666.

[194] Agathe Balayn and Seda Gürses. 2021. Beyond Debiasing: Regulating AI and its inequalities. Technical Report. https://edri.org/wp-content/uploads/2021/09/ EDRi_Beyond-Debiasing-Report_Online.pdf.

[195] Ben Wagner. 2018. Ethics As An Escape From Regulation. From "Ethics-Washing" To Ethics-Shopping? Amsterdam University Press, Amsterdam, 84–89. https: //doi.org/10.1515/9789048550180-016.

[196] Elettra Bietti. 2020. From ethics washing to ethics bashing: a view on tech ethics from within moral philosophy. In FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, Mireille Hildebrandt, Carlos Castillo, L. Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna (Eds.). ACM, 210–219. https://doi.org/10.1145/3351095.3372860.

[197] Ben Green. 2021. The Contestation of Tech Ethics: A Sociotechnical Approach to Technology Ethics in Practice. J. Soc. Comput. 2, 3 (2021), 209–225. https: //doi.org/10.23919/JSC.2021.0018.

[198] Merve Hickok. 2021. Lessons learned from AI ethics principles for future actions. AI and Ethics 1, 1 (2021), 41–47. https://doi.org/10.1007/s43681-020-00008-1.

[199] Federico Cabitza, Andrea Campagner, Gianclaudio Malgieri, Chiara Natali, David Schneeberger, Karl Stoeger, and Andreas Holzinger. 2023. Quod erat demonstrandum? - Towards a typology of the concept of explanation for the design of explainable AI. Expert Systems with Applications 213 (2023), 118888. https://doi.org/10.1016/j.eswa.2022.118888.

[200] Scott Robbins. 2019. A misdirected principle with a catch: explicability for AI. Minds and Machines 29, 4 (2019), 495–514. https://doi.org/10.1007/s11023-019- 09509-3.