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

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.

1 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 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 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

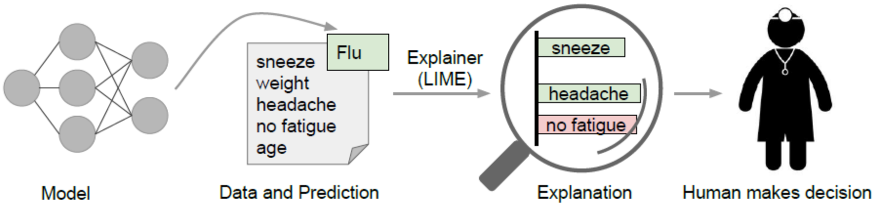


Figure 1: Elucidating the rationale behind individual predictions. According to the model's forecast, the patient exhibits the condition "u," and the LIME technique effectively identifies the specific symptoms within the patient's medical history that contributed to this prediction. The symptoms of sneezing and headache are depicted as factors that support the hypothesis; however, the absence of exhaustion serves as contradictory evidence. Based on these factors, a medical professional can make a well-informed determination regarding the reliability of the model's prognosis.[10]

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.

2 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. In this section, we will examine the preprocessing techniques employed in the recent study, the algorithms implemented in their respective models, and the resulting outcomes.

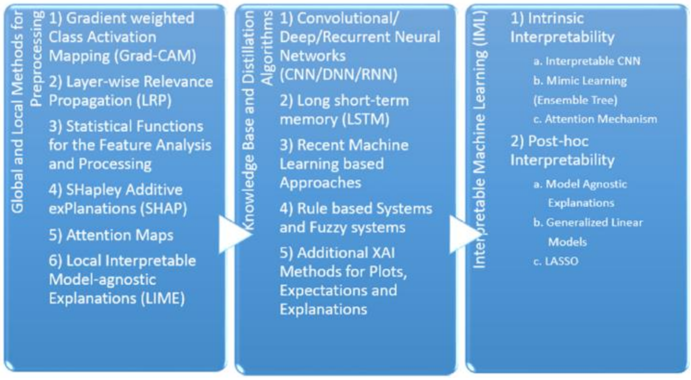

Global and Local Methods for Preprocessing
1) Gradient weighted Class Activation Mapping (Grad-CAM)
2) Layer-wise Relevance Propagation (LRP)
3) Statistical Functions for the Feature Analysis and Processing
4) SHapley Additive exPlanations (SHAP)
5) Attention Maps
6) Local Interpretable Model-agnostic Explanations (LIME)
Knowledge Base and Distillation Algorithms
1) Convolutional/ Deep/Recurrent Neural Networks (CNN/DNN/RNN)
2) Long short-term memory (LSTM)
3) Recent Machine Learning based Approaches
4) Rule based Systems and Fuzzy systems
5) Additional XAI Methods for Plots, Expectations and Explanations
Interpretable Machine Learning (IML)
1) Intrinsic Interpretability
a. Interpretable CNN
b. Mimic Learning (Ensemble Tree)
c. Attention Mechanism
2) Post-hoc Interpretability
a. Model Agnostic Explanations
b. Generalized Linear Models
c. LASSO

Figure 2: XAI Methods Categorization [32].

2.1 Gradient Weighted Class Activation Mapping (Grad-CAM)

The Grad-CAM technique [19] is employed to forecast the corresponding notion by leveraging the gradients of the target, which are propagated to the final convolutional 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 image 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 number of classes as provided in Equation 1. The Multi-layer Grad-CAM method is employed to calculate the conditional probability of the chosen feature 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.

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

2.2 Layer-Wise Relevance Propagation (LRP)

It is also one of the popularly used propagation methods, which operates by using the propagation rules for propagating the prediction backward in the neural network. The LRP can flexibly operate on input such as images, videos, and texts. The relevance scores can be recorded in each layer by applying different rules. The LRP is based and justified using a deep taylor decomposition (DTD). It can be set on a single or set of layers in the neural network and can be scaled in the complex DNN by providing high explanation quality. It is also popularly used in the medical domain consisting of CXR, axial brain slices, brain relevance maps, and abnormalities, etc. The versions available in LRP are LRP CNN, LRP DNN, LRP BiLRP, LRP DeepLight for the heatmap visualizations. The LRP relevance is higher as compared to other visualization/sensitivity analysis. The input representations are forward-propagated using CNN until the output is reached and back-propagated by the LRP until the input is reached. Thus, the relevance scores for the categories are yielded in LRP CNN [20]. For the LRP DNN [21], the CNN is tuned with initial weights for the activity recognition with pixel intensity. In LRP BiLRP [22], the input features pairs having similarity scores are systematically decomposed by this method. The high nonlinear functions are scaled and explained by using composition of LRP. Thus, the BiLRP provides a similarity model for the specific problem by verifiability and robustness. The BiLRP is presented as a multiple LRP combined procedure and recombined on input layer. Here, x and x' are input which are to be compared for similarity, $\emptyset_x$ as a group of network layer with $\{\emptyset_1\ to\ \emptyset_L\}$, and $y(x, x')$ as the combined output given in Equation (2). The DeepLight LRP [23] performs decoding decision decomposition, which is used to analyze the dependencies between multiple factors on multiple levels of granularity. It is used to study the fine-grained temporo-spatial variability of the high dimension and low sample size structures.

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

2.3 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.

2.4 Shapely Additive exPlanations (SHAP)

The SHAP [29] uses ranking based algorithms for feature selection. The best feature is listed in the descending values by using SHAP scores. It is based on the features attribution magnitude and is an additive feature attribution method. SHAP is a framework that uses shapley values to explain any model’s output. This idea is a part of game theoretic approach which is known for its usability in optimal credit allocation. SHAP can compute well on the black box models as well as tree ensemble models. It is efficient to calculate SHAP values on optimized model classes but can suffer in equivalent settings of model-agnostic settings. Individual aggregated local SHAP values can also be used for global explanations due to their additive property. For deeper ML analysis such as fairness, model monitoring, and cohort analysis, SHAP can provide a better foundation.

2.5 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.

$$a_k = softmax(W_k x_k) \qquad (3)[32]$$

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.

$$y_k = a_k \odot x_k \qquad (4)[32]$$

3 MODEL SELECTION

3.1 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.

3.2 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.

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

3.3 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 $\frac{\partial p(v,\theta)}{\partial \theta}$.

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

3.4 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.

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

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.

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

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

3.5 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.

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

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

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.

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

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.

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

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.

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

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

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

4 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.

4.1 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].

4.2 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.

4.3 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 $d_{model} = 512$.

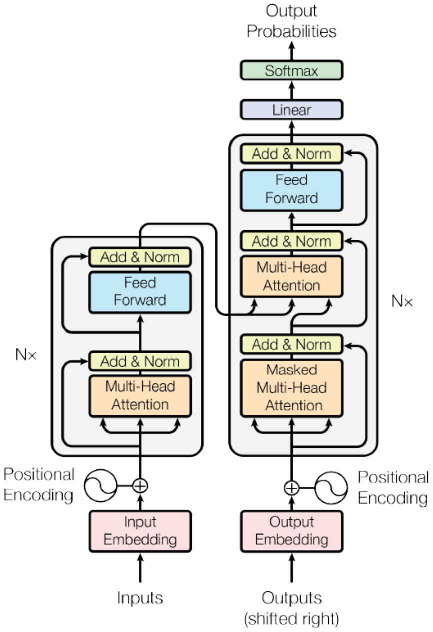


Figure 3: The Transformer – model architecture [1].

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.

4.4 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.

4.5 Scaled Dot-Product Attention

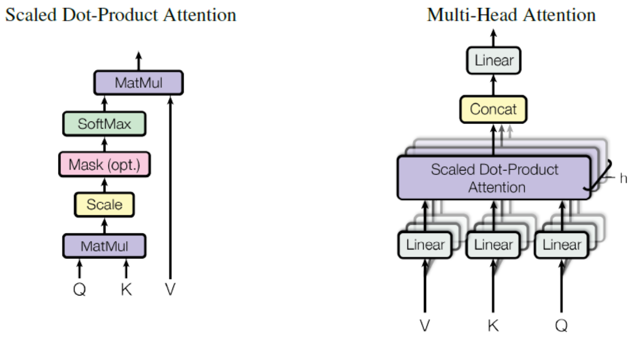


Figure 4: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel [1].

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:

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

4.6 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.

4.7 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.

4.8 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.

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

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.

4.9 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:

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

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.

5 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

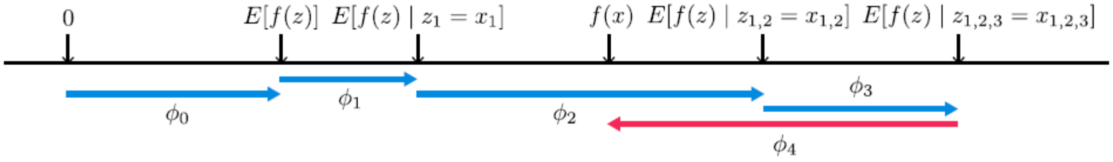


Figure 5: The SHAP (SHapley Additive exPlanation) values assign importance to each feature by measuring the change in the predicted model prediction when that feature is taken into account. The authors elucidate the process of transitioning from the initial predicted value E[f(z)], which would be obtained in the absence of any knowledge about the features, to the present output f(x). The diagram presented illustrates a solitary arrangement. In cases when the model exhibits non-linearity or the input features are not independent, it is important to consider the order in which features are included in the expectation. The SHAP values are derived by calculating the average of the individual values (denoted as ϕ_i) across all potential 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 $f\big(h_x(z')\big) = E|f(z)z_S|$, 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]

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

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

Property 2 (Missingness) [2]

$$x'_i = 0 \Longrightarrow \phi_i = 0 \qquad (23)[2]$$

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*

$$f'_x(z') - f'_x\big(z'\backslash i\big) \geq f_x(z') - f_x(z'\backslash i) \qquad (24)[2]$$

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 $\bar{S}$.

$$\begin{aligned} f\left(h_x(z')\right) &= E[f(z)|z_S] && \text{SHAP explanation model simplified input mapping} \quad (25)[2] \\ &= E_{z_S|z_S}[f(z)] && \text{expectation over } z_{\bar{S}}|z_S \quad (26)[2] \\ &= E_{z_S}[f(z)] && \text{assume feature independence ([46],[10],[47],[48])} \quad (27)[2] \\ &\approx f(|z_S, E[z_S]|) && \text{assume model linearity} \quad (28)[2] \end{aligned}$$

5.1 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.

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

5.1.1Kernel 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 $\pi_{x\prime}$ 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 $\pi_{x'}$, 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).

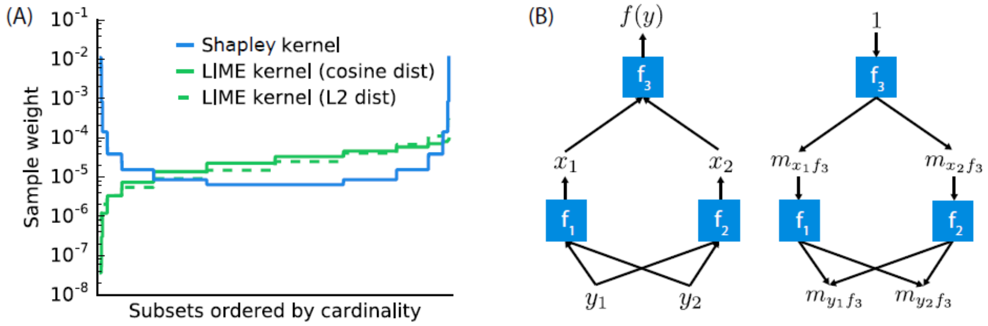


Figure 6: (A) The Shapley kernel weighting exhibits symmetry when the set of all potential z0 vectors is arranged in ascending order based on their cardinality. In the specific example under consideration, there are a total of 215 vectors. This exhibits a clear divergence from previously selected kernels based on heuristics. (B) Compositional models, such as deep neural networks, consist of numerous elementary components. By utilizing the analytic answers for the Shapley values of the components, it is possible to efficiently approximate the whole model using the back-propagation technique employed by DeepLIFT. [2]

5.2 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.

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

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

6 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.

6.1 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 $\mathrm{x} \in \mathrm{R}^{\mathrm{d}}$ be the original representation of an instance being explained, and we use $x' \in \{0, 1\}^{\mathrm{d}'}$ to denote a binary vector for its interpretable representation.

6.2 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 $\Omega(\mathrm{g})$ as a metric of complexity, rather than interpretability, for the explanation $g \in G$. As an illustration, in the case of decision trees, $\Omega(\mathrm{g})$ might represent the depth of the tree, whereas in the context of linear models, $\Omega(\mathrm{g})$ 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 $\pi_x(\mathrm{z})$ 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 $\Omega(\mathrm{g})$ 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.

6.3 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 ξ(x). The fundamental concept underlying LIME is illustrated in Figure 7, where we select examples that are

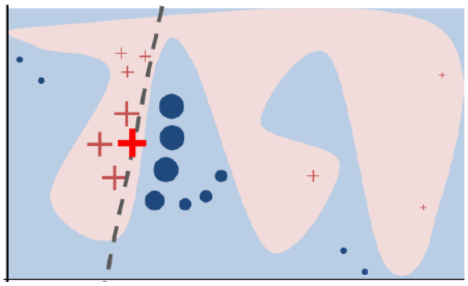


Figure 7: Illustrative example to demonstrate the conceptual understanding of LIME. The decision function f of the black-box model, which is not known to LIME, is of a complex nature. It is visually depicted by the blue/pink background, indicating that it cannot be accurately approximated by a linear model. The bold red cross represents the specific example that is being elucidated. LIME employs a sampling technique to choose instances, using a function f to generate predictions, and assigns weights based on the proximity of these

instances to the instance under explanation, as depicted by their respective sizes. The dashed line represents the acquired explanation that exhibits local fidelity, but not global fidelity.[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.

6.4 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.

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

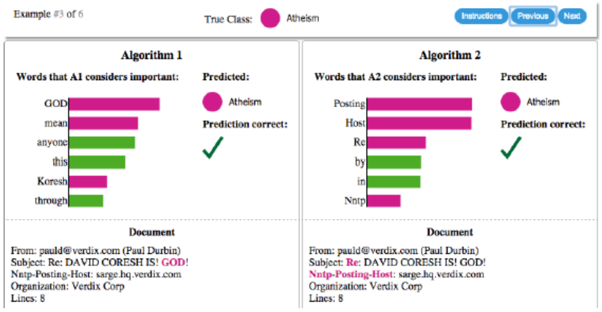
In the context of text classification, it is imperative to guarantee that the provided explanation is easily understandable. This is achieved by employing an interpretable representation known as a bag of words. Additionally, a constraint is imposed by placing a limit, denoted as K, on the number of words. Mathematically, this constraint can be expressed as $\Omega(g)\ =\ \infty 1[\left|w_g\right|_0\ >\ K]$. 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 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.

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 $\pi_x$, Length of explanation $K$
  $Z \leftarrow \{\}$
  for $i \epsilon \{1, 2, 3, \ldots, N$ 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 $w$
```

6.5 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



From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Figure 8: This image aims to elucidate the rationale behind individual predictions made by competing classifiers in the task of discerning the thematic focus of a given document, namely whether it pertains to "Christianity" or "Atheism". The bar chart depicts the significance

attributed to the most pertinent terms, which are also emphasized in the accompanying text. The color scheme employed in this context serves to denote the specific class to which a given term pertains. The color green is utilized to represent the class of words associated with "Christianity," whereas the color magenta is employed to signify the class of words associated with "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.

6.6 Example 2: Deep networks for images

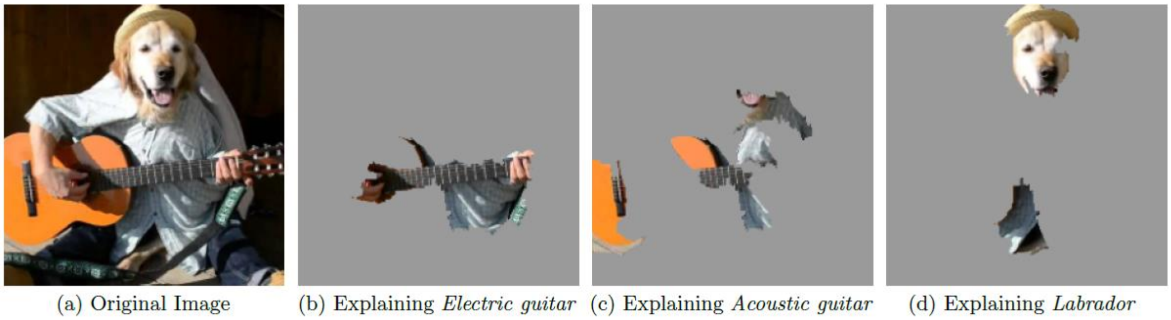

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

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.

7 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.

8 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.

8.1 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.

8.2 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.

8.3 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.

8.4 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.

8.5 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.

8.6 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.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Amber Wagner for her invaluable guidance, unwavering support, and mentorship throughout the course of this research.

Professor Wagner's dedication to excellence and her commitment to nurturing young researchers have been truly inspiring. Her insightful feedback, constructive criticism, and patient guidance have significantly contributed to the refinement and depth of this research paper.

Furthermore, I would like to extend my thanks to the entire research team for their collaborative efforts and valuable contributions, which have enriched the scope of this study.

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 recogniti on 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.