Received: 28 September 2019 | Revised: 8 June 2020 | Accepted: 1 August 2020

DOI: 10.1111/coin.12410

SPECIAL ISSUE ARTICLE

Computational Intelligence WILEY

# Interpretability in healthcare: A comparative study of local machine learning interpretability techniques

Radwa ElShawi[1] | Youssef Sherif[1] | Mouaz Al-Mallah[2] | Sherif Sakr[1]

[1]Tartu University, Tartu, Estonia

[2]Houston Methodist Center, Houston, Texas, USA

**Correspondence**
Sherif Sakr, Tartu University, Tartu, Estonia.
Email: sherif.sakr@ut.ee

**Funding information**
European Regional Development Fund, Grant/Award Numbers: MOBJD341, grant MOBTT75

## Abstract

Although complex machine learning models (eg, random forest, neural networks) are commonly outperforming the traditional and simple interpretable models (eg, linear regression, decision tree), in the healthcare domain, clinicians find it hard to understand and trust these complex models due to the lack of intuition and explanation of their predictions. With the new general data protection regulation (GDPR), the importance for plausibility and verifiability of the predictions made by machine learning models has become essential. Hence, interpretability techniques for machine learning models are an area focus of research. In general, the main aim of these interpretability techniques is to shed light and provide insights into the prediction process of the machine learning models and to be able to explain how the results from the prediction was generated. A major problem in this context is that both the quality of the interpretability techniques and trust of the machine learning model predictions are challenging to measure. In this article, we propose four fundamental quantitative measures for assessing the quality of interpretability techniques—`similarity`, `bias detection`, `execution time`, and `trust`. We present a comprehensive experimental evaluation of six recent and popular *local* model agnostic interpretability techniques, namely, LIME, SHAP, Anchors, LORE, ILIME“ and MAPLE on different types of real-world healthcare data. Building on previous work, our experimental evaluation covers different aspects for its comparison including identity, stability, separability, similarity, execution time, bias detection, and trust. The results of our experiments show that MAPLE achieves the highest performance for the identity across all data sets included in this study, while LIME achieves the lowest performance for the identity metric. LIME achieves the highest performance for the separability metric across all data sets. On average, SHAP has the smallest average time to output explanation across all data sets included in this study. For detecting the bias, SHAP and MAPLE enable the participants to better detect the bias. For the trust metric, Anchors achieves the highest performance on all data sets included in this work.

**KEYWORDS**

big data, data science, interpretability, machine learning

## 1 | INTRODUCTION

Machine learning models have been proven to be successful in many application domains such as financial systems, advertising, marketing, criminal justice system, and medicine. Despite the growing use of machine learning-based prediction models in the medical domains,[1] clinicians still find it hard to rely on these models. In practice, most of the models developed by data scientists mainly focus on prediction accuracy as a performance metric but rarely explain their prediction in a meaningful way.[2,3] Generally speaking, there is a trade-off between the performance of machine learning models and their interpretability. That is the more interpretable the model such as linear models and decision trees, the lower their performance would be compared with complex models such as deep learning models. Thus, there have been some criticism for using complex machine learning models in the medical domain even with their promise of high accuracy.

One way to define the interpretability of machine learning predictions is defined as the degree to which users can understand and comprehend the predictions made by the machine learning models.[4] In particular, the main aim of interpretability is to assist in understanding the most influential features that lead the model to a specific prediction which would significantly help the clinicians to understand the reasoning behind a specific prediction and hence they will be able to accept or reject the prediction. In addition, understating the prediction process is always useful for getting some insights of how this model is working and can help in improving the prediction process and model performance for future predictions.

Recently, interpretability has been receiving a notable attention especially after the new general data protection regulation (GDPR) imposed by the European Parliament in May 2018 that forces industries to "explain" any decision taken when automated decision making takes place: "a right of explanation for all individuals to obtain meaningful explanations of the logic involved."[5] One of the main challenges in the research on machine learning interpretability is that it is hard for the community to agree on a specific definition of interpretability.[6] Consequently, it is difficult to access the quality of interpretability techniques and hence various interpretability frameworks cannot be easily compared in unified benchmark tests. Quantitative metrics for assessing interpretability enable machine learning practitioners to choose among different interpretability techniques and also to choose the most trustworthy machine learning model among arsenal of possible machine learning models. Agreeing upon clear quantitative metrics for interpretability contributes toward significant improvement in developing new trusted interpretability techniques. Our work builds on existing line of research that focuses on quantifying measures for assessing interpretability techniques. The main contribution of this article is as follows:

- We propose four quantitative indicators, namely, `similarity`, `bias detection`, `execution time`, and `trust` for measuring the quality of the explanations of different interpretability frameworks.
- We present a detailed experimental evaluation of six recent and popular *local* model agnostic interpretability techniques, namely, `LIME`,[7] `Anchors`,[8] `SHAP`,[9] `LORE`,[10] `ILIME`,[11] and `MAPLE`[12] on different types (tabular and text data) of real-world healthcare data using seven different metrics, namely, `identity`, `stability`, `separability`, `similarity`, `execution time`, `bias detection` (used to evaluate tabular data only), and `trust`. For ensuring repeatability as one of the main targets of this work, we provide access to the source codes and the detailed results for the experiments of our study[1].

The remainder of this article is organized as follows: after recapitulating some of the related work in Section 2, we introduce quantitative metrics for evaluation and comparison of interpretability techniques in Section 3. Section 4 provides an overview of the different local interpretability techniques that have been considered in this study. Section 5 describes the details of our experimental setup in terms of used data sets and machine learning models. The detailed results of our experiments are presented in Section 6 before we conclude the article in Section 7.

## 2 | RELATED WORK

Research in the topic of machine learning interpretability can be broadly partitioned into two main categories: conceptual and technical. In the category of conceptual contribution, a vital aspect considered is the trade-off between interpretability and explaining the prediction of the machine learning model faithfully.[13] Explaining the prediction of machine learning model in a way that is understandable and comprehendible by humans is called `simulatability`.[6] Our work builds on these findings in that our proposed metrics are grounded on human intuition

[1] https://github.com/DataSystemsGroupUT/Interpretability-comparison

in evaluating the explanation of machine learning interpretability techniques. Some of the main common aspects that have been highlighted in the conceptual work are (i) a good machine learning explanation should be easy to understand and in line with human intuition and (ii) there is a lack of well-defined quantitative metrics for assessing the quality of machine learning interpretability frameworks.[14] Nevertheless, there are few studies focus on the evaluation of interpretability techniques.[15,16] Indicators used for assessing the quality of explanations are categorized into qualitative and quantitative indicators.[16] Doshi-Velez and Kim[14,17] proposed five qualitative indicators as follows.

- *Form* of cognitive chunks (basic unit of the explanation): refers to the form of explanation provided by the interpretability framework such as feature importance values, set of rules, or crops from the image to be explained that mainly contribute to the prediction.
- *Number* of cognitive chunks that the explanation contains. The fewer the number of cognitive chunks used, the more the explanation is understandable for humans.
- *Compositionality*: refers to the structure of the cognitive chunks such as rules, hierarchies, or other type of abstraction that affect the human ability to comprehend and understand the explanation. Other forms of compositionality includes the order of features attribution in an explanation or the threshold used to constrain the number of features included in an explanation.
- *Monotonicity and interaction between cognitive chunks*: refers to the type of interaction between cognitive chunks such as linear, nonlinear, and so on. Some relations between the cognitive chunks are more intuitive for humans and thus easier to understand than others.
- *Uncertainty and stochasticity*: refers to the uncertainty in the explanation provided that may be due to some randomization in the explanation process.

Honegger[18] proposed three axioms for evaluating model-agnostic interpretability frameworks that are based on human intuition. Such axioms relate the instances to be explained with their corresponding explanations. Wilson et al[19] proposed three proxy metrics for evaluating the quality of the explanations which are *Completeness*, *Correctness*, and *Compactness*. *Completeness* refers to the audience ability to verify the validity of the explanation, that is, the number of instances that are covered by an explanation. *Correctness* refers the accuracy of the explanation. *Compactness* refers to the degree of how an explanation can be succinct, for example, the number of conditions included in a decision rule that explains particular instance. Lundberg and Lee[20] presents an interesting quantitative measure for evaluating SHAP interpretability technique which is based on measuring the overlap between human intuitions and SHAP explanation. The main limitation of this quantitative measure is that it is hard to scale as it needs task specific user studies. Another quantitative measure proposed by Ribeiro et al[7] in which explanations of a machine learning model have been introduced for students of machine learning class and then they were asked to guess the model prediction based on the introduced explanation. The faster and more accurate the students answers, the better the interpretability technique would be.

The category of technical contribution can be further partitioned into two categories: *global* or *local*.[21] In principle, global techniques enable clinicians to understand the entire conditional distribution modeled by the trained response function and obtained based on average values. By contrast, local interpretations, which is the main focus of this work, promote the understanding of small parts of the conditional distribution for specific instances. Since conditional distribution decomposes of small parts that are more likely to be linear or well-behaved, they can be explained by interpretable models such as linear regression and decision trees.[7,11,12,22] The idea of using influence functions[23] is to measure the influence of particular data point or feature by perturbing its value and then measuring the influence of this change on the model prediction in different interpretability techniques.[9,20] Since deep neural networks (DNN) have been well performing in many application domains,[24] there has been great attention for designing various interpretability techniques for explaining the local predictions of DNN. Some key examples of such local techniques include guided backpropagation,[25] SmoothGrad saliency maps,[26] integrated based techniques,[27] Grad-CAM,[28] and testing with concept activation vectors (TCAV).[29]

## 3 | QUANTIFYING THE QUALITY OF MACHINE LEARNING EXPLANATIONS

Our work builds on existing three axioms proposed by Honegger[18] which relates an instance to its corresponding explanation. The three axioms are as follows.

1. *Identity*: This metric states that if there are two identical instances, then they must have identical explanations.
2. *Stability*: This metric states that instances belong to the same class must have comparable explanations.
3. *Separability*: This metric states that if there are two dissimilar instances, then they must have dissimilar explanations. This metric holds the assumption that the model does not have degree of freedom;[27] this means that all the features used in the model are relevant to the prediction.

We propose four quantitative measures for quantifying the quality of interpretability techniques that can also be used to guide the design of new techniques.

1. *Similarity*: This metric states that the more similar the instances to be explained, the closer their explanations should be and vice versa.
2. *Time*: This metric represents the average time taken by the interpretability framework to output an explanation. The time evaluated on a standard machine with Intel Core i7 6500U and 8 GB RAM.
3. *Bias detection*: This metric capture the interpretability framework tendency to detect bias in training data (used to train a machine learning model) from the explanations of instances in the testing data set.
4. *Trust*: This metric captures the mutual agreement between the explanations provided by the interpretability frameworks and the black-box model behavior.

In principle, measuring the `identity` is straightforward. In particular, for every two instances in the testing data set, if the distance between the two instances is equal to zero (identical), then the distance between their explanations should be equal to zero. To measure the `separability` metric, we choose a subset $S$ of the testing data set that has no duplicates and get their explanations. Then for every instance $s$ in $S$, we compare its explanation with all other explanations of instances in $S$ and if such explanation has no duplicate then it satisfies the separability metric. To measure the `similarity` metric, we cluster instances in the testing data set, after normalization using DBSCAN algorithm. For each framework, we normalize the explanations and calculate the mean pairwise Euclidean distances between explanations of testing instances in the same cluster. The framework with the smallest mean pairwise Euclidean distances across its clusters is the best reflecting the similarity metric. Measuring the `stability` metric is done by clustering the explanations of all instances in the testing data set using `K-means` clustering algorithm such that the number of clusters equals to the number of labels of the data set. For each instance in the testing data set, we compare the cluster label assigned to its explanation after clustering with the instance's predicted class label and if they match then this explanation satisfies the stability metric. To measure the `trust` metric, for each framework we do the following. First, for each instance $x$ to be explained in the testing data set, select a feature $j$ in $x$ and the number of iteration $M$. For each iteration, a random instance $x'$ is selected from the testing data set and a random permutation of features is chosen and applied on $x$ and $x'$. Two new instances are created by combining features' values of $x$ and $x'$. The first instance $z$ is created by taking the values of all features before and including $j$ from $x$ and the rest from $x'$. The second instance $z'$ is created by taking the values of all features before $j$ from $x$ and rest of the features from $x'$. Next, calculate the importance of feature $j$ by calculating the difference in the prediction of the black-box model between $z$ and $z'$ averaged over $M$. The procedure has to be repeated for each feature in $x$. Finally, retrieve the most important six features (gold set) and compute the fraction of these gold features that are recovered by the explanation. The trust value for an interpretability framework is this recall averaged over all instances in the testing data set.

For the similarity metric, it is desirable that the higher the similarity between two instances to be explained, the more closer should be their corresponding explanations, and vice versa. For the time metric, the faster the average time taken to output an explanation, the better the interpretability technique. This metric is related to computational complexity of the interpretability framework which is very important to consider regarding feasibility, especially when computation time is a bottleneck in generating explanations.[30] Sometimes a bias in the data used in training and testing a machine learning model is hard to detect through model performance alone. Hence, a human understanding of the underlying data through interpretability techniques is needed. For example, Caruana et al[31] proposed a machine learning model for predicting risk of readmission for patients with pneumonia. Counterintuitively, the trained machine learning model learned that patients with asthma are at lower risk of readmission due to certain bias in the data. So the bias detection metric measures the ability of users to detect systematic bias from machine learning explanations. We conducted a user study using aggregated instance-based explanations and their corresponding individual counterparts from different interpretability techniques. We provide users with two different models with biased and unbiased data for each interpretability framework. We measure the quality of the explanation technique by how fast and accurate the participants are able to detect the bias on the underlying data from the explanations of each interpretability framework. The trust metric captures the ability of the interpretability technique to return the main features that contributed to the prediction of the black-box model. More specifically, we measure how relevant the features returned by the interpretability technique are for the model prediction. Intuitively, the more relevant features included in the explanation of the interpretability technique, the more the technique is trusted.

## 4 | OVERVIEW ON INTERPRETABILITY FRAMEWORKS

In this section, we give an overview on the local interpretability techniques which we consider in this study.

## 4.1 | LIME

The LIME technique[7] has been introduced as a local interpretability technique that relies on the assumption that the decision boundary of a complex machine learning model is linear locally around the instance to be explained. It explains the instance of interest by fitting an interpretable model on perturbed sample around the input instance of interest. In particular, LIME generates a perturbed sample around the instance to be explained. For each instance in the perturbed sample, LIME gets the prediction from the model to be explained (the perturbed sample along with the prediction will act as the training data set for the interpretable model). Then, the technique assign weights to the instances in the new training data set according to their proximity to the instance to be explained. Finally, LIME fits an interpretable model on the new training data set.

## 4.2 | Anchors

Anchors is a rule-based model-agnostic local explainer technique.[8] In general, LIME explanations do not have a clear coverage; it is unclear whether the explanation given for a specific instance $x$ is applicable in the region where $x$ is located. Anchors guarantee that the predictions of instances in the same anchor is almost the same. In other words, Anchors find the features that are enough to fix the prediction such that changing the other features has no impact on the prediction. One way to construct anchors is the bottom-up approach[8] in which anchor is constructed incrementally. In particular, Anchors starts with an empty rule and in each iteration the rule is extended with one feature such that the new rule has the highest estimated precision. To select the best rule in each iteration, KL-LUCB algorithm is used.[32] The KL-LUCB algorithm works by constructing confidence regions based on the KL divergence.[33] Another way to construct anchors is the beam-search approach which maintains a set of candidate rules that guide the search to select the anchor with the highest coverage among many possible anchors. In practice, the greedy approach suffers from the following limitations: (i) can only maintain one rule at a time and hence any suboptimal choice cannot be changed and (ii) returns the shortest anchor which may not be the anchor with the highest coverage. Thus, another way to construct anchors is the beam-search approach which address the limitations of the greedy approach. The beam-search maintains a set of candidate rules and guide the search to select the anchor with the highest coverage among many possible anchors.

## 4.3 | SHAP

An idea from game theory was applied to measure the role of each feature on the prediction process. The Shapley value[34] is a method from coalitional game theory that can fairly distribute the gain among players (features), where contributions of players are unequal. In other words, Shapley values is based on the idea that features play a role together to change the model's prediction toward some value. It then tries to make a fair distribution of their contributions across all subsets of features. In particular, Shapley value fairly distributes the difference between the prediction and the average prediction among the feature values of the instance to be explained. Shapely value satisfies three interesting properties.[9] The first one is *local accuracy*, local accuracy requires the output of the explanation model for input $x'$ to match the output of the original model for input $x$, where $x'$ is the simplified input of $x$. The second property is *missingness*, missingness requires that features that not presented in the instance to be explained to have no impact on the explanation. The third property is *consistency*, consistency states that if we have two models $f$ and $f'$ such that the input contribution of a particular feature $i$ in an instance $x$ in model $f$ is greater than the contribution of the same feature in the same instance $x$ in model $f'$, then the impact of feature $i$ in the explanation of $x$ in model $f$ should be greater than the contribution of feature $i$ in $x$ in model $f'$.

LIME does not guarantee a fair distribution of the effect among the features, while Shapley value does. This might make Shapely value a robust technique in generating individual explanations. Moreover, the calculation is based on solid theory while methods like LIME assume linearity in the local model behavior which is still questionable. Additionally, the Shapley value makes it feasible to compare a prediction with the average prediction of the whole data set as well as comparing it with only a subset or a single data point. In practice, one of the main challenges in the Shapely value approach is the computation time. In particular, for exact computation of Shapley value, all possible sets (coalitions) of features need to be evaluated (with and without the feature of interest). The exact value calculation becomes hard to compute when the number of features is large as the number of sets increases exponentially with the number of features. To avoid this issue, sampling techniques were introduced to sample coalitions with fixed number of samples.[35] While other attempts proposed different computation method for Shapely value that comprises weight kernels and regularized linear regression[20] which will be evaluated in this work (SHAP).

## 4.4 | LORE

LOcal Rule-based Explanations (LORE) is a rule-based model-agnostic local explainer technique.[10] The main idea of LORE is to learn an interpretable model on a synthetic data generated through a genetic algorithm technique and then drive explanations in the form of decision rules explaining the main causes of the decision of the instance to be explained, in addition to a set of the counterfactual rules, suggesting the main changes in the instance to be explained which could results in different decision. More specifically, given a black-box model $b$, and the instance to be explained $x$ labeled with outcome $y$, LORE builds a decision tree model of balanced synthetic data set $Z$ of neighbor instances to $x$ through generic algorithm such that $Z$ includes instances from both decision values $Z = Z_y \cup Z'_y$, where instances $z \in Z_y$ and $z \in Z'_y$ are synthetic examples from the same class of $x$ and from the other class $y'$, respectively. The genetic algorithm generates instances by maximizing the following fitness functions:

$$fitness_x^y = I_{b(x)=b(z)} + (1 - d(x,z)) - I_{x=z},$$
$$fitness_x^{y'} = I_{b(x)\neq b(z)} + (1 - d(x,z)) - I_{x=z},$$

where $d$ is a distance function, $I_{true} = 1$, and $I_{false} = 0$. The first fitness function aims to find instances from the same class of $x$ and close to $x$ (term $1 - d(x,z)$), but not the same (term $I_{x=z}$), while the second fitness function aims to find instances close to $x$ but from different class. One of the main limitations of neighborhood generation of LORE is that it does not guarantee that it would find instances from different classes. In this case, LORE tries to find instances from the testing data set that are close to the instances generated from the genetic algorithm. The local explanation for $x$ consists of two parts which are (i) simple logic rule extracted from the decision tree and corresponding to the path in the tree that explains why instance $x$ is predicted as class $y$ and (ii) a set of counterfactual rules explained the main features that could reverse the outcome of $x$.

## 4.5 | ILIME

Influence-based Local Interpretable Model-agnostic Explanation (ILIME)[11] is similar to LIME in the way it generates perturbed samples around the instance to be explained. ILIME fits a linear model on the synthetic data set and uses the regression coefficient to identify the importance of the features. One of the main limitations of LIME is that the explanation it provides heavily depends on the weights assigned to the perturbed samples. ILIME addresses this issue by weighting the perturbed instances by their influence on the instance to be explained in addition to the distance to the instance to be explained. One way to measure the influence of a particular example on the instance to be explained is to retrain the model which is computationally expensive. Instead of retraining the model to get the influence of a particular instance on the instance to be explained, ILIME uses influence functions[36] which is based on upweighting an instance in the loss function by a small amount $\epsilon$ in the empirical risk and then approximating the loss using gradient and Hessian matrix that captures the effect of a particular instance on the model.

## 4.6 | MAPLE

Model Agnostic Supervised Local Explanations (`MAPLE`) is a model-agnostic local interpretability technique[12] that combines the idea of using random forests as a technique for supervised neighborhood selection for local linear modeling, known as SILO,[37] with a feature selection technique knows as DStump.[38] More specifically, SILO defines a local neighborhood by assigning a weight to each training instance, in the training data set used to train the model being explained, based on the number of its occurrences in the same leaf node as the instance to be explained across all the trees of the random forest. Then DStump assign a score for each feature which is based on the reduction of the impurity on the class label when each feature is used as a split node at the root of the trees in the forest. DStump chooses the subset of features with the highest score. Finally, MAPLE combines SILIO's local training distribution with the best feature set selected from the DStump to solve a weighted linear regression problem. MAPLE can be used in almost the same way as an explainer for a black-box model or as a predictive model; the only difference is that in the first case MAPLE is fit on the prediction of the black-box model, while in the second case MAPLE is fit on the response variable. MAPLE has several interesting properties that are summarized as follows: (i) avoids the trade-off between model performance and model interpretability, as MAPLE is a highly accurate predictive model (as accurate as tree ensembles) that capable of providing example-based and local explanations and (ii) detects global patterns by leveraging local training distributions, which distinguish MAPLE from other interpretability frameworks.

# 5 | EXPERIMENTAL SETUP

## 5.1 | Data sets

In our experiments, we used two types of data sets: *tabular data sets* and *text data sets*. The tabular data sets of this study have been collected from patients who underwent treadmill stress testing by physician referrals at Henry Ford Affiliated Hospitals in metropolitan Detroit, MI in the United States, FIT Project.[39] In particular, the data have been obtained from the electronic medical records, administrative databases, and the linked claim files and death registry of the hospital between 1 January 1991 and 28 May 2009.[39] Study participants underwent routine clinical treadmill exercise stress testing using the standard Bruce protocol between 1 January 1991 and 28 May 2009. The data set includes 43 attributes containing information on vital signs, diagnosis and clinical laboratory measurements. Examples of these attributes include *sex*, *age*, *race*, *%heart rate achieved*, *resting systolic blood pressure*, *resting diastolic blood pressure*, *obesity*, *hypertension*, *history of smoking*, and *METS*. In the experiments of this work, we have used the two tabular data sets of our previous studies: `mortality` data set and the `diabetes` data set. The cohort of the mortality data set includes 34 212 patients who completed 10-year follow-up, while the cohort of the diabetes data set includes 32 555 patients who completed 5-year follow-up. The text data sets used in this study are the `Drug Review` (`Drugs.com`) and `Side Effects` (`Druglib.com`) data sets form the UCI Repository[2]. Both data sets provide patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting the overall patient satisfaction. Reviews in `Druglib.com` data set are grouped into reports benefits and side effects. For both of `Drug Review` and `Druglib.com` data sets, a review is considered positive if it has five or more starts and negative otherwise.

### 5.2 | Machine learning model

In all the experiments on tabular data, random forest is the model that has shown the best performance in predicting the risk of mortality and diabetes over these data sets. For more information about the details of the data set and modeling process of this study, we refer the interested readers to References 40,41. In all experiments on text data, we preprocess each text data set as follows. First, we removed stop words that do not have a sentiment value such as "the" using stopwords corpus from *nltk* package[3]. Second, we removed the `HTML` tags for the text to prevent them from taking place in our dictionary. Finally, we used snowball stemmer in *nltk* to remove morphological affixes from words. This is done to prevent frequent words from taking more than one place in our dictionary. We used unigram-bag-of-words features that are term frequency inverse document frequency normalized. The bag-of-words feature vectors along with the original labels are used to train a random forest model for predicting whether the review is positive (more than five stars) or negative (less than or equal five stars) in the drug review data set and whether the side effect is mild, moderate, severe, extreme, or extremely severe in the side effects data set.

## 6 | EXPERIMENTAL RESULTS

### 6.1 | Tabular data

In our experiments, we follow the same pipeline on mortality and diabetes data sets: (i) partition the data set into 80% for training and 20% for testing, (ii) train a random forest model on the training data set, (iii) for each instance in the testing data set, use LIME, SHAP, Anchors, LORE, ILIME, and MAPLE to explain the prediction of the model. For LORE framework, we set the population size to be 500 and the number of generations to be 5.

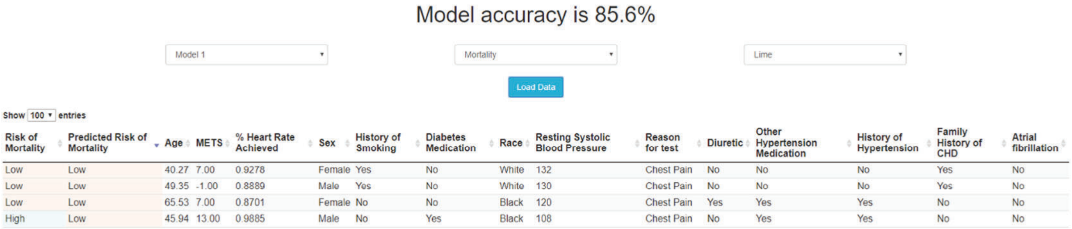
[2] https://archive.ics.uci.edu/ml/
[3] https://www.nltk.org/

Table 1 shows the experimental results on the `mortality` and `diabetes` data sets, respectively. The numbers in this table represent the percentage of instances that satisfy the defined metrics. For each row metric, we highlighted the highest performance in bold font and underlined the lowest performance. For the `identity metric`, the LIME technique has shown the worst performance on the two data sets. The reason behind that is that the sampling technique used to generate the data set that acts as the training data set for the linear model used to approximate the behavior of the complex black-box model. More specifically, such training data set is generated by sampling instances around the instance to be explained uniformly at random. LORE is the second worst technique for the `identity metric` due to the randomly generation process of the neighborhood that used to build the decision tree. On the other side, the SHAP and MAPLE techniques satisfy the identity metric for all tested instances. For the mortality data set, the LORE technique achieves the highest performance for the stability metric (100%) followed by the MAPLE technique (94%), while SHAP comes in the last place (75%). For the diabetes data set, LORE and MAPLE achieve the highest performance for the stability metric (95.5%) followed by ILIME (81%), while LIME comes in the last place (52%). For the diabetes data set, all the interpretability frameworks satisfy the separability metric for all tested instances. For the separability metric on the mortality data set, all the frameworks achieve comparable performance between 97% and 100%. For the similarity metric, SHAP achieves the highest performance on diabetes (6.06), while MAPLE achieves the highest performance on mortality data set (1.77). SHAP comes in the second place for the similarity metric on the mortality data set (3.23), while MAPLE comes in the second place on the diabetes data set (6.556). For the similarity metric, LIME comes in the last place on diabetes data set achieving 13.13, while LORE comes in the last place on the mortality data set achieving 10.925. For the diabetes and mortality data sets, LIME and SHAP achieve comparable average processing time between 0.21 and 0.23 seconds. LORE framework takes the longest average processing time on diabetes data set (1639 seconds), while MAPLE takes the longest average processing time on mortality data set (2026.02 seconds). The huge processing time taken by LORE was mainly because the majority of the instances to be explained in both the diabetes and mortality data sets, the neighborhood generation process generates synthetic instances from only one class and hence a set of instances close to the synthetic generated ones are retrieved from the testing data set which makes the process quite lengthy. For MAPLE framework, the time taken to explain an instance consists of two main components. The first one is the time taken to build the random forest model which depends on the size of the training data set used to train the black-box model (calculated only once for all instances to be explained). The second component, which calculated for each instance to be explained, consists of the time taken to obtain the weights for training instances, the time taken to select the features, and the time taken to build the linear model. The reported time in Table 1 consists of the summation of the two time components. For the trust metric, Anchors achieves the highest performance on both data sets (55% on mortality data set and 63% on diabetes data set), followed by ILIME (53% on mortality data set and 58% on diabetes data set). LORE achieves the lowest performance for the trust metric on the mortality data set (18%), while MAPLE achieves the lowest performance on diabetes data set (43%).

For `Bias detection`, we used visual analytics methods to detect the bias on the mortality data set inspired by Krause et al.[42] In particular, we compare between two different user interfaces (tabular interface and aggregate interface) for detecting the bias in each of the studied frameworks. We created a biased data set from the mortality data set such that the bias is detectable in both interfaces. The biased data set is created such that the smoking feature is inversely related to the risk of mortality; if the patient is a smoker then the patient is at low risk of mortality which is

**TABLE 1** Evaluation of interpretability frameworks on tabular data sets

| | Mortality | | | | | | Diabetes | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Metric** | **LIME** | **Anchors** | **SHAP** | **LORE** | **ILIME** | **MAPLE** | **LIME** | **Anchors** | **SHAP** | **LORE** | **ILIME** | **MAPLE** |
| Identity | 0% | 23% | **100%** | 7.25% | 20% | **100%** | 0% | 11% | **100%** | 0% | 15% | **100%** |
| Stability | 83% | 80% | 75% | **100%** | 85% | 94% | 52% | 74% | 60% | **95.5%** | 81% | **95.5%** |
| Separability | **100%** | 99% | 98% | 97% | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** |
| Similarity | 7.3 | 6.33 | 3.23 | 10.925 | 4.2 | **1.77** | 13.13 | 9.96 | **6.06** | 11.104 | 6.7 | 6.556 |
| Time (s) | **0.23** | 9.1 | **0.23** | 1642 | 2.35 | 2026.02 | **0.21** | 10.38 | 0.22 | 1639 | 2.1 | 375.02 |
| Trust | 51% | **55%** | 41% | 18% | 53% | 22% | 58% | **63%** | 50% | 54% | 58% | 43% |



Model accuracy is 85.6%

| Risk of Mortality | Predicted Risk of Mortality | Age | METS | % Heart Rate Achieved | Sex | History of Smoking | Diabetes Medication | Race | Resting Systolic Blood Pressure | Reason for test | Diuretic | Other Hypertension Medication | History of Hypertension | Family History of CHD | Atrial fibrillation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Low | Low | 40.27 | 7.00 | 0.9278 | Female | Yes | No | White | 132 | Chest Pain | No | No | No | Yes | No |
| Low | Low | 49.35 | -1.00 | 0.8889 | Male | Yes | No | White | 130 | Chest Pain | No | No | No | Yes | No |
| Low | Low | 65.53 | 7.00 | 0.8701 | Female | No | No | Black | 120 | Chest Pain | Yes | Yes | Yes | No | No |
| High | Low | 45.94 | 13.00 | 0.9885 | Male | No | Yes | Black | 108 | Chest Pain | No | Yes | Yes | No | No |

**FIGURE 1** Tabular user interface for the mortality model using unbiased data on LIME interpretability framework

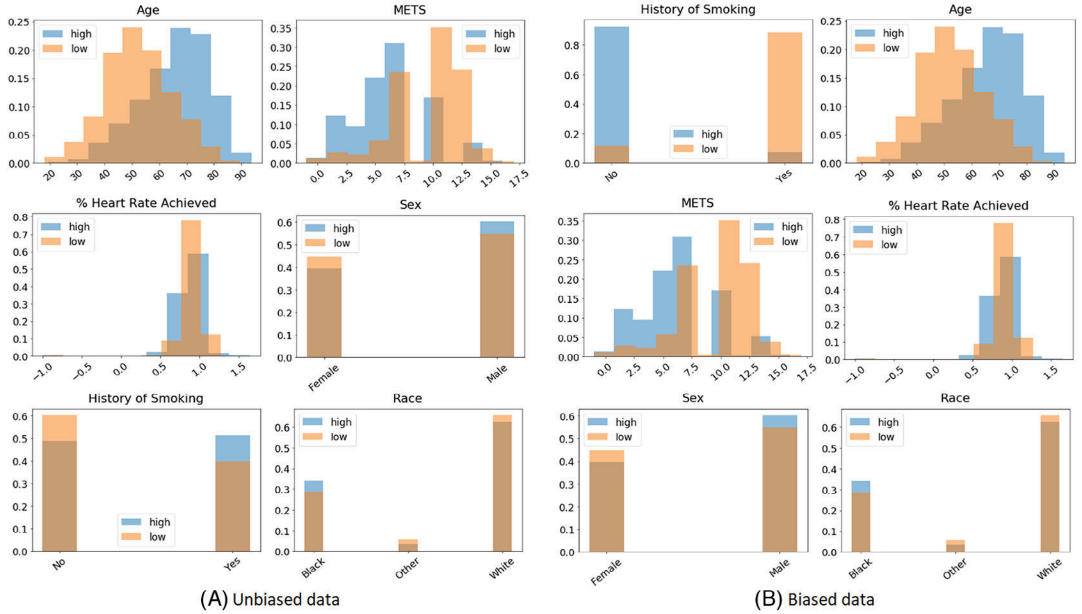
counterintuitive. The bias is created such that the biased model achieves higher testing accuracy than the unbiased model. The bias is created with the same degree in both training and testing data sets. We trained a random forest model on both data sets. The testing accuracies on the unbiased and biased data sets are 86% and 94%, respectively. The users evaluated the bias were people with basic knowledge in medical domain.

In each of the biased and unbiased testing data sets, we get the explanation of each instance from the six different interpretability frameworks based on all 14 features. To make the interface simple, we consider features that contribute to the prediction without specifying whether the contribution is toward or against the prediction. For comparing the interfaces for bias evaluation, we compare patients who are at high risk mortality and patients of low risk of mortality. In particular, we have implemented and used the following two user interfaces:

**Tabular user interface**: The tabular user interface for the mortality model is shown as a tabular view such that each row represents the explanation features used (Figure 1). The user can navigate between the biased model and the unbiased one across the six interpretability frameworks. The accuracy of each of the biased and unbiased model is shown at the top of the interface. For LIME, SHAP, ILIME, and MAPLE, the columns in the table are sorted according to the average contribution of features on the testing data set in each of explanation framework such that the leftmost column has the highest average contribution in the explanations. For Anchors and LORE, the columns are sorted according to the number of occurrences of features in the explanations such that the leftmost column has the highest occurrence across the testing data set. To make it easier to compare between patients who are at high risk and who are at low risk of mortality, we use different color for each group.

**Aggregated user interface**: The aggregate user interface shows the distribution of features' values as histograms sorted such that the top-left histogram in each interpretability framework is for the feature with the highest average contribution and the bottom-right histogram is for the feature with the lowest average weight. The aggregated user interface for LIME framework, shown in Figure 2, illustrates only six features, due to space limitations. For each histogram, the height of the bars represent the percentage of instances in each group. The aggregated user interfaces for all the interpretability frameworks using all features are available in the project repository.

We conducted a user study to evaluate the ability to detect the bias by comparing the explanations of the biased and unbiased testing data sets using the tabular and aggregated user interfaces. This study involved 23 fresh graduate students. We divided the participants into two groups, one group to evaluate the tabular user interface and the other to evaluate the aggregated user interface. For both groups, we introduced the meaning of accuracy and how it is used to evaluate the model's performance. We then explained the mortality data set by informing the participants with the meaning of the features and how they logically affect the output class, that is, as age increases,



(A) Unbiased data

(B) Biased data

**FIGURE 2** Aggregate explanation user interface for the mortality model on LIME interpretability framework

there is a higher chance of mortality. Finally, we explained to the participants how to use the evaluation interface. Out of the 23 participants, only 20 responses were valid. We evaluated validity by asking the participants which model has better accuracy which was a pretty obvious question. All the participants were able to identify the bias from the tabular interface, while only 80% were able to do the same using aggregated user interface. It is clear from the results that the tabular user interface enables participants to better detect the bias. Based on these results, we ranked the interpretability frameworks that enabled the participants to correctly detect the bias using the tabular and aggregated user interfaces as SHAP, MAPLE followed by Anchors, while LIME, ILIME, and LORE were all tied in the last place.

## 6.2 | Text data

For each of the `Drug Review` data set and the `Side Effects` data set, we follow the same pipeline used on tabular data, except that the data set has been partitioned into 70% for training the model and 30% for testing the model. The `identity`, `separability`, `similarity`, `stability`, `time`, and `trust` metrics are measured on text data in the same ways as of tabular data. The LORE framework is computationally expensive as it takes more than an hour and half to generate the explanation for a single instance from `Drug Review` data set and the `Side Effects` data set, which make it infeasible to evaluate it on the text data sets included in this study.

Table 2 shows the results of evaluating the six different interpretability techniques on the `Drug Review` data set and the `Side Effects` data set, respectively. For each row metric, we highlighted the highest performance in bold font and underlined the lowest performance. The results show that MAPLE achieves the highest identity performance of 95% on both data sets,

**TABLE 2** Evaluation of interpretability frameworks on text data sets

| | Drug review | | | | | Side effects | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Metric** | **LIME** | **Anchors** | **SHAP** | **ILIME** | **MAPLE** | **LIME** | **Anchors** | **SHAP** | **ILIME** | **MAPLE** |
| Identity | 0% | 66% | 12% | 19% | **95%** | 0.5% | 78% | 50% | 20% | **95%** |
| Stability | 74% | 82% | 85% | 80% | **90%** | 51% | 70% | 53% | 60% | **80%** |
| Separability | **100%** | 83% | 99% | 90% | 99% | **100%** | 96% | 99% | 97% | 100% |
| Similarity | **2.1** | 2.28 | 6.47 | 2.11 | 2.0 | **8.61** | 17.4 | 21.3 | **8.61** | 8.7 |
| Time (s) | 1.13 | 10.3 | **0.8** | 8.15 | 20135 | 0.93 | 3.4 | **0.46** | 1.97 | 540 |
| Trust | 52% | **57%** | 43% | 50% | 15% | 61% | **70%** | 53% | 61% | 50% |

followed by Anchors (66% on drug review data set and 78% on the side effects data set). LIME achieves the lowest identity performance of 0% and 0.5% on the drug review and side effects data sets, respectively. For the stability metric, MAPLE achieves the highest performance on both data sets (90% on drug review data set and 80% on the side effects data set). For the stability metric, SHAP comes in the second place on the drug review data set (85%), while Anchors comes in the second place on the side effects data set (70%). ILIME and Anchors achieve comparable performance for the stability metric on drug review data set . LIME achieves the lowest stability performance of 74% and 51% on the drug review and side effects data sets, respectively. For the separability metric on both data sets, LIME, SHAP, and MAPLE achieve the highest comparable performance, while ILIME comes in the second place. Anchors achieves the lowest performance on separability metric on both data sets (83% on drug review data set and 96% on side effects data set). On both data sets, LIME, ILIME, and MAPLE achieve the highest comparable performance for the similarity metric (between 2 and 2.11 on drug review data set and between 8.61 and 8.7 on side effects data set), followed by Anchors, while SHAP comes in the last place (6.47 on drug review data set and 21.3 on side effects data set). SHAP has the lowest average time to output explanation, while MAPLE has the highest on both data sets that is mainly because the MAPLE explanation times depends on the time taken to build the random forest that mainly depends on the size of the training data set. For the trust metric, Anchors achieves the highest performance on both data sets (57% drug review data set and 70% side effects data set), followed by LIME (52% drug review data set and 61% side effects data set). MAPLE achieves the lowest performance for the trust metric on the both data sets.

## 7 | CONCLUSION

In this work, we address one of the main challenges in the context of quantifying the quality of machine learning interpretability techniques. More specifically, we propose four metrics that can be used as unified quantitative measure for assessing the quality of different interpretability techniques. Additionally, these metrics may contribute toward improving the existing interpretability techniques. We evaluated six different frameworks namely, `LIME`, `SHAP`, `Anchors`, `ILIME`, `LORE`, and `MAPLE` on different types of data (tabular and text). The results show that there is no clear winner. In other words, there is no single interpretability technique that can achieve the best performance for all metrics across different types of data. Thus, it is crucial for the users of the interpretability techniques to clearly specify their interpretability focus (metric) and understand the strength and weakness of each interpretability techniques so that they can achieve their goal for getting reasonable and effective explanations for their used complex machine learning model. As a future work, we plan to extend our work to evaluate the performance of different interpretability frameworks on medical image data sets.

## ACKNOWLEDGMENTS

The work of Radwa Elshawi is funded by the European Regional Development Funds via the Mobilitas Plus programme (MOBJD341). The work of Sherif Sakr and Youssef Sherif is funded by the European Regional Development Funds via the Mobilitas Plus programme (grant MOBTT75).

## DATA AVAILABILITY STATEMENT

Data openly available in a public repository https://github.com/DataSystemsGroupUT/Interpretability-comparison

## ORCID

*Sherif Sakr* https://orcid.org/0000-0002-2503-523X

## REFERENCES

1. Darcy AM, Louie AK, Roberts LW. Machine learning and the profession of medicine. *Jama*. 2016;315(6);551-552.
2. Basu-Roy S, Teredesai A, Zolfaghar K, et al. Dynamic hierarchical classification for patient risk-of-readmission. Paper presented at: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia; 2015:1691-1700.
3. Futoma J, Morris J, Lucas J. A comparison of models for predicting early hospital readmissions. *J Biomed Inform*. 2015;56:229-238.
4. Lim BY, Dey AK, Avrahami D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. Paper presented at: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, MA; 2009.
5. Goodman B, Flaxman S. European union regulations on algorithmic decision-making and a “right to explanation”. *AI Mag*. 2017;38(3):50-57.
6. Lipton ZC. The mythos of model interpretability. *Queue*. 2018;16(3):31-57.
7. Ribeiro MT, Singh S, Guestrin C. Why should i trust you? Explaining the predictions of any classifier. Paper presented at: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco; 2016.
8. Ribeiro MT, Singh S, Guestrin C. Anchors: high-precision model-agnostic explanations. Paper presented at: Proceedings of the AAAI Conference on Artificial Intelligence, Louisiana, USA; 2018.
9. Štrumbelj E, Kononenko I. A general method for visualizing and explaining black-box regression models. Paper presented at: Proceedings of the International Conference on Adaptive and Natural Computing Algorithms, Ljubljana, Slovenia; 2011:21-30.
10. Guidotti R, Monreale A, Ruggieri S, Pedreschi D, Turini F, Giannotti F. Local rule-based explanations of black box decision systems; 2018:arXiv preprint arXiv:1805.10820.
11. ElShawi R, Sherif Y, Al-Mallah M, Sakr S. ILIME: local and global interpretable model-agnostic explainer of black-box decision. Paper presented at: Proceedings of the European Conference on Advances in Databases and Information Systems; 2019:53-68.
12. Plumb G, Molitor D, Talwalkar AS. Model agnostic supervised local explanations. *Advances in Neural Information Processing Systems*; Red Hook, NY: Curran Associates Inc; 2018:2515-2524.
13. Herman B. The promise and peril of human evaluation for model interpretability; 2017. arXiv preprint arXiv:1711.07414.
14. Doshi-Velez F , Kortz, M., Budish, R et al. Accountability of AI under the law: the role of explanation; 2017. arXiv preprint arXiv:1711.01134.

15. Mohseni S, Zarei N, Ragan ED. A survey of evaluation methods and measures for interpretable machine learning; 2018. arXiv preprint arXiv:1811.11839.
16. Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: a survey on methods and metrics. *Electronics*. 2019;8(8):832.
17. Doshi-Velez F, Kim B. Considerations for evaluation and generalization in interpretable machine learning. *Explainable and Interpretable Models in Computer Vision and Machine Learning*. New York, NY: Springer; 2018:3-17.
18. Honegger M. Shedding light on black box machine learning algorithms: development of an axiomatic framework to assess the quality of methods that explain individual predictions; 2018. arXiv preprint arXiv:1808.05054.
19. Silva W, Fernandes K, Cardoso MJ, Cardoso JS. Towards complementary explanations using deep neural networks. *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. New York, NY: Springer; 2018.
20. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*. Vol 30. NY; United States: Curran Associates, Inc; 2017.
21. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. *ACM Comput Surv*. 2018;51(5):93.
22. White A, Garcez ADA. Measurable counterfactual local explanations for any classifier. Paper presented at: 24th European Conference on Artificial Intelligence - ECAI 2020, Santiago de Compostela, Spain; 2019.
23. Cook RD. Detection of influential observation in linear regression. *Technometrics*. 1977;19(1):15-18.
24. Domhan T, Springenberg JT, Hutter F. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. Paper presented at: Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina; 2015.
25. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: the all convolutional net. San Diego, CA: ICLR (workshop track); 2015.
26. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M. Smoothgrad: removing noise by adding noise; Paper presented at: Workshop on Visualization for Deep Learning, ICML, 2017, Sydney, Australia; 2017.
27. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. Paper presented at: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia; Vol. 70; 2017:3319-3328.
28. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. Paper presented at: Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy; 2017:618-626.
29. Kim B, Wattenberg M, Gilmer J, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). Paper presented at: International conference on machine learning, PMLR, Stockholm, Sweden; 2018:2668-2677.
30. Robnik-Šikonja M, Bohanec M. Perturbation-based explanations of prediction models. *Human and Machine Learning*. New York, NY: Springer; 2018:159-175.
31. Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N. Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. Paper presented at: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2015.
32. Kaufmann E, Kalyanakrishnan S. Information complexity in bandit subset selection. Paper presented at: Proceedings of the Conference on Learning Theory, Princeton, NJ; 2013:228-251.
33. Cover TM, Thomas JA. *Elements of Information Theory*. Boca Raton, FL: John Wiley & Sons; 2012.
34. Shapley LS. A value for n-person games. *Contribut Theory Games*. 1953;2(28):307-317.
35. Kononenko I, Kononenko I. An efficient explanation of individual classifications using game theory. *J Mach Learn Res*. 2010;11(Jan):1-18.
36. Koh PW, Liang P. Understanding black-box predictions via influence functions; International Conference on Machine Learning, Sydney, Australia; 2017:1885-1894.
37. Bloniarz A, Talwalkar A, Yu B, Wu C. Supervised neighborhoods for distributed nonparametric regression. *Artif Intell Stat*. 2016;51:1450-1459.
38. Kazemitabar J, Amini A, Bloniarz A, Talwalkar AS. Variable importance using decision trees. *Advances in Neural Information Processing Systems*; Red Hook, NY: Curran Associates Inc; 2017:426-435.

39. Al-Mallah MH, Keteyian SJ, Brawner CA, Whelton S, Blaha MJ. Rationale and design of the henry ford exercise testing project (the FIT project). *Clin Cardiol*. 2014;37(8):456-461.
40. Sakr S, Elshawi R, Ahmed AM, et al. Comparison of machine learning techniques to predict all-cause mortality using fitness data: the Henry ford exercIse testing (FIT) project. *BMC Med Inform Decis Mak*. 2017;17(1):174.
41. Alghamdi M, Al-Mallah M, Keteyian S, Brawner C, Ehrman J, Sakr S. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: the Henry Ford exercise testing (FIT) project. *PLoS One*. 2017;12(7):e0179805.
42. Krause J, Perer A, Bertini E. A user study on the effect of aggregating explanations for interpreting machine learning models. Paper presented at: Proceedings of the KDD Workshops; 2018;1-14.

**How to cite this article:** ElShawi R, Sherif Y, Al-Mallah M, Sakr S. Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. *Computational Intelligence*. 2021;37:1633–1650. https://doi.org/10.1111/coin.12410