A comparative study of additive local explanation methods based on feature influences

Emmanuel Doumard
emmanuel.doumard@irit.fr
Université de Toulouse-Paul Sabatier, IRIT, (CNRS/UMR 5505)
Toulouse, France

Julien Aligon
julien.aligon@irit.fr
Université de Toulouse-Capitole, IRIT, (CNRS/UMR 5505)
Toulouse, France

Elodie Escriva
elodie.escriva@kaduceo.com
Kaduceo
Université de Toulouse-Capitole, IRIT, (CNRS/UMR 5505)
Toulouse, France

Jean-Baptiste Excoffier
jeanbaptiste.excoffier@kaduceo.com
Kaduceo
Toulouse, France

Paul Monsarrat
paul.monsarrat@univ-tlse3.fr
RESTORE Research Center & Artificial and Natural Intelligence Toulouse Institute ANITI & Oral Medicine Department
Toulouse, France

Chantal Soulé-Dupuy
chantal.soule-dupuy@irit.fr
Université de Toulouse-Capitole, IRIT, (CNRS/UMR 5505)
Toulouse, France

ABSTRACT

Local additive explanation methods are increasingly used to understand the predictions of complex Machine Learning (ML) models. The most used additive methods, *SHAP* and *LIME*, suffer from limitations that are rarely measured in the literature. This paper aims to measure these limitations on a wide range (304) of OpenML datasets, and also evaluate emergent coalitional-based methods to tackle the weaknesses of other methods. We illustrate and validate results on a specific medical dataset, SA-Heart. Our findings reveal that *LIME* and *SHAP*'s approximations are particularly efficient in high dimension and generate intelligible global explanations, but they suffer from a lack of precision regarding local explanations. Coalitional-based methods are computationally expensive in high dimension, but offer higher quality local explanations. Finally, we present a roadmap summarizing our work by pointing out the most appropriate method depending on dataset dimensionality and user's objectives.

KEYWORDS

Explainable Artificial Intelligence (XAI), Prediction explanation, Machine learning,

1 INTRODUCTION

Machine Learning (ML) represents a real revolution in various domains, such as finance, insurance, healthcare, biomedical. However, machine learning models give a prediction without necessarily being accompanied by an understandable explanation. These models, often referred as "black-boxes", raise the challenging question of how humans can understand the determinants of the prediction. Explainability is also more than a technological problem, it involves among other ethical, societal and legal issues. In healthcare, this may involve the professional being able to explain to the patient how the algorithm works and the criteria for the decision process. The results of ML models must therefore be expressed in a way that can be understood by domain-experts, like medical practitioners [1, 5]. Since *SHAP* [15], machine learning experts show a very clear interest for the additive methods as a huge number of works using these methods are published each year. The additive methods include *LIME* [19], *SHAP* [15] and more recently the coalitional-based methods [8]. The user-friendly representation of explanations, based on feature influences, allows domain and non-domain experts to better understand models predictions [18]. Existing explanation methods are model-specific or model-agnostic depending on whether they can be applied to some or all types of machine learning models, with local or global explanations to understand either an individual prediction or the behaviour of the model as a whole. While these methods have been evaluated in a number of contexts, no *in-depth* evaluation is available for a rational choice of one technique over another. The objective of this work is to study the advantages and disadvantages of using each additive method to provide pertinent insights. In particular, we study the effects of the models used and the type of dataset considered on the feature influences (both at the instance and feature level).

The paper is organised as follows. Section 2 reviews the existing work, classifying and comparing explanation methods for tabular data. Section 3 describes the four additive methods to be compared in this paper. The experiments are presented in Section 4 where we study the explanation characteristics, the impact of the predictive model on explanation profiles, highlighting the behavior of explanation methods based on a practical medical use case. Conclusive lessons-learned are then detailed in Section 5.

2 RELATED WORKS

Few works [3, 27] exist in the literature to classify and categorize machine learning explanation methods. In [18], a complete description of explanation approaches from literature is given. In particular, the authors explain their advantages and disadvantages, giving an overview of their limits. For example, even if the *LIME* and *SHAP* approaches are model-agnostic and human-friendly, they suffer from no consideration of feature correlation and possible instability of the explanations. Another paper tackling the limits of the additive methods (*LIME* and *SHAP*) is presented in [23]. The paper shows that biased classifiers can fool explanation methods, whose problem is even more accentuated on *LIME*.

Comparative studies between local explanation methods are also available, such as [6, 8, 16]. In [8], a new additive method was proposed based on Shapley values and taking into account

© 2022 Copyright for this paper by its author(s). Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

CEUR-WS.org/Vol-3130/paper4.pdf

feature correlation. This method was compared with *LIME* and *SHAP* through computation time and accuracy score. For this last measure, the authors consider as baseline the *complete method*, computing all Shapley values with each possible coalition of features. The authors show that their proposal is competitive with the literature, both in accuracy and in computation time. In [16], *LIME* and *SHAP* are used in a context of feature selection and compared to a Mean Decrease Accuracy (MDA) approach. A stability measure indicates that a feature selection obtained with *LIME* or *SHAP* seems more stable than via MDA. In [6], the authors compare 6 local model-agnostic techniques using custom quantitative measures, such as similarity, bias detection, execution time, and trust. From these experiments, no single method stands out for all metrics and all data sets. Each ones have strengths and weaknesses based on the metrics used and choices between methods can only be made based on the users' goal and dataset.

The latest results are therefore indications for the absence of a single method that would provide the best explanations in all situations. However, none of these previous works clearly indicates in which situation a method should be preferred to another one. Consequently, our aim is to give the key factors to make an informed decision among the existing additive methods. As indicated in [17], evaluating explanations methods is very subjective and no consensus yet exists to propose relevant metrics. As all additive methods give an influence score for each feature, we propose to compare them based on these influences. From there, we want to analyse and compare the effects of different predictive models and dataset on these influence scores.

3 ADDITIVE METHODS TO COMPARE

Additive methods are described as explanation models that produce a vector of weights to represent the influence of each feature, the sum of which approximates the output of the original model. Explanations can be computed for a single instance, so for every instance of a data set, hence the term "local". In this section, we explore several existing methods that fit this definition. We focus on post-hoc methods that deliver their explanations for a given model already trained. Methods used in this study are all agnostic, meaning that they can be applied to any kind of machine learning model, except for the *TreeSHAP* method [14] that is designed specifically for tree-based models.

3.1 LIME

LIME method is a well-known local explanation method described in [19]. *LIME* uses explainable models to locally approximate a complex black-box model and, for each instance, explain the influence of each feature on the prediction. For each instance to be explained, *LIME* generates new data in a close neighborhood and computes the predictions of these new instances with the black-box model. A regressor linear model, an interpretable model, is trained with the new dataset. This local model is then used to explain the prediction of the instance of interest in the form of a weight vector associating each feature with its influence on the prediction. A well-known limitation of *LIME* is the restrictive hypothesis on which *LIME* is based, such as local linearity and feature independence [9, 23]. Defining the locality around an instance of interest can also be a challenge, as the fit of the surrogate model has a significant impact on the accuracy of the explanations [11] as well as their stability [7].

The full implementation of *LIME* is available on GitHub : https://github.com/marcotcr/lime.

3.2 Shapley Values (complete method)

To explain individual predictions, a method based on Shapley values is described in [24, 25, 28]. Shapley values 'fairly' weight groups of features according to their relative importance to a defined gain [21]. In machine learning, the gain can be linked to the prediction made by the model. Influences of each feature are computed based on its impact on the prediction for each coalition of features. The explanation method based on Shapley values is called the *complete method*. All coalitions are evaluated with and without each feature and the change on the prediction is used to compute the influence of the feature. The *complete method* can be used as a baseline to compare other methods as it is an exhaustive method close to the original intuition behind feature influence [8]. This method is however very expensive to compute, with an exponential complexity in relation to the number of features in the dataset.

Several more recent methods, including *SHAP* [15] and coalitional methods [8], are based on Shapley values with the aim to solve limitations of the *complete method*.

3.3 SHAP

SHAP (SHapley Additive exPlanations) [15] method worked on improving computation time and explanation precision, especially for tree-based models [14]. It combines *LIME* [19] and Shapley values [28], along with other methods from the literature [2, 4, 13, 22], in a unique framework to produce local explanations. The main idea is to create perturbations to simulate the absence of a feature and to use a linear local model to approximate the change in the prediction, as in *LIME*. This avoids retraining the complex model without the feature of interest. Local explanations can be aggregated to explain the global behaviour of the model. Global and local explanations are then consistent with each other as they have the same foundation. *SHAP* includes an agnostic explainer, *KernelSHAP*, as well as model-specific explainers, such as *TreeSHAP*, *LinearSHAP* or *DeepSHAP* for tree-based models, linear models and deep models respectively. While commonly used in Machine Learning context [12], *SHAP* still suffers from lack of precision [10, 23] mostly due to their restrictive hypothesis (local linearity and feature independance) as with *LIME*. Moreover, computation time is still high for other models than tree-based models [26].

The full implementation of *SHAP* is available in GitHub : https://github.com/slundberg/shap.

3.4 Coalitional-based method

Another agnostic explainer based on Shapley values, the *coalitional method*, was introduced to take into account the interdependence of features and solve some restrictions of SHAP. It uses grouping methods such as *Principal Component Analysis* (PCA), *Spearman correlation factor* (Spearman) and *Variance Inflation Factor* (VIF) to pre-compute groups of features for explanations [8]. These groups are then used as coalitions to compute Shapley values as in the *complete method*. The influence of each feature is defined as its impact on the prediction only on the pre-computed groups of features, approximating the *complete method* and reducing the computational time. Grouping methods are defined with a parameter that changes the number and size of feature groups

Number of features	Number of datasets	Number of instances Min	Max	Mean
1	5	130	9100	3079
2	21	52	5456	901
3	43	60	9989	1729
4	23	96	8641	1016
5	35	62	7129	941
6	27	51	9517	949
7	33	54	4052	499
8	32	52	8192	1473
9	23	52	1473	484
10	37	57	5473	712
11	8	66	4898	942
12	12	123	8192	1175
13	5	178	506	293
Total	304	51	9989	1035

Table 1: Datasets description

in order to prioritise a low computational time or an higher accuracy. As for *SHAP*, local explanations can be aggregated into global explanations with a common foundation to study global and local behavior of the model.

The full implementation of *Coalitional-based method* is available on GitHub :
https://github.com/kaduceo/coalitional_explanation_methods.

4 EXPERIMENTS

In this section, we propose experiments comparing the explanation methods presented in the previous section. The goal is to identify the general behavior of each method and how this behavior eventually differs according to a predictive model (learned from data) and the dimensionality of the data (number of features).

4.1 Experimental protocol

All experiments are run on an Intel Xeon Gold 6230 processor with 125 GB of RAM using Python 3.9.7. All runs are performed on a single core of CPU for optimization and reproducibility. To compare explanation methods, we apply them to a wide range of 304 datasets available on OpenML (www.openml.org). Due to computational constraints of explanation methods, we only considered datasets with at most 13 features, and at most 10 000 instances. We also only considered classification tasks to use comparable predictive models and metrics. We describe the amount and size of datasets per number of features in Table 1.

As an explanation method needs a model to be applied to, we choose four widely used types of ML models for classification: Logistic Regression (LR), Support Vector Machines (SVM), Random Forests (RF) and Gradient Boosted Machines (GBM). For the first three, we use the implementation of Python library scikit-learn version 1.0.1. For GBM, we use the Python library XGBoost version 1.5. We use default values for models hyperparameters. For explanations methods, we use Python libraries shap 0.40 and lime 0.2.0.1.

Then, to be able to compare the explanations, we need to define metrics of interest. In Section 4.2, we present three metrics that we will use for this study.

Section 4.3 aims to compare the four additive methods introduced in Section 3. In particular, we use two distinct coalitional-based methods: the *Complete* method, which serves as reference for an influence deviation measurement (second metric), and the *Spearman* method with a threshold of 25% of all groups of features. Regarding *SHAP*, we use the model-agnostic *KernelSHAP* on all datasets. As this method is very slow to execute if we use the whole dataset as background samples for permutations, we choose to follow *SHAP*'s recommendation[1] by doing a *K-Means* clustering on the input dataset, and then taking the centroids as background samples. We choose $K = 10$ clusters for each dataset, thus naming the method *KernelSHAP10*. In addition, for the two tree-based predictive models XGBoost and Random Forests, we use the model-specific explainer *TreeSHAP* by two implementations. The first one determines *SHAP* values with background samples, similarly to *KernelSHAP* but optimised for tree-based methods. We use the whole dataset as background samples for this method. The second one approximates *SHAP* values by considering the trees structures, and does not need background samples in input, so we name it *TreeSHAPapprox*. Last, we consider *LIME*, which requires a number of perturbed samples to be created for explaining each instance. We choose to set this number to 100 samples for all datasets.

With similar methodology, Section 4.4 identifies the impact of the predictive model on specific explanation methods.

Lastly, we present in Section 4.5 a practical example of the different explanations methods applied to a specific dataset, *SA-Heart*. This dataset is chosen for its medical context (coronary heart diseases), a sufficient number of instances (462) and features (10) to train a coherent model and compute the explanations in acceptable computational times. The underlying idea is to illustrate the highlighted behaviors by taking a concrete example as it could be used by an end user (*e.g.* a physician).

4.2 Metrics of interest

Because of the subjective nature of explanations, there is no consensus on objective mathematical ways to evaluate the explanations. Therefore, to evaluate explanation methods performances and compare them over a high number of datasets, we define three different metrics that only need the influence values given by the method. The first one is the computational time per instance, which is the amount of time taken by a given method to compute the local influences of a whole dataset, divided by the number of instances in the dataset. The second one is a quantification of the average deviation of the influence given by a method from the *Complete* method (see Section 3). This error rate is defined as:

$$err(I, X) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{p} \sum_{k=1}^{p} \left| I_k(X_i) - I_k^C(X_i) \right|$$

where, for a given a dataset, n the number of instances, p the number of features, X_i the features vector for the instance i, $I_k(x)$ the influence of a feature k for a given instance x, a given explanation method and a given machine learning model, and $I_k^C(x)$ the influence given by the *Complete* method for the same model,

[1] *KernelSHAP* documentation includes recommendation to use K-Means algorithm to speed up computation time https://shap-lrjball.readthedocs.io/en/latest/generated/shap.KernelExplainer.html

same feature and same instance. The third metric evaluates the distribution of feature importance assigned by a given explanation. The raw value being not necessarily comparable between explanation methods, the cumulative importance proportion of features given by a method was considered. This metric shows whether an explanation method favours the attribution of great importance to a few features or, on the contrary, a more homogeneous distribution among a larger number of features. The importance of a feature is defined here as the mean absolute value of influence assigned to instances for such feature. For example, in a dataset with 2 features, if a method gives 80% of the importance to the most important feature (and so 20% to the second), it would have a cumulative importance proportion vector of $[0, 0.8, 1]$. We can then compute the (normalised) Area Under Curve (AUC) of such a vector C with :

$$AUC(I, X) = \frac{1}{p} \sum_{i=0}^{p-1} \frac{C_i + C_{i+1}}{2}$$

where C_i is the total importance proportion taken by the i most important features.

As this cumulative sum is sorted by construction from most important to least important features, this value is bound between 0.5 and 1. A value of 0.5 means that the explanation method gives the same importance to all features, while a value of 1 means that the explanation method gives non-zero influences only to a single feature, explaining the model's predictions with a single feature.

4.3 Additive methods comparison

We show in Figure 1 the evolution of the execution time of each method for each predictive model, averaged over datasets that share the same number of features. *LIME*, having a linear complexity with the number of features, is computationally expensive compared to other methods in low dimension (few features), but is less expensive than coalitional-based methods and *KernelSHAP* in higher dimensions. *LIME* also seems to have very low inter-dataset time variability, resulting in smaller error bars on the graph. Coalitional-based methods show an exponential complexity with the number of features, having high execution time in high dimension, but have a similar execution time with other methods in low dimension. *Spearman* method execution time seems naturally correlated to the *Complete* method execution time, taking a fraction of the time (roughly 25%) of the *Complete* method. *KernelSHAP*, despite a limitation on the amount of background samples, has a high execution time in high dimension, comparable to coalitional-based methods for non-tree based methods. For tree-based methods, *KernelSHAP* is slower in low dimension, but faster in high dimension than coalitional-based methods. Last, tree-based explainers seem to have constant execution time per instance no matter the number of features, and the approximate tree path dependent version of *TreeSHAP* has the lowest execution time per instance.

Regarding the second metric, Figure 2 shows the average absolute difference in influence between each method and the *Complete* method (reference). First, we can see that overall, the more features there are in a dataset, the closest (measured by the second metric) the influences are to the *Complete* method. This is probably due to the fact that usually, the more features there are, the less influence amplitude each individual feature has in the prediction. We also note that no matter the model, common methods are ranked in the same way. In low dimension (less than

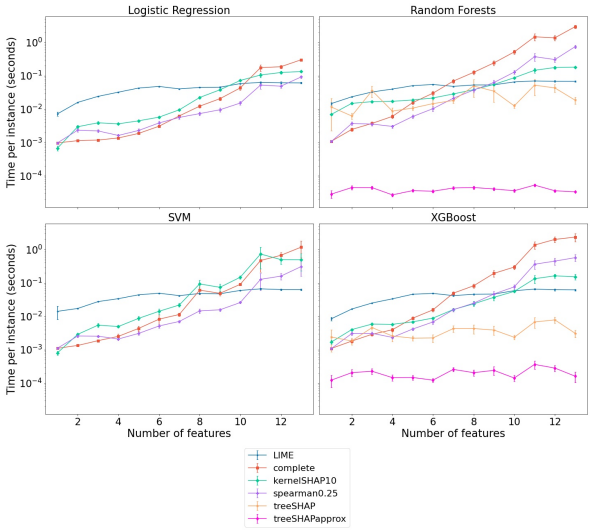

Figure 1: Execution time of each method per instance, averaged by number of features, for each model

6 features), *KernelSHAP* is the closest to the *Complete* method, followed by *Spearman*, while *LIME* is the the farthest. In higher dimensions, *Spearman* becomes more precise than *KernelSHAP*. *TreeSHAP* (both the approximate and the data dependent version) is more precise than *KernelSHAP*, but still less precise than *Spearman* in high dimensions. Note that the approximate version of *TreeSHAP* is not showed on the graph for XGBoost because its implementation forces its *SHAP* values to be in log odds instead of probabilities, making it impossible to compare to other methods.

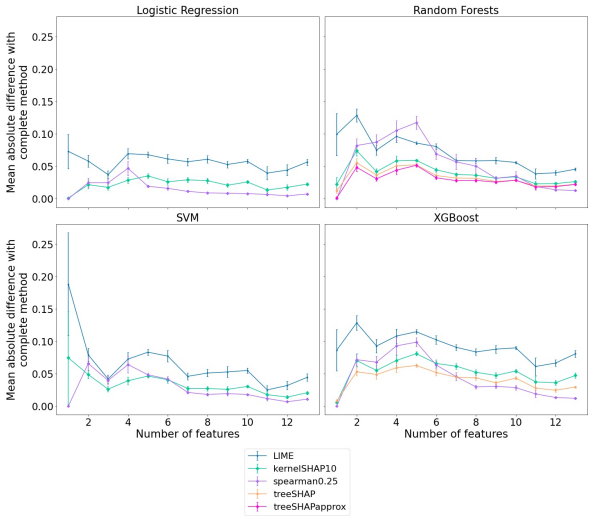

Figure 2: Mean absolute difference of each method with the *Complete*, averaged by number of features, for each model

Finally, we show in Figure 3 an example of the graphical representation of the cumulative feature importance proportion. The figure shows the averaging of the cumulative importance proportion of the most-important features for the 37 datasets having 10 features. This way, for each predictive model and for each method, we obtain a curve from which we compute the third

metric: the AUC of the curve. We see on the figure that some methods present steeper curves than others. For example, with Logistic Regression and SVM, *LIME* gives less proportion of the total importance to the few first most-important features, compared to coalitional-based and *SHAP* methods. For tree-based models, we see that *SHAP*, no matter the method, gives much more importance to the first few most-important features than the other methods.

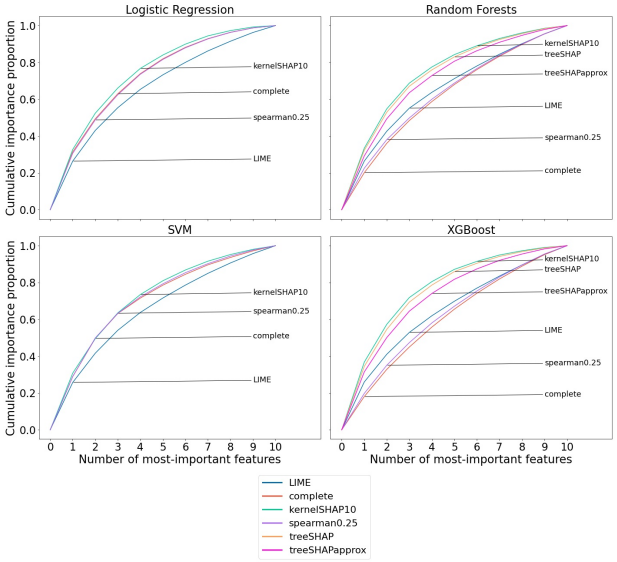

Figure 3: Most-important features cumulative importance proportion by method, for each model. Only influences computed on datasets with 10 features are shown.

According to the method for computing AUC illustrated in Figure 3, we represent the average values of AUC for datasets from 2 to 13 features for each ML model and explanation method in Figure 4. For all models, we can see that *SHAP* methods tend to produce influences with a higher AUC compared to other methods. This means that *SHAP* methods tend to assign most of the feature importance to fewer most-important features, while other methods tend to distribute the feature importance more uniformly over all features. The two coalitional-based methods seem to generate similar AUCs for the features importance. Finally, *LIME* tends to produce influences with lower AUCs for non-tree-based methods, while it produces AUCs closer to the coalitional-based methods for tree-based methods.

4.4 Machine Learning models explanations comparison

We show in Figure 5 the computational time per instance needed to compute the explanations of each predictive model, for each explanation method.

We can see that *LIME*'s execution time has almost no inter-model variability: the computation time per instance is the same no matter the model. For the other methods, the ranking of the method's computational performances according to the model is roughly the same, from slowest to fastest: Random Forests, XGBoost, SVM and Logistic Regression. SVM has overall higher variability, presenting steeper curves and higher error bars. SVM even presents outlying results when applied to *KernelSHAP* in higher dimensions. Overall, we do not observe specific behavior

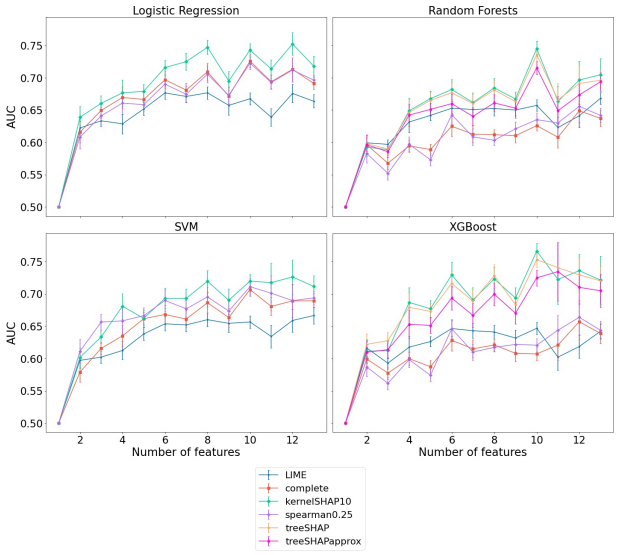

Figure 4: AUC of each method, averaged by number of features, for each model

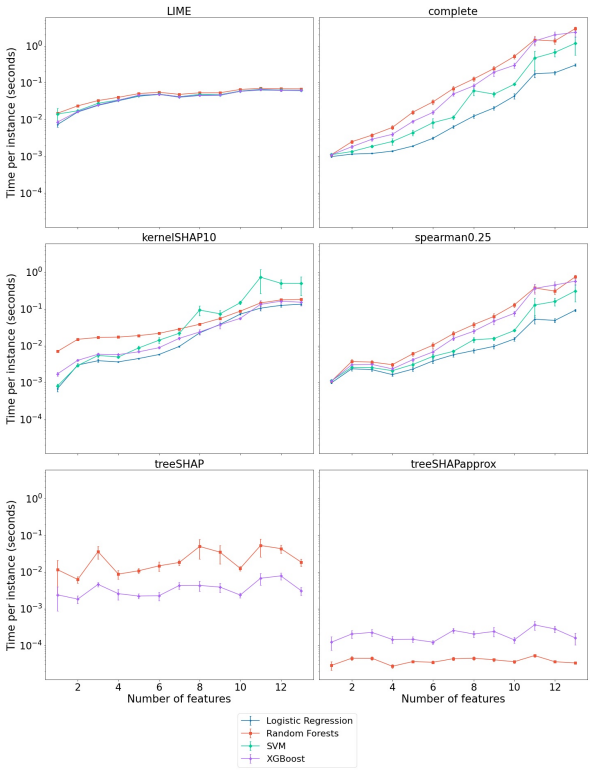

Figure 5: Execution time of each model per instance, averaged by number of features, for each method

of method's computation time in regards to the model used, except for *TreeSHAPapprox* where Random Forests are faster to compute. This may be related to the fact that *TreeSHAPapprox* only considers tree structures, as Random Forests tree structures are simpler than XGBoost's. In general, the faster a model is to train and predict values and the simpler it is, the faster the explanations are to compute, no matter the method,

We present in Figure 6 the mean absolute difference between each method applied to each model and the *Complete* method applied to each model. The figure does not present the results for *TreeSHAPapprox* because the only relevant model for this method is Random Forests, there is no other model to compare the results with.

For the three model-agnostic methods (*LIME*, *KernelSHAP* and *Spearman*), the Logistic Regression and SVM models generate the most precise explanations compared to the *Complete* method on the same models. We can see that the explanations based on Logistic Regression are usually more precise than SVM's, especially in low dimensions. XGBoost explanations are less precise than Random Forest's, except for the *Spearman* method (similar results observed). Overall, it seems that the simpler the model, the more precise it is in regards to the *Complete* method.

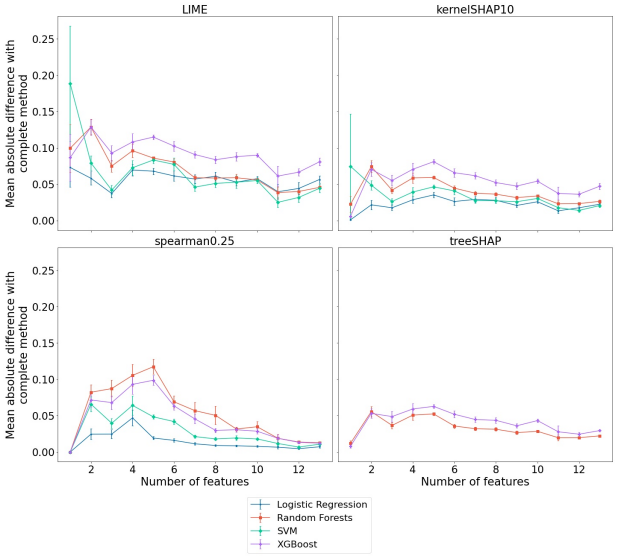

Figure 6: Mean absolute difference of each method with the *Complete*, averaged by number of features, for each model

Finally, regarding the AUC, we present all the results in Figure 7. We observe that for *LIME* and *KernelSHAP*, there is no significant difference between the AUC of the model's explanations. However, for the coalitional-based methods, we can see a clear separation between tree-based methods and non tree-based methods: the latter have higher AUC than the others. This means that, when using coalitional-based methods, one should be aware that different models may yield a different importance distribution over the features. For the tree-specific methods, we can see that XGBoost generates explanations with slightly higher AUCs than Random Forests on average.

4.5 Example on a medical dataset

Amongst the OpenML datasets previously studied, we choose a medical dataset, SA-Heart, to compare the explanations given by the different additive methods on an example. This way, we aim to both illustrate and validate the conclusions of the previous sections regarding explanation methods characteristics. We also aim to highlight practical differences that we can see on the influences of different methods for the same model and dataset.

SA-Heart is a dataset extracted from a larger database of South-Africans detailed in a 1983 study [20]. The extracted dataset is a

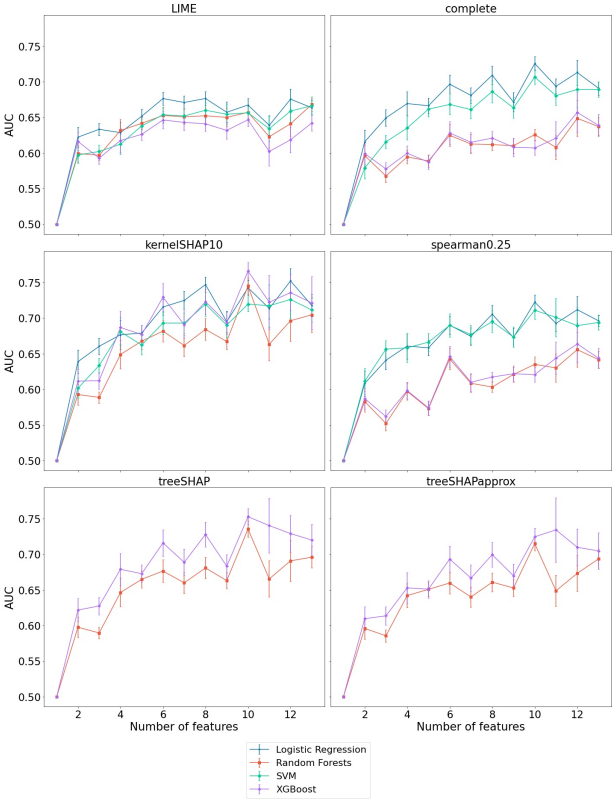

Figure 7: AUC of each model, averaged by number of features, for each method

retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. The dataset is composed of 462 individuals for 10 features. The main objective is to predict the binary target feature '**chd**', a coronary heart disease, according to 9 explanatory factors: **tobacco** (cumulative consumption tobacco), **age** (at the onset), **ldl** (low density lipoprotein cholesterol), **adiposity** (estimation of the body fat percentage), **obesity** (through the body mass index), **family** (family history of heart disease, present or absent), **alcohol** (current alcohol consumption), **sbp** (systolic blood pressure) and **type-A** (Type-A behavior scale). After model training, the different explanatory profiles obtained between the different methods of explanation are compared. By considering a reflection on the end-user side, the health care practitioners, explanatory profiles should be used 1) at the population level (global explanations), for example to highlight high-risk patient profiles, develop new prevention programs, develop new physio-pathological hypotheses but also 2) at the instance level (local explanations), for personalized medicine.

For conciseness in this paper, we limit the analysis to a single machine learning model. We choose Random Forests, as every explanation method that we benchmark is applicable to it. We present the results with SVM, Logistic Regression and XGBoost models in supplementary data.

To compare the explanations of the different additive methods, we look at global explanations given by each method. We use *SHAP*-like representations to visualize global explanations by aggregating local explanations on the same representation. This way, we build different figures. The first one, in Figure 8, represents a global explanation of the predictive model, given by each explanation method, by plotting the explanation profile of

each feature on a separated line. For each method, the features are sorted in decreasing feature importance, the top one being the most contributing feature on average, while the bottom one being the least contributing feature on average. For each feature, each dot represents an individual from the dataset, its color representing the value of the associated feature. Its position on the x-axis represents the contribution of the feature to the prediction of this individual, and overlapping dots are jittered on the y-axis.

We can see that most of the features have similar ranking among the different methods: tobacco and age are the two most important features except for the *Spearman* method which ranks age 5^{th}. On the opposite side, alcohol, spb, and type-A are always in the 4 least important features. These features have also similar explanation profiles. Conversely, some other features exhibit more marked difference depending on the methods. The most important difference is observed on the binary feature family history of heart disease. This feature is assigned fairly low importance by the coalitional-based method, relatively high importance (3rd most important feature) by *SHAP* methods, and very high importance by *LIME* (most important feature). Obesity and adiposity have also different influences depending on the method: obesity is ranked second least contributing by *LIME* and *SHAP*, but more important by the coalitional-based methods. It is important to note that obesity and adiposity are highly correlated (Pearson's correlation r=0.72). We hypothesize that it may be the reason for such differences. Overall, the three *SHAP* methods give similar explanations and have almost identical ranking of the features. From a global perspective, we can also see that *SHAP* and *LIME* present a more homogeneous "gradient" of colors for the explanations, where coalitional-based methods present mixed up colors in the explanations. This means that *LIME* and *SHAP*'s explanations are more locally monotonic, in the sense that the influence value of a feature for an individual is more locally correlated to the value of the feature for *LIME* and *SHAP* than it is for coalitional-based methods.

The second visualization that we present are Partial Dependence Plots (PDP). PDPs focus on the relationship between a feature and the influence of this feature on the model's prediction by plotting each pair of feature value and influence value on a 2-dimensional axis. We compare the PDPs of several important features in Figure 9.

Looking at the PDPs for the **age** feature, we show that *LIME* seems to form clusters of points around specific cut-off age values. To a lesser extent, this phenomenon can also be seen on the other *SHAP* methods. Conversely, coalitional-based methods have similar PDPs, and do not seem to find such cut-offs. However, it seems to be a special behavior of the explanation at specific ages. For example, subjects around 50 years have a marked lower contribution of this feature to the prediction of the presence of coronary heart disease than people even slightly younger or older. This may hint at an over-fitting of the machine learning model that would not have been captured by the other explanation methods. The explanation of the tobacco feature also largely differs among explanation methods. Where all the methods agree on attributing a low value to non-smoking individuals, the evolution of the contribution varies with the quantity of tobacco. Once again, *LIME* and *SHAP* explanations seem to find a cut-off value for tobacco consumption, of around 7 and 9 respectively, while coalitional-based methods capture a non-monotonic, more complex relationship.

We also look at adiposity PDPs. Once again, the three *SHAP* explanations are close to each other. Interestingly, they capture a non-monotonic relationship between the feature and the outcome, giving people around 30% of adiposity a higher influence for this feature (in absolute value) than people close to this value. This relationship seems to be captured in a lesser extent by coalitional-based methods, but not captured at all by *LIME*. We also note that the *Complete* and *Spearman* influences are more scattered, which means that more variance exists amongst subjects of the same adiposity for these methods than for the others.

Lastly, looking at obesity PDPs, *LIME* and *SHAP* methods find a negative relationship between obesity and the chd prediction. This seems counter intuitive, as obesity is a strong known comorbidity factor of heart diseases. As previously mentioned, obesity and adiposity are strongly correlated (r=0.72), and it may be the reason for such observation. Furthermore, we have mentioned in section 3 that *SHAP* works under the hypothesis that features are independent, but with such correlation, it is very unlikely that obesity and adiposity are independent. To better understand the relationship between these two features, as found by the methods, we plot in Figure 10 the influence values of adiposity and obesity given by each method.

The *Complete* and *Spearman* methods seems to find a positive correlation between the influences of the two features: when an individual is assigned a high influence value for obesity, a high influence value for adiposity is usually assigned, and conversely. We can even distinguish two clusters of individuals: one for individuals that have a high influence value for both features, and one for individuals that have a low influence value for both features. Such pattern is not found by *LIME* or *SHAP*, thus confirming the lack of ability of these methods to consider dependent features.

On a more global scale, we see that *LIME* and *SHAP* produce explanations that are easier to read at a first glance compared to *Complete* and *Spearman* explanations. However, *LIME* and *SHAP* seem to capture different cut-offs and relationships, and it is hard to confirm such values without further biological knowledge. Coalitional-based methods seems to produce explanations that are harder to read on a global scale, but more precise at an individual level and able to take into account the dependencies between features. PDPs for all features are available at https://github.com/EmmanuelDoumard/local_explanation_comparative_study.

5 LESSONS-LEARNED FOR THE USE OF ADDITIVE LOCAL METHODS

Table 2 summarizes advantages and drawbacks of each method studied in this paper. Overall, we highlight the fact that coalitional-based methods should be better at producing precise local explanations while *SHAP* should be better at producing coherent and easily interpretable global explanations. It is also confirmed by the fact that *SHAP* tends to assign more importance to few features than other methods, producing global explanations that are easier to read, but potentially hiding other features contributions and inter-dependences. Technically, *KernelSHAP* gives access to hyper-parameters to balance between execution time and explanation precision, but they are less accessible than *Spearman*'s and *LIME*'s parameters. Indeed, without extensive *KernelSHAP* knowledge or documentation readout, users can easily miss on these parameters.

We use all the results presented in this paper to show a simplified roadmap in the form of a decision tree in Figure 11 with

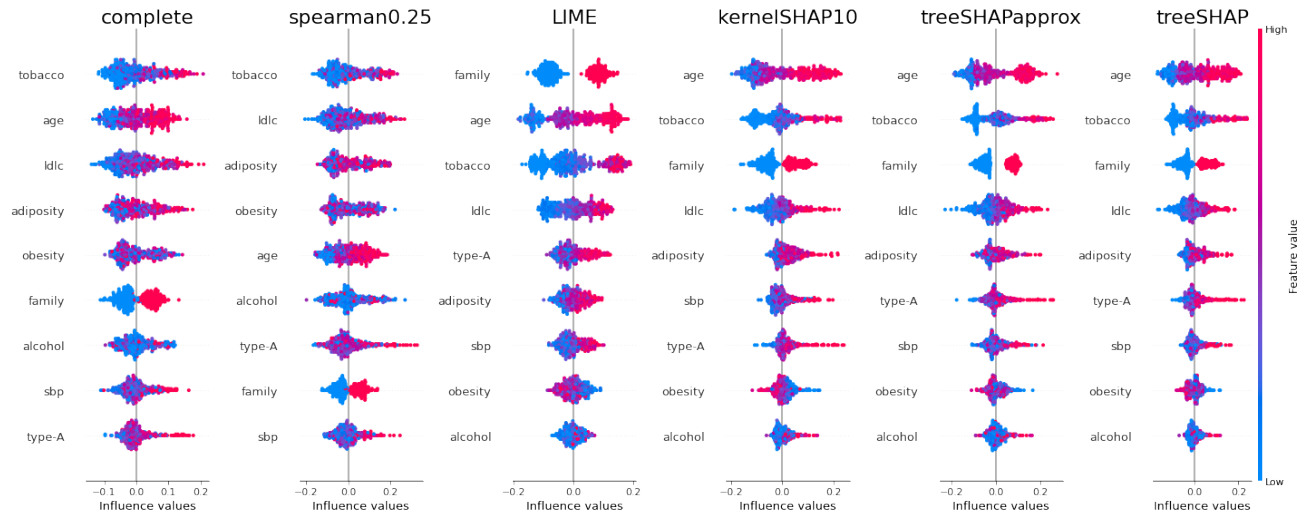

Figure 8: Summary plots of each method on the SA-Heart dataset

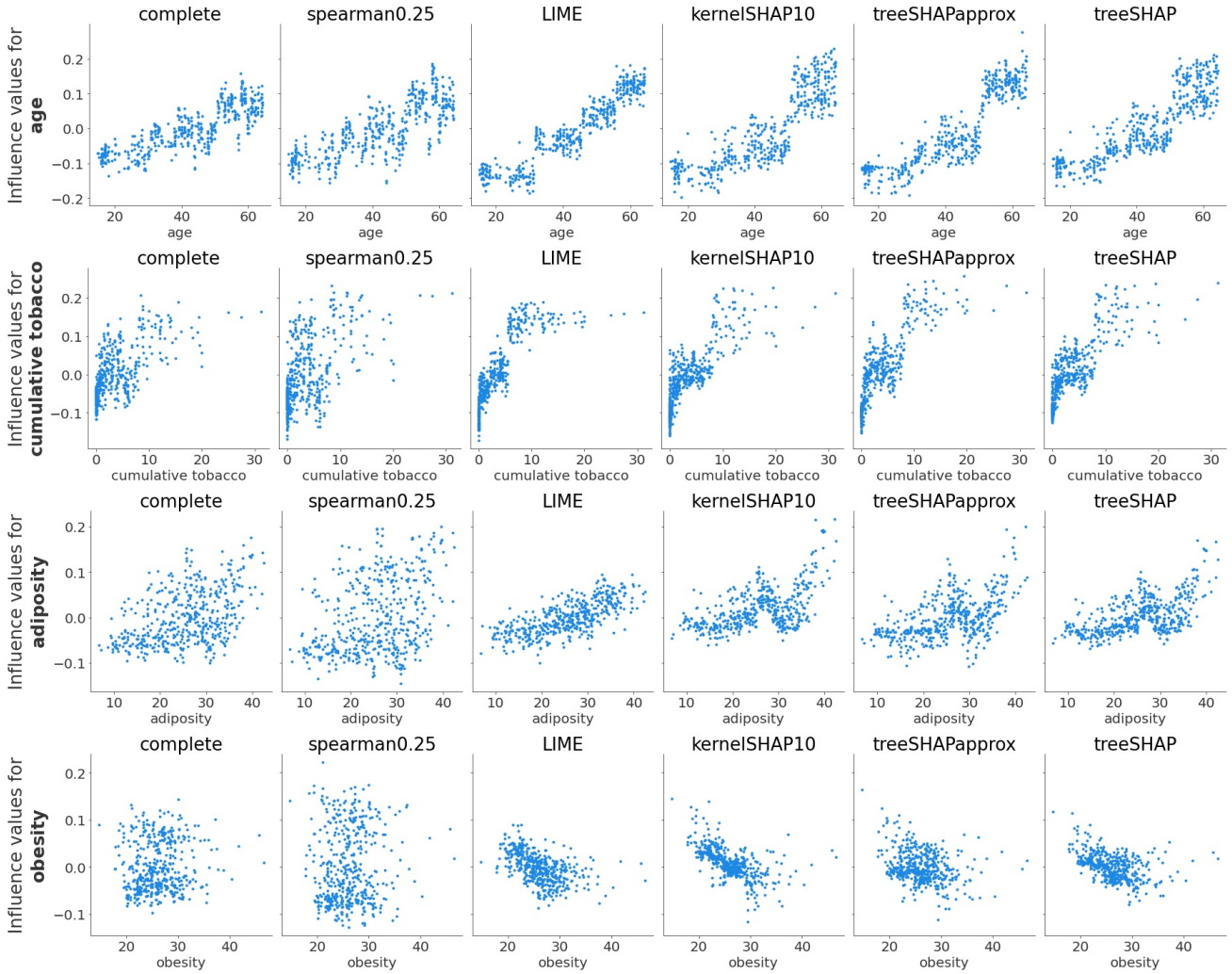

Figure 9: Partial dependence plots of age, tobacco, adiposity and obesity for each method

the intent to help readers finding the most suitable explanation method according to their datasets and objectives.

On this figure, high dimension represents the number of features present in the studied dataset. Indeed, there is no "hard"

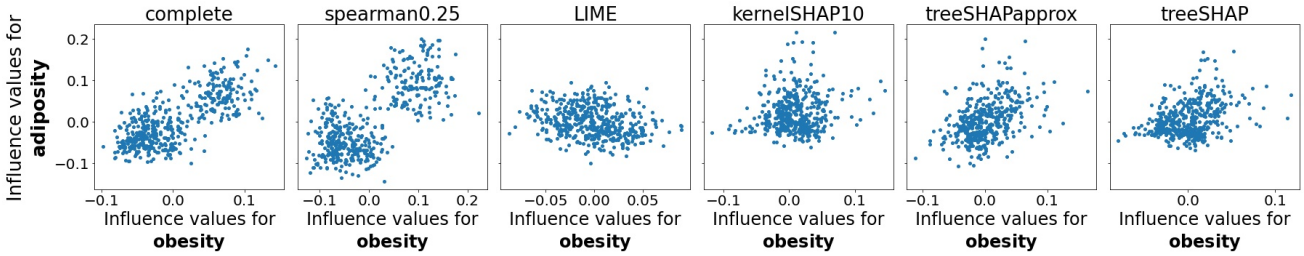

Figure 10: Influence value of adiposity against the influence value of obesity

Method name		Advantages		Drawbacks	
Coalitional based	Complete	Consider feature interdependence	Exact shapley values	Slow in high dimension Global explanations can be hard to read	
	Spearman		Parameter α to control the level of approximation		
LIME		Fast in high dimension Parameters to control approximation		Slow in low dimension Low quality explanations Tends to miss non linear and non monotonic influences	
SHAP	KernelSHAP	Easy to interpret global explanations		Approximations may be inprecise	Slow in high dimension
	TreeSHAP		Very fast in low and high dimensions		Tree-based models specific
	TreeSHAPapprox				

Table 2: Summary table of advantages and drawbacks of each method

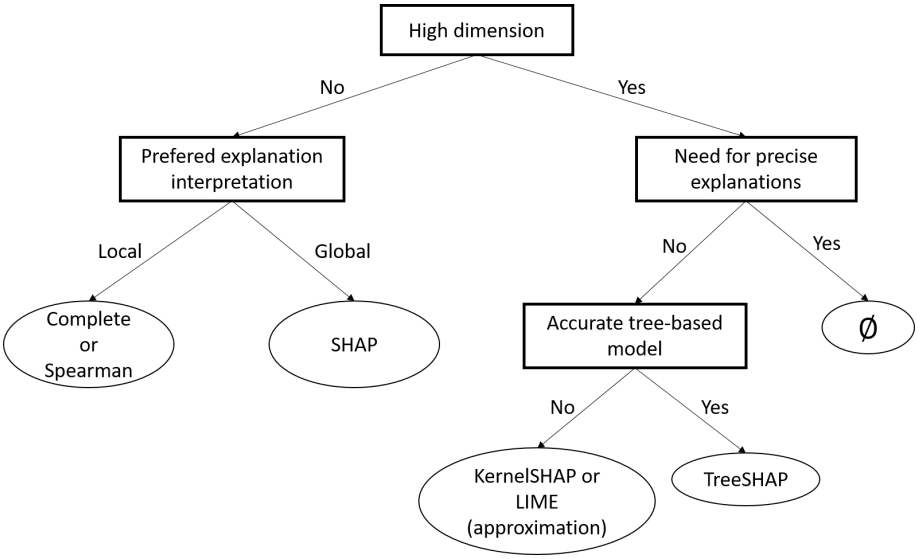

Figure 11: Roadmap for the most appropriate use of methods

cut-off to define when it goes from low to high dimension, but with our experiments, we can consider this cut-off somewhere between 11 and 15 features, depending on the dataset complexity and the user computational time and material available. "Accurate tree-based model" represents the ability of training a satisfactory (defined by the user's objectives) tree-based model on the dataset. The model can then be explained thanks to the optimization done in *TreeSHAP*. If the desired model is not tree-based, we advise the user to look at *KernelSHAP* and *LIME*'s parameters to reduce the number of background samples and perturbation samples respectively, until the explanations are computed in a reasonable time. However, we warn the user about the loss of precision induced by such method approximations.

Finally, we show that *SHAP* and *LIME* can make important approximations in some cases, and that coalitional-based methods cannot be executed in reasonable time in high dimension. This leaves an empty space for high dimension precise explanations that is not yet addressed to our knowledge.

6 CONCLUSION AND PERSPECTIVES

In this paper we performed a practical analysis of several local explainability methods for tabular data. Our findings indicate that there is not a single method that is the most appropriate for

every usage. Such usages include the need of a high precision for local explanations or on the contrary the need of explanations that can be aggregated to produce a better and clearer global understanding, while taking into account the complexity level of data especially concerning the high dimension case. Therefore, this thorough analysis allowed to identify strengths and limitations of each method along with practical recommendations on which method is most suitable for the use case of the user. The *Complete* is of course the most accurate but suffer for very long computational time. Nevertheless, *Coalitional* based methods allow an acceptable computational time while maintaining a strong precision of explanations. On the contrary, *LIME* and *SHAP* methods offer a more intelligible global view of feature effects. The greatest problem arises when high dimension (*i.e.,* high number of features) is involved, as it is often the case in statistics and Machine Learning. In this case, the exponential complexity of *Coalitional-based* methods make them too long to compute. Indeed, the worst case scenario is the need for high precision local explanations in high dimension since there is a clear lack of methods addressing this problem in the current literature. However, it is still possible to have local explanations with limited quality in high dimension, with the level of quality mostly depending on the time available for the user to generate such explanations. It is thus a very interesting future axis of work to benchmark the performances, in terms of precision of local explanations, of every local explainability method in a high dimension context under the constraint of a time limit. This would add value to our recommendations by filling out the 'high-precision in high-dimension' gap identified in our study. It would also be interesting to look into other machine learning models, especially deep neural networks which are more and more used. The very high complexity of this type of models hints at a different behavior for the explanation methods, but also an increase in computation time.

ACKNOWLEDGMENTS

This study has been partially supported through the grant EUR CARe N°ANR-18-EURE-0003 in the framework of the Programme des Investissements d'Avenir.

We also thank the French National Association for Research and Technology (ANRT) and Kaduceo company for providing us with PhD grants (no. 2020/0964).

REFERENCES

[1] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, and Vince I. Madai. 2020. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC Medical Informatics and Decision Making* 20 (Nov. 2020), 310. https://doi.org/10.1186/s12911-020-01332-6

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE* 10, 7 (2015). https://doi.org/10.1371/journal.pone.0130140 Publisher: Public Library of Science.

[3] Nadia Burkart and Marco F. Huber. 2021. A Survey on the Explainability of Supervised Machine Learning. *J. Artif. Int. Res.* 70 (May 2021), 245–317. https://doi.org/10.1613/jair.1.12228

[4] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *2016 IEEE Symposium on Security and Privacy (SP)*. 598–617. https://doi.org/10.1109/SP.2016.42

[5] William K Diprose, Nicholas Buist, Ning Hua, Quentin Thurier, George Shand, and Reece Robinson. 2020. Physician understanding, explainability, and trust in a hypothetical machine learning risk calculator. *Journal of the American Medical Informatics Association : JAMIA* 27, 4 (Feb. 2020), 592–600. https://doi.org/10.1093/jamia/ocz229

[6] Radwa El Shawi, Youssef Sherif, Mouaz Al-Mallah, and Sherif Sakr. 2019. Interpretability in HealthCare A Comparative Study of Local Machine Learning Interpretability Techniques. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. 275–280. https://doi.org/10.1109/CBMS.2019.00065

[7] Radwa ElShawi, Youssef Sherif, Mouaz Al-Mallah, and Sherif Sakr. 2020. Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. *Computational Intelligence* (2020).

[8] Gabriel Ferrettini, Elodie Escriva, Julien Aligon, Jean-Baptiste Excoffier, and Chantal Soulé-Dupuy. 2021. Coalitional Strategies for Efficient Individual Prediction Explanation. *Information Systems Frontiers* (2021). https://doi.org/10.1007/s10796-021-10141-9

[9] D. Garreau and U. von Luxburg. 2020. Explaining the Explainer: A First Theoretical Analysis of LIME. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) (Proceedings of Machine Learning Research, Vol. 108)*. PMLR, 1287–1296. http://proceedings.mlr.press/v108/garreau20a.html

[10] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*. PMLR, 5491–5500.

[11] Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. 2018. Defining Locality for Surrogates in Post-hoc Interpretablity. *Workshop on Human Interpretability for Machine Learning (WHI) - International Conference on Machine Learning (ICML)* (2018). https://hal.sorbonne-universite.fr/hal-01905924

[12] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable ai: A review of machine learning interpretability methods. *Entropy* 23, 1 (2021), 18.

[13] Stan Lipovetsky and Michael Conklin. 2001. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry* 17, 4 (2001), 319–330. https://doi.org/10.1002/asmb.446

[14] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018).

[15] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[16] Xin Man and Ernest P. Chan. 2021. The Best Way to Select Features? Comparing MDA, LIME, and SHAP. *The Journal of Financial Data Science* 3, 1 (2021), 127–139. https://doi.org/10.3905/jfds.2020.1.047 arXiv:https://jfds.pm-research.com/content/3/1/127.full.pdf

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

[18] Christoph Molnar. 2018. *A guide for making black box models explainable*. https://christophm.github.io/interpretable-ml-book/v

[19] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. 97–101. https://doi.org/10.18653/v1/N16-3020

[20] Rossouw, du Plessis, Benade, Jordaan, Kotze, Jooste, and Ferreira. 1983. Coronary risk factor screening in three rural communities-the CORIS baseline study. *South African medical journal* 64, 12 (1983), 430–436.

[21] Lloyd S Shapley. 2016. *17. A value for n-person games*. Princeton University Press.

[22] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning*. PMLR, 3145–3153. http://proceedings.mlr.press/v70/shrikumar17a.html

[23] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (2020).

[24] Erik Štrumbelj and Igor Kononenko. 2008. Towards a model independent method for explaining classification for individual instances. In *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 273–282.

[25] Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications Using Game Theory. *J. Mach. Learn. Res.* 11 (March 2010), 1–18. Publisher: JMLR.org.

[26] Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. 2021. On the tractability of SHAP explanations. In *Proceedings of AAAI*.

[27] Giulia Vilone and Luca Longo. 2021. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion* 76 (2021), 89–106. https://doi.org/10.1016/j.inffus.2021.05.009

[28] Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* 41, 3 (2014), 647–665. https://doi.org/10.1007/s10115-013-0679-x