Model-Agnostic Interpretability with Shapley Values

Andreas Messalas
Computer Engineering & Informatics Dept.
University of Patras
Patras, Greece
amessalas@ceid.upatras.gr

Yiannis Kanellopoulos
Code4Thought
Patras, Greece
yiannis@code4thought.eu

Christos Makris
Computer Engineering & Informatics Dept.
University of Patras
Patras, Greece
makri@ceid.upatras.gr

***Abstract*—The ability to explain in understandable terms, why a machine learning model makes a certain prediction is becoming immensely important, as it ensures trust and transparency in the decision process of the model. Complex models, such as ensemble or deep learning models, are hard to interpret. Various methods have been proposed that deal with this matter. Shapley values provide accurate explanations, as they assign each feature an importance value for a particular prediction. However, the exponential complexity of their calculation is dealt efficiently only in decision tree-based models. Another method is surrogate models, which emulate a black-box model's behavior and provide explanations effortlessly, since they are constructed to be interpretable. Surrogate models are model-agnostic, but they produce only approximate explanations, which cannot always be trusted. We propose a method that combines these two approaches, so that we can take advantage of the model-agnostic part of the surrogate models, as well as the explanatory power of the Shapley values. We introduce a new metric, *$Top_j Similarity$*, that measures the similitude of two given explanations, produced by Shapley values, in order to evaluate our work. Finally, we recommend ways on how this method could be improved further.**

***Index Terms*—Machine learning, interpretability, explanations, FATML, XAI, transparency, Shapley values, Surrogate**

I. Introduction

Interpretability [1] of machine learning models is a complex and evolving topic. Understanding why a model formed a certain prediction is crucial for achieving trust, fairness, accountability and transparency. Many machine learning algorithms, such as deep neural networks (DNNs), gradient-boosted methods and random forests, are treated as "black box" models because of their intricate structure. Their complexity may lead to high accuracy scores but also to low-interpretability, which is usually a fundamental trade-off in machine learning. Understanding the decisions of a model regardless of its accuracy is important, and in some cases critical (e.g., in medical diagnosis, when an action is depended on a model's prediction). Furthermore, people desire to know how automated decisions are being made for them. This is highlighted even more with the latest EU General Data Protection Regulation (GDPR), which gives users the right to ask for an explanation of an algorithmic decision concerning them [2]. Interpretability can discover bias (e.g, racial bias against black inmates [3]) and promote fairness, and has already become a necessary component in many machine learning systems, such as in banking (reason codes for loan disapproval) [4], insurance, healthcare [5] and many other industries. Moreover, intepretability offers accountability, transparency and can diagnose ill-conditioned systems, which otherwise would be considered accurate and precise. For example in [6], artificially produced images that make no sense to humans, are labeled by state-of-the-art DNNs as recognizable objects with high confidence (99.99%). Classifying a white-noise image as an animal may not seem that serious, but it becomes critical when people can use these vulnerabilities for malicious hacking (e.g., a terrorist tricking an airport's security scanning system or a self-driving car not recognising an altered stop sign [7]).

A common way to explain a prediction is through feature importance, which is to calculate the contribution of each feature to the prediction of the model. A basic categorization of interpretability methods is whether the model to be explained is known or not. A *model-agnostic* method does not require any knowledge of the inner workings of the black box model, only access to the data and the predictions of the model is necessary, in contrast to *model specific* methods, which are applicable only for a single type of algorithm. Another distinction is whether explaining a single instance, consisting *local interpretability*, or explaining the whole model in a holistic view, leading to *global interpretability* [8].

A well-known interpretability method is Shapley explanations [9], where the features of a machine learning problem are treated as players in a coalitional game from Game Theory. A specific value called Shapley value, is assigned to each feature and demonstrates its contribution to the result. While Shapley values produce high quality explanations, their exact computation can be implemented efficiently only in decision tree-based models, using the *Tree SHAP* algorithm from [10], [11].

Another straightforward and intuitive approach is the creation of surrogate models, which are interpretable models trained to approximate the predictions of a black box model. Explanations are derived effortlessly from the surrogate, since it is chosen to be interpretable, in a model-agnostic way. The fidelity (distance between the black box's and the surrogate model's predictions) and the interpretability (ease at producing explanations) of the surrogate model are hard to be satisfied

978-1-7281-4959-2/19/$31.00 ©2019 European Union

simultaneously. In the general case, high-fidelity leads to low-interpretability and vice versa. For this reason, the surrogate models are usually interpretable models but with low-fidelity, so their explanations are not accurate [11].

Our method combines these two approaches, by creating an *XGBoost* [12] surrogate model and then extracting the Shapley explanations. In our case, the complexity of the surrogate model is irrelevant, since the explanations will be derived by *Tree SHAP*. Therefore, we can achieve high-fidelity and high-interpretability, by exploiting the advantages of the two methods individually. Our method is model-agnostic, since it requires only access to the data and the predictions of the original model.

The correctness of this method relies on the similarity between the explanations of the original model and the surrogate model. To evaluate this, we introduce a new metric called *Top$_j$Similarity*, which calculates the distance of the two explanations. The results of our experiments show that we can achieve high similarity in some cases, but in the general case this depends on the structure of the dataset. In future work, we propose with optimism that a different approach on building the surrogate model could lead to high similarity, regardless of the structure of the dataset.

II. Background Work

A. Shapley Value method - Tree SHAP

SHAP (SHapley Additive exPlanations) [10] is a unified framework for interpreting predictions and it is based on the Shapley regression values [13] from cooperative game theory. SHAP assigns each feature an importance value for a particular prediction to compute the explanation. This value is the unified measure of additive feature attributions and is called *SHAP value*, $\phi_i \in \mathbb{R}$. The formula for ϕ_i is:

$$\phi_i = \sum_{S \in F \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{1}$$

where F is the set of input features, S is a subset of input features and $M = |F|$ is the number of input features. This formula computes the gravity of each feature by calculating its importance when it is present in the prediction and then subtracting it when it is not present.

- $f_{S\cup\{i\}}(x_{S\cup\{i\}})$: is the output when the i^{th} feature is present
- $f_S(x_S)$: is the output when the i^{th} feature is withheld
- $\sum_{S\in F\setminus\{i\}} \frac{|S|!(M-|S|-1)!}{M!}$: is the weighted average of all possible subsets of S in F

The SHAP value is the only explanation method with a solid theory (since 1950) and being the only possible locally accurate and consistent feature contribution values [9], they can produce high-quality explanations (both local and global).

There are various approaches of approximating the SHAP values: model-agnostic (*Shapley sampling values* and *Kernel SHAP*) and model-specific (*Max SHAP, Deep SHAP*). However, the most novel method is *Tree SHAP*, which implements an exact computation of Shapley explanations (SHAP values), which works by leveraging decision trees structures to disaggregate the contribution of each input in a decision tree or decision tree ensemble model. Given a number of trees T, L being the maximum number of leaves in any tree and M being the number of features, the complexity of equation 1 is $O(TL2^M)$. In [10] for balanced trees, where the depth becomes $D = logL$, the *Tree SHAP* algorithm has complexity $O(TLD^2)$, which reduces the computational complexity from exponential to low-order polynomial for trees and sums of trees. The other methods are slower approximate methods.

The *Tree SHAP* method takes as input a trained model (currently works only for XGBoost [12], LightGBM [14], CATBoost [15] or Scikit-learn's [16] decision tree models) alongside with input data X ($N \times M$ matrix of N instances and M features) and produces an $N \times M$ matrix with the SHAP values. Each value represents the impact (positive or negative) of the feature to the corresponding instance. Local interpretability can be accomplished by extracting those features for a specific instance with the highest absolute SHAP value and demonstrating the amount of their positive or negative influence. Global interpretability can be achieved by aggregating the SHAP values across the instances. The Python package of *Tree SHAP* [10] provides tools that implement graphs of local and global explanations, as well as dependency plots and interaction value dependency plots. Nonetheless, custom explanation methods can be constructed that fit the needs of a given task.

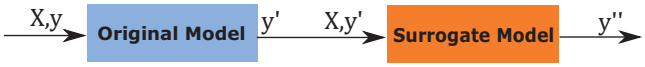

Fig. 1: Overview of the surrogate model method

B. Surrogate Model method

A surrogate is an interpretable model that is used to explain a complex black box model. It is created by training usually a much simpler model (e.g, shallow-depth decision tree) with original input data X and predictions y' of the original model ("Fig. 1"). Fidelity can be measured through the R^2 metric between the predictions of the surrogate model and the original model.

The produced model can be viewed as an approximated flow chart description of the original model. Surrogate models are model-agnostic and enable some primary deductions about the most important features and interactions of the complex model, especially when combined with Partial Dependence (PD) [17], [18] and Individual Conditional Expectation (ICE) charts [19].

However, the simplicity required to make the model explainable contrasts the need of fidelity. Fidelity and interpretability share a disproportional relationship, so this method has certain limitations. To ensure good fidelity, a more complex model is necessary. Nevertheless, a complex model is difficult to be explained. Our goal is to overcome this trade-off and achieve high-fidelity as well as high-interpretability.

The process of creating a surrogate model is also known as *model extraction.* In [20] the *TREPAN* algorithm and in [21] the *DeepRed* algorithm extract a decision tree from trained neural networks. Similar work is found in [22] and [23], where deep neural networks are reverse-engineered. In [24], although the goal here is not interpretability, it is demonstrated that the models of ML-as-a-service systems (e.g., BigMl, Amazon) can be replicated with high-fidelity. In a more recent work, the model-agnostic method described in [25] induces a decision tree from a black box model by actively sampling new training points to avoid overfitting and is used for interpretability purposes. Finally, similar work can also be found in engineering, where a surrogate model can replace a complex and costly simulation model (e.g., shape for an aircraft wing and airflow around it).

C. Related Work

A well-known interpretability method is LIME (Local Interpretable Model-Agnostic Explanations) [26]. LIME is model-agnostic and works by perturbing the original data and then observing how this affects the predictions. Perturbation for example can be extracting words from text or hiding parts from an image (i.e., creating 'superpixels'). The processed data then are fed to an interpretable model thus generating an explanation by approximating the original model with a simpler one. Another aspect of LIME is that it is implemented *locally*, in the neighbourhood of the prediction to be explained. In more recent work, the creators of LIME have released a novel approach to LIME, *anchors* [27], which generate high-precision sets of plain-language rules to describe a machine learning model's prediction in terms of the model's input variable values. A drawback of LIME is that it can be difficult to deploy, as its locality requires multiple implementations in order to give a highly interpretable explanation. *K-LIME* [28] is a modification of LIME, where the local regions are constructed by K clusters or user-defined segments instead of simulating perturbed data.

Supplementary method is Partial Dependence (PD) plots [17], [18], which show the interactions between the target and a feature and their effect on a prediction. Usually, they are used together with Individual Conditional Expectation (ICE) charts [19], which depict how an instance's prediction changes when a single feature changes. Moreover, Accumulated Local Effects (ALE) plots [29], demonstrate the influence of the features to the prediction on average and are an unbiased alternative to PD plots, which output misleading results when the predictors are dependent. Finally, Leave-one-covariate-out (LOCO) variable importance [30] creates local interpretations for each row of data, but suffers also from inaccuracy when nonlinear dependencies exist in a model. For this reason, Shapley explanations is a better alternative.

There are already plenty of communities with different notions on this subject. FATML [31] is a group of academics that is concerned with fairness, accountability, and transparency in machine learning. Another prominent community is a group of researchers funded by Defense Advanced Research Projects Agency (DARPA) [32], a division of the American Defense Department that investigates new technologies. They label their work as Explainable Artificial Intelligence XAI).

Fig. 2: Overview of our method

There are also commercial packages that provide interpretability techniques. *Driverless AI* [28] produced by the company H2O.ai, employs different explanatory methods (K-LIME, Shapley, Decision Tree Surrogate, PD, ICE and more). IBM's *Watson OpenScale* [33] platform also provides services that deal with bias in models. Here, the inner-workings of their explanatory methods are not available.

III. Our Method: Surrogate Model + Tree SHAP

An overview of our method is shown in "Fig. 2". Firstly, an XGBoost model is trained on input data X and the predictions y' of the black-box model. Then, the produced model alongside with the input data X is given to the *Tree SHAP* method, which produces a matrix with the SHAP values. The XGBoost package is chosen, as it is applicable to the *Tree SHAP* method, which also has a fast-approximate implementation specifically for XGBoost models. Our aim is obtaining high-interpretability and high-fidelity simultaneously. High-interpretability is accomplished just by using Shapley explanations. For high-fidelity, our approach is overfitting the data into the surrogate model. The main idea is that by overfitting the data, we obtain almost perfect accuracy, which means that we can explain every instance of the dataset. The trained surrogate model of our method makes the same (99.99%) predictions as the original model. Therefore, the produced SHAP values of the next step will have actual meaning, since if the prediction of the surrogate model was wrong (i.e., not the same as the original model's prediction), then the explanation would be pointless.

This method is model-agnostic as it requires as input only the dataset X and the predictions of the black-box model y'. The "King County Housing, USA" [34] dataset contains structured data about houses in King County, Washington and the prediction target is the housing price. "Fig. 3, 4" demonstrate the global and local explanations derived by our method. The original model was a k-Nearest-Neighbors model, which cannot be explained directly by the Tree SHAP method. In "Fig. 3" we can see that the size of a house's living room (*sqft_living*) is the most important factor for predicting the pricing of the house. The higher the SHAP value of a feature, the higher its contributions is to the deducted decision. Every house in the dataset is run through the model and a dot is created for each individual SHAP value. Dots are colored by the feature's value of that house and pile up vertically to show density. "Fig. 4" shows the most important feature for the least expensive house (75.000$) is *sqft_living*=670, followed by *yr_built*=1966 (year the house was constructed),

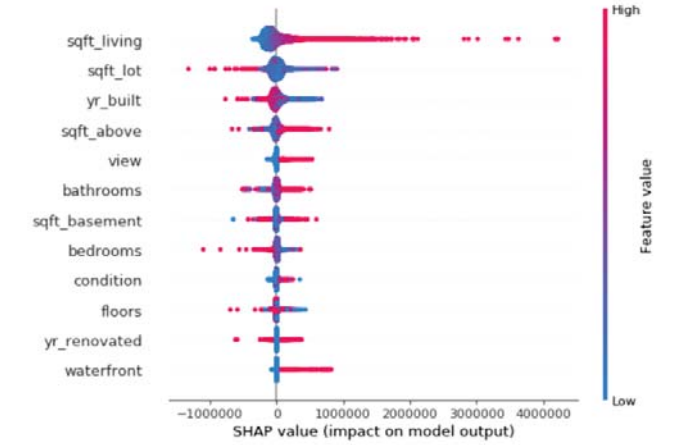

Fig. 3: Global Interpretation: Summary plot

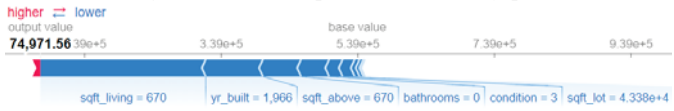

Fig. 4: Local Interpretation: Force plot

sqft_above=670 (total square footage minus square footage of the basement) and *bathrooms*=0 (number of bathrooms).

Attempts were made also on unstructured data (MNIST dataset). Although explanations were successfully extracted using our method, more experiments are needed in order to test the scalability of our process.

IV. Evaluation

A. Metric: Top$_j$Similarity

The fact that the surrogate model takes almost the same decisions as the original model does not necessary imply that the decision process of the two models is the same. By decision process, we mean the features that the model relied on in order to make a prediction. In a decision tree, a decision process is the path from the root of the tree to a leaf containing the features in each node. For example, in the "King County Housing" dataset a decision tree-based model may predict a price of 100.000$ and its surrogate predicts 100.001$. However, the original model might have used a decision path different than the surrogate's path, although the predicted price is almost the same. Therefore, the explanation by the surrogate model will be not fully trustworthy. The decision process is more difficult to pinpoint in more complex models, such as Deep Neural Networks, but the problem remains the same. This highlights the fact that a high R^2 score does not imply high fidelity, since it only guarantees fidelity in the predictions, but not in the internal decision processes of the original and surrogate model. We could refer to the first kind of fidelity as *external fidelity* and the second one as *internal fidelity*.

This uncertainty needs to be cleared, since it questions the trust on the explanations of our model. In our knowledge, the only measure of fidelity in the bibliography of surrogate models, regards only to the *external fidelity*, with metrics like

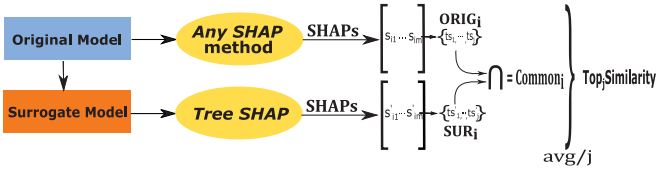

Fig. 5: Overview of the method calculating the Top$_j$Similarity metric

R^2. The insufficiency of such like metrics led as to introduce a new metric called Top$_j$Similarity, which measures the internal fidelity between the original and the surrogate model. The main idea ("Fig. 5") is to train as original models, models that *Tree SHAP* is applicable (or any model that the SHAP values can be extracted) and derive their SHAP values. Then we calculate the surrogate model's SHAP values and we compare them with the original SHAP values. The comparison is based on the order of the most important features. We order by absolute value the SHAP values from the two models for each instance and we place the j top features in sets $ORIG_j$ and SUR_j. Array $common$ is created as follows: $common_i = ORIG_j(i) \cap SUR_j(i), \forall i \in N$, where N is the number of instances. Finally, the formula for the proposed metric is:

$$Top_jSimilarity = \frac{avg(common)}{j} \quad (2)$$

For instance, Top$_1$Similarity = 80% means that the two models agree on the most important feature for the 80% of the instances. The range of j depends on the distribution of the SHAP values. Usually for a dataset of 10-20 features, the Top-5 features are the most contributive. "Fig. 6" shows the distribution of the SHAP values, after each row of the Shapley matrix has been ordered by their absolute value and then the average of each column was calculated (from the standard UCI Adult Income [35] dataset on a LightGBM model). The appropriate range of j can be decided by examining this diagram.

B. Experiments - Results

To evaluate out method we experimented on 4 datasets described in "TABLE I".

The format of the process is:

1) Train original model, calculate metrics (R^2, RMSE, MAE) and get SHAP values
2) Train surrogate model, calculate metrics (R^2, RMSE, MAE) and get SHAP values
3) Find Top$_j$Similarity for $j \in [1, 8]$

CATBoost, LightGBM, XGBoost and Scikit-learn's decision tree models were used as original models and XGBoost as surrogate. Moreover, for some experiments the original models were trained more than once and were tuned appropriately in order to get a different R^2 metric each time. For example, in the House Sales in King County, USA dataset the original model using Scikit's-learn decision tree was trained three times with an R^2 value each time of 75%, 81% and 93%. This was done in order to increase diversity in the original model

Experiment	#instances	#features	Target
UCI Adult income	32561	12	predict probability of an individual making over $50K a year in annual income
House Sales in King County, USA	21613	12	predict the price of a house in King County, USA
OpenML Elevators [36]	16599	16	predict an action taken on the elevators of a F16 aircraft
Default of Credit Card Clients [37]	30000	23	predict if credit card customer will default on credit card bill

TABLE I: Overview of the datasets used in the experiments

Experiment Setup	Train/Test split	Whole dataset	K-Fold(1st)	K-Fold(2nd)	K-Fold(3rd)	K-Fold(4th)	K-Fold(5th)
Top$_1$Similarity	74.85	79.44	77.43	79.58	80.27	79.99	80.61

TABLE II: Different Top$_1$Similarities by surrogate models trained on different parts of the dataset

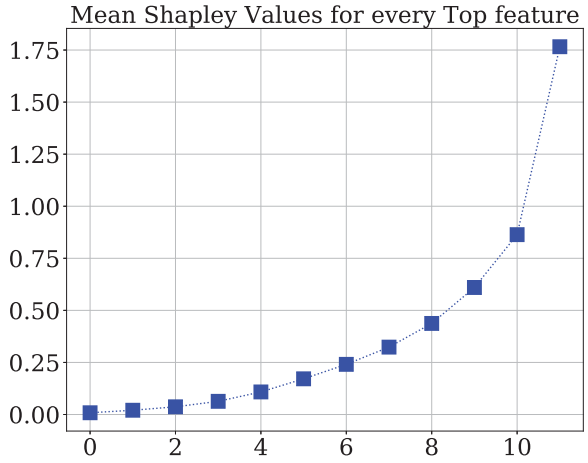

Fig. 6: Mean SHAP value for each column of the absolute ordered Shapley matrix

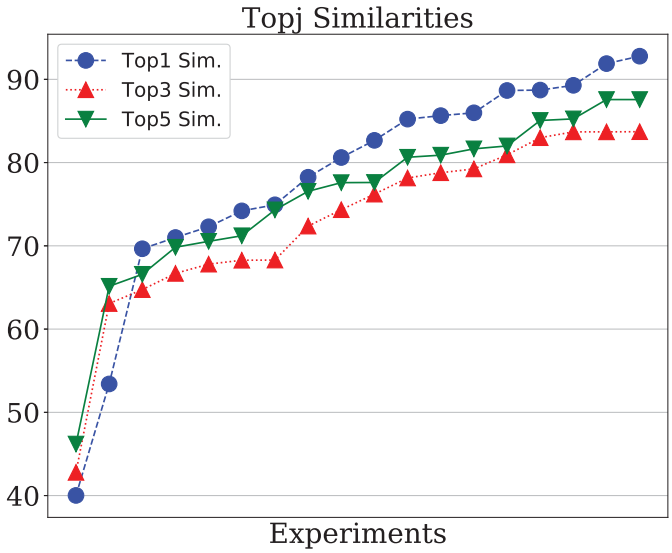

Fig. 7: Top$_1$, Top$_3$ and Top$_5$ Similarities for all 18 experiments

selection. Altogether 18 experiments were performed using these four datasets with different original models and different initial tunings.

The Similarity metric can be viewed as the percentage of confidence in the explanations of the surrogate model. Top$_5$Similarity = 80% means that the explanations of the surrogate model are on average 80% accurate. The boundary line between a good and a bad similarity is rather subjective, but intuitively we can say that for a model containing more or less 20 features having Top$_j$Similarity values for $j \in [1, 5]$ of more than 80% is acceptable.

The results in "Fig. 7" demonstrate some encouraging Similarities, but there is no consistency across all experiments. Similarity was treated as accuracy in a typical supervised learning problem, in a sense that the tuning of the surrogate model was adjusted each time to achieve a higher Similarity value. The most negative aspect of our method is that we have not find a way to know a priori, if the trained surrogate model will have a good Similarity or not. The R^2, RMSE and MAE metrics are not linked with Similarity, so its quality is determined only after it has been calculated. We used overfitting in our advantage, as it enables the explanation of every instance, but it turns out that it infringes the similitude between the decision process of the original and the surrogate model. Other approaches were implemented to decrease the effect of overfitting but still achieving high-fidelity ($R^2 > 99\%$). One method was using train/test splits. Another was adding artificial data created by the *Imbalanced-learn* [38] library (with oversampling, undersampling and a combination of these two methods) to deal with unbalanced datasets. Finally, the k-fold cross-validation technique was used, by implementing our method on the data of each random fold. The Similarity of each fold was different and in some cases was much better than all the other approaches overall. This is maybe an insight that a thorough selection of input data for the surrogate model may lead to a better Similarity consistently. "TABLE II" demonstrates this effect: a LightGBM model was trained as an original model on the "UCI Adult income" dataset and 3 methods were implemented to produce the surrogate model, using train/test split, overfitting model on the whole dataset and training the model on 5 random folds, produced by the

K-Fold method. We can see that the last fold of the K-Fold method, produced the best Top_1Similarity. Overall, the results from all experiments were produced either by overfitting the whole dataset on the surrogate or by one of the previous alternative methods.

Lastly, the complexity of our method depends mainly only on the value of the parameter *max-depth* of the ensemble trees by the XGBoost model, since the Tree SHAP complexity is $O(TLD^2)$ and the training of the surrogate with overfitting is relatively fast. Usually a value between 2 and 9 is enough to produce high-fidelity surrogate models. Taking into consideration that *Tree SHAP* has a fast-approximate implementation specifically for XGBoost models, the overall process is computationally fast.

V. Conclusions and Future Work

A strong aspect of our method is that it is model-agnostic. It requires only access to the data and the prediction of the original function, without knowing any details about the predictive model. We showed that it can produce high quality explanations, using the the SHAP values in an efficient way. The experiments did not provide a consistent conclusion about the similarity between the explanations of the original and the surrogate model. However, they gave an insight on how this problem could be fixed. Creating a custom surrogate model as described in [25] and combining with techniques from [20], [21], [23], [24] could possibly lead to a surrogate that is simultaneously highly interpretable and an accurate replica of the original. Realizing that the complexity and the interpretability of the surrogate model is irrelevant, as the explanations will be derived by the SHAP values, can solve the current problem that the surrogate model approaches have, which is this trade-off between interpretability and fidelity.

Moreover, we have introduced a new metric, *Top$_j$Similarity*, that we believe is indicative measure of the distance between explanations. We argue that the R^2 metric is not sufficient in capturing the "true" fidelity of a surrogate model, thus making *Top$_j$Similarity* a better measure of fidelity. This metric can also be used to test the similitude of tree-based models (or any model that its SHAP values can be extracted). An alternative to calculating multiple *Top$_j$Similarities* could be combining them in one linear relationship, with weights derived by the diagram in "Fig. 5".

Future work will include the construction of a custom surrogate tree-based model, with focus on fidelity, as well as a modification of the *Tree SHAP* algorithm in order to be applicable to the new custom tree model. We are optimistic that this proposed model-agnostic method can be a powerful and flexible approach in this immensely developing area of interpretability of machine learning models.

References

[1] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[2] B. Goodman and S. R. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI Magazine*, vol. 38, no. 3, pp. 50–57, 2017. [Online]. Available: https://www.aaai.org/ojs/index.php/aimagazine/article/view/2741

[3] C. Wadsworth, F. Vera, and C. Piech, "Achieving fairness through adversarial learning: an application to recidivism prediction," *CoRR*, vol. abs/1807.00199, 2018. [Online]. Available: http://arxiv.org/abs/1807.00199

[4] T. Ogunyale, D. G. Bryant, and A. Howard, "Does removing stereotype priming remove bias? A pilot human-robot interaction study," *CoRR*, vol. abs/1807.00948, 2018. [Online]. Available: http://arxiv.org/abs/1807.00948

[5] M. A. Ahmad, A. Teredesai, and C. Eckert, "Interpretable machine learning in healthcare," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, June 2018, pp. 447–447.

[6] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 427–436. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298640

[7] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, 2017. [Online]. Available: http://arxiv.org/abs/1707.08945

[8] P. Hall and N. Gill, *An Introduction to Machine Learning Interpretability*, 1st ed. O'Reilly Media, Inc., 2018.

[9] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *CoRR*, vol. abs/1705.07874, 2017. [Online]. Available: http://arxiv.org/abs/1705.07874

[10] S. M. Lundberg, G. G. Erion, and S. Lee, "Consistent individualized feature attribution for tree ensembles," *CoRR*, vol. abs/1802.03888, 2018. [Online]. Available: http://arxiv.org/abs/1802.03888

[11] P. Hall, "On the art and science of machine learning explanations," *CoRR*, vol. abs/1810.02909, 2018.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[13] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton: Princeton University Press, 1953, pp. 307–317.

[14] D. Wang, Y. Zhang, and Y. Zhao, "Lightgbm: An effective mirna classification method in breast cancer patients," in *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics*, ser. ICCBB 2017. New York, NY, USA: ACM, 2017, pp. 7–11. [Online]. Available: http://doi.acm.org/10.1145/3155077.3155079

[15] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *CoRR*, vol. abs/1810.11363, 2017.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078195

[17] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[18] Q. Zhao and T. J. Hastie, "Causal interpretations of black-box models," 2017.

[19] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015.

[20] M. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, 1995, pp. 24–30. [Online]. Available: http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks

[21] J. R. Zilke, E. Loza Menc'ia, and F. Janssen, "Deepred - rule extraction from deep neural networks," in *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*, 2016, pp. 457–473. [Online]. Available: https://doi.org/10.1007/978-3-319-46307-0_29

[22] S. J. Oh, M. Augustin, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," *international conference on learning representations*, 2018.

[23] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 506–519.

[24] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, 2016, pp. 601–618. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer

[25] O. Bastani, C. Kim, and H. Bastani, "Interpreting blackbox models via model extraction," *CoRR*, vol. abs/1705.08504, 2017. [Online]. Available: http://arxiv.org/abs/1705.08504

[26] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[27] M. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 1527–1535. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982

[28] P. Hall, G. Navdeep, and W. Phen, "Machine learning interpretability with h2o driverless ai," H2O, February 2019. [Online]. Available: http://docs.h2o.ai

[29] D. W. Apley, "Visualizing the effects of predictor variables in black box supervised learning models," *arXiv preprint arXiv:1612.08468*, 2016.

[30] J. Lei, M. GSell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1094–1111, 2018.

[31] "Fairness, accountability, and transparency in machine learning." [Online]. Available: http://www.fatml.org/

[32] "Explainable artificial intelligence (xai)." [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence

[33] "Watson openscale." [Online]. Available: https://www.ibm.com/cloud/watson-openscale

[34] Kaggle.com, "House sales in king county, usa," 2017. [Online]. Available: https://www.kaggle.com/harlfoxem/housesalesprediction

[35] R. Kohavi and B. Becker, "UCI adult data set," 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult

[36] "OpenML elevators," 2014. [Online]. Available: https://www.openml.org/d/846

[37] Y. I-Cheng, "Default of credit card clients data set," 2016. [Online]. Available: https://www.openml.org/d/846

[38] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3122009.3122026