Evaluating Attribution Methods in Machine Learning Interpretability

Qudrat E Alahy Ratul
Computer Science Dept.
Boise State University
Boise, Idaho, United States
qudratealahyratu@u.boisestate.edu

Edoardo Serra
Computer Science Dept.
Boise State University
Boise, Idaho, United States
edoardoserra@boisestate.edu

Alfredo Cuzzocrea
iDEA Lab
University of Calabria,
Rende, Calabria, Italy
alfredo.cuzzocrea@unical.it

***Abstract*—Interpretability is a key feature to broaden a conscious adoption of machine learning models in domains involving safety, security, and fairness. To achieve the interpretability of complex machine learning models, one approach consists in explaining the outcome of machine learning models through input features attribution. Attribution consists in scoring the features of an input instance by establishing how important is each feature value in a fixed instance to obtain a specific classification outcome from the machine learning model. In literature, several attribution methods are defined for specific machine learning models (e.g., neural networks) or more general ones that are model agnostic (i.e., can interpret any machine learning models). Attribution is particularly appreciated for its easy understanding of the interpretation, which is the attribution. In domains involving safety, security, and fairness, properties of the explanation such as precision and generality are crucial to establish human trust in machine learning interpretability and then on the machine learning model itself. However, even if precision and generality are clearly defined in rule-based interpretation models, they are not defined or measure on attribution models. In this work, we propose a general methodology to estimate the degree of precision and generality in attribution methods. In addition, we propose a way to measured consistency in attribution between two attribution methods. Our experiments focus on the two most popular model agnostic attribution methods, SHAP and LIME, and we evaluate them to two real applications in the field of attack detection. Our proposed methodology shows in these experiments that both SHAP and LIME lack precision, generality, and consistency and that still more investigation in the attribution research field is required.**

***Index Terms*—Machine Learning Interpretability, Attribution Methods, Evaluation Methodology.**

I. Introduction

Machine learning models are widely adopted for solving various problems. From the range of movie recommendation systems to personal voice assistants, or in highly regulated domains involving decisions of significant impact, such as mortgage approval models or healthcare decision support systems, the democratization of Artificial Intelligence (A.I.) in our society is undeniable [1]. Though the use of the ML model is expanding, most machine learning models' inner logic and mechanism are still hidden to the users and experts. These models are considered black boxes [2]. Relying on ML algorithms for sophisticated decision-making like aircraft collision detection systems without understanding the models can result in severe consequences [3]. Hence many interpretable models and explanation methods [2] are developed.

As the social impact of ML algorithms are becoming more significant day by day, the necessity of understanding the reason behind the decision-making process is also increasing [4]–[7].

A large amount of study has been done to address the issue. Explainable Artificial Intelligence (X.A.I.) is a field of study that aims to develop interpretable ML models and to make a shift of transparent A.I. [2]. This research field aims to develop a more explainable model and methods to explain existing black box models without compromising their predictive performance. A notable initiative in this research field is the Defense Advanced Research Projects Agency (DARPA), funded by the U.S. Department of Defense, which created the X.A.I. program for funding academic and military research [8]. Another example of government initiative is "Preparing for the Future of Artificial Intelligence"- a report published by the White House Office of Science and Technology Policy (OSTP), emphasized that A.I. systems should be open, transparent, and understandable so that people can interrogate the assumptions and decisions behind the models' decisions [9]. Outside the U.S.A., several countries have already taken the initiative for transparent A.I. . French Strategy for Artificial Intelligence, The United Kingdom's Academy of Sciences, Portugal government, has published their roadmap towards interpretable A.I. [10]–[12]. European Union stated that "A.I. systems should be developed in a manner which allows humans to understand (the basis of) their actions" to increase transparency and to minimize the risk of bias error [13].

The Tech industry has already started to practice interpretable A.I. in various A.I.-related fields. In addition, to invest in interpretable A.I. research, companies are focusing on interpretability for commercializing interpretable A.I. products. Google is advocating interpretability by including planning for interpretability, treating interpretability as a core part of the user experience, designing the model to be interpretable, understanding the trained model, and communicating explanations to model users [14]. FICO, the renowned credit score company, has published a white paper titled "Developing Transparent Credit Risk Scorecards More Effectively: An Explainable Artificial Intelligence Approach" to address interpretable credit scoring systems [15].

This paper focuses on outcome explanations that are methods that explain the outcome (e.g., classification result) of machine learning models for the specific instance. In the category of the outcome, explanation includes rule-based methods (explaining instances with simple logic rules) and attribution procedures (which gives an important score for each feature in input to the machine learning model). For rule-based outcome explanation, [16] is the first in introducing the requirement of precision and generality. Precision imposes that a rule explaining an instance with a classification outcome (e.g., classification 1) should not explain instances with a different classification outcome (e.g., classification 0). While generality implies that a rule explaining an instance of a particular classification outcome (e.g., classification 1) should potentially explain also other instances with the same classification outcome. Precision and generality are meaningful requirements increasing human trust in explanation models. These requirements are only defined and measured on rules, but they are not tested or even enforced in the attribution methods.

This paper provides an overview of model agnostic (the machine learning model is a black box, and it can only classify instances, but how the classification is performed is not considered) attribution procedures and provides a methodology to evaluate attribution procedures in terms of precision and generality. In addition, this work also provides a methodology to measure consistency between different generic attribution procedures.

The paper is organized as follows. In Section II is provided an overview of the outcome explanation methods. In Section III is provided our methodology to evaluate the performance of the outcome explanation methods. While in Section IV are reported the empirical evaluations of the explanation models. Ultimately, in Section V conclusion are provided.

II. Overview of Outcome Model Agnostic Explanation Methods

In a model agnostic method, the explanation is separated from the ML model. It gives the flexibility to use any interpretable ML method regardless of the ML model is defined, i.e., the ML model is used as an oracle model. This category includes basic approaches such us partial dependency plot (PDP) [17], individual conditional expectation [18], feature interaction based on H-statistic [19], and local surrogate models. The local surrogate models, differently from the other, focus on the explanation of a specific outcome of a single instance. In [20]–[34] are reported several related works in the field of machine learning.

In the following are reported the three relevant local surrogate models. Two of them LIME and SHAP are attribution methods, i.e., they provide an importance score for each feature in input to the classification model. In contrast, ANCHOR as a surrogate model uses simple rules.

A. LIME

LIME explains an outcome of a model by learning an interpretable model locally around the instance. LIME modifies a single instance data sample by tweaking the feature values and observes the resulting impact on the output.

The idea behind LIME is: for each individual instance that is passed to the model and for each outcome it makes, it performs a "local sensitivity analysis" in order to understand how sensitive is the prediction with regards to each feature of a particular instance. Figure 1 shows how LIME works in theory.

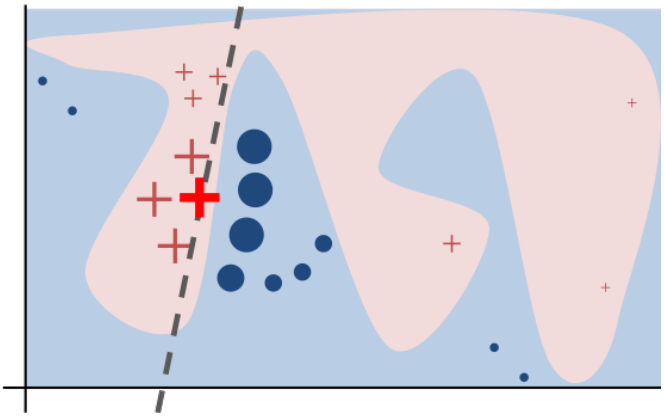

Figure 1: Lime abstraction from [35].

The original decision function is represented by the blue/orange background and is clearly nonlinear. The largest red X is the instance to explain. The approach simply perturbs instances around X and assigning weight according to their proximity to X. The weights here are represented by the sizes of the symbols — blue circles and red X_s. Given the model's outcome confidence on these perturbed instances, the approach learns a linear model (black line) that approximates the complex model well in the proximity of X. Please note that the explanation, in this case, is not true globally, but it is true locally around the instance X. The explanation produced by LIME at a local point x is obtained by the following generic formula:

$$\xi(x) = \underset{g \epsilon G}{argmin}\ L(f, g, \pi_x) + \Omega(g)$$

Where f is the real function (aka the machine learning model to explain), g — is a surrogate model used to approximate f in the proximity of x and πx defines the locality. This formulation can be used with different explanation families G, fidelity functions L, and complexity measures Ω. Here it is assumed that the complexity is opposed to explainability. Typically, g would belong to the family of linear functions (low complexity). The loss function L (measuring fidelity) minimizes the local mismatch between the complex machine learning function f and the approximating function g. Typically, g is a linear combination of the input features of the predictive model, and L is the well-familiar root mean square error loss function $RMSE$.

Under the use of the linear model, the coefficients of the linear model determine the importance scores of each feature. Then, LIME is an attribution method.

B. SHAP

SHAP (**SH**aply **A**dditive ex**P**lanations) [36] introduces a unified approach for interpreting model prediction from different interpretable techniques. For a particular prediction, SHAP assigns each feature an importance value. It unifies six explanation techniques LIME [35], Shapley sampling values [37], DeepLIFT [38], QII [39], Layer-wise relevance propagation [40], Shapley regression values [41] by defining the class of *additive feature attribution method*. SHAP employees game theory to compute the attribution and, in particular, uses the Shapley value to compute the attribution. The Shapley value indicates the reward that each player receives in a coalition game for his participation in the coalition. Such computation is done in the following way. Let F be the set of all features in input to the ML model and given an instance x and a machine learning model f, the attribution score $\phi_i^f(x)$ for each feature $i \in S$ is obtained by the following formula (which is an adaptation of the Shapely values):

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F|-|S|-1)\,!}{F!} \left[f_c(x_{S\cup\{i\}}) - f_c(x_S) \right]$$

where f_c is the confidence value of the ML model f for a particular outcome (e.g., a specific class) and x_S is the instance x where each value $x[u]$ of the feature $u \in F \setminus S$ is substituted with the mean of all the values that u has among all the possible instances. Then, this score indicates how relevant is in the instance x the specific value assigned to a feature u for the classification compared with the mean value of the feature.

As it is possible to observe, the computation of the feature score is exponential in the number of features. To overcome such complexity, approximations are provided.

One of these approximations is called Kernel SHAP [36] which is based on fitting a linear model defined as follows:

$$g(S) = \phi_0 + \sum_{j \in S} \phi_j$$

The fitting procedure consists in minimizing the following loss function:

$$\sum_{S \subseteq F} \mu(S)(g(S) - f_c(x_S))^2$$

where the kernel $\mu(S)$ is defined as $\mu(S) = \frac{|F|-1}{\binom{|F|}{|S|}|S|(|F|-|S|}$.

The speed up is obtained by considering in the loss function only a random subsamples $H \subset \{S | S \subseteq F\}$ and the optimizing the loss function $\sum_{S \in H} \mu(S)(g(S) - f_c(x_S))^2$.

This kernel approximation aligns SHAP with LIME.

The work [42] provides fast algorithm computation for SHAP when the machine learning model to explain is based on the decision tree, e.g., Random Forest.

C. Anchor

Anchor [16] explains individual prediction by defining a decision rule that "anchors" the prediction.

Anchor usages perturbation-based strategies like LIME for generating local explanations. Unlikely LIME, it uses If-Then rules for the explanation rather than a linear surrogate model. The goal of Anchor if produce rules that are precise and general. As explained in the introduction, this is the first approach that studying the problem to create rules precise and general.

Given x an instance to be explained, A is the anchor such that when all feature predicates defined by A correspond to x's value, such that $A(x) = 1$, f is the model to be explained, $D_x(.|A)$ is the neighbor distribution of x, considering τ as precision threshold, anchor A can be defined as:

$$\mathbb{E}_{\mathcal{D}_x(z|A)}[1_{f(x)=f(z)}] \geq \tau, A(x) = 1$$

The anchors are constructed bottom-up in combination with beam search. It starts with an empty rule or anchor and incrementally adds an if-then rule in each iteration until the minimal confidence constraint is satisfied. If multiple valid anchors are found, the one with the largest coverage is returned.

Anchor usages modified beam search or greedy search for searching be best candidate rules. Anchor can produce precise but not so general and some time due to its non-determinism, for the same instance may return different rules.

III. Measuring Attribution Precision, Generality, and Consistency

In this section, we describe the methodology to measure Precision, Generality, and Consistency in attribution models. Ways how to measure precision and generality for rules are already defined in [16], but not yet for attributions. In addition, this section proposes a method to measure the consistency between two generic attribution techniques.

A. Precision

Precision imposes that a rule explaining an instance with a classification outcome (e.g., classification 1) should not explain instances with a different classification outcome (e.g., classification 0). Intuitively, providing a rule that both explains two different outcomes of a machine learning model result inconsistent and not trustable by a human. In [16], the precision of a rule r explaining an outcome a is measured inversely by the percentage of instances obtaining from the machine learning model an outcome different from a and r applies to such instances. To define precision to attribution results, we first introduce two functions. The first one is $sel(S, x)$ which returns a vector in $\mathbb{R}^{|S|}$ which is the selection of the values of the feature in S. More formally, given $S = i_1, \ldots, i_k$ the subset of features for each $j \in \{1, \ldots, k\}$, $sel(S, x)[j] = x[i_j]$. The second function is $top_k : \mathbb{R}^n \rightarrow 2^{|k|}$ which given the attribution att vector returns the top-k feature according to att. Let I_a be the set of instances receiving the outcome a and $I_{\neg a}$ be the set of instances not receiving the outcome a, given an instance $x \in I_a$ and its attribution att_x, the attribution precision can be inversely measured by the Reverse Precision (RP) as follows:

$$RP^k(x, att_x) = \frac{|\{\hat{x} | \hat{x} \in I_{\neg a}, sel(S^x_{at}, x) = sel(S^x_{at}, \hat{x})\}|}{|I_{\neg a}|}$$

where $S^x_{at} = top_k(att_x)$. Intuitively, the reverse precision measures the percentage of instances with outcome a that have

the same values of top-k feature with the specific explained instance. Given a particular outcome a we compute the average of the reverse precision scores at k for each instance in I_a as

$$avgRP^k(I_a) = \frac{\sum_{x \in I_a} RP^k(x, att_x)}{|I_a|}$$

B. Generality

Generality implies that a rule explaining an instance of a particular classification outcome (e.g., classification 1) should potentially explain also other instances with the same classification outcome. Given two instances x_1 and x_2 with their attribution vectors att_{x_1} and att_{x_2}, the function that measures how many top-k features are in common between att_{x_1} and att_{x_2} is defined as follow:

$$common_k(att_{x_1}, att_{x_2}) = |top_k(att_1) \cap top_k(att_2)|$$

where the functions top_k is defined in the section of the precision. Given an instance $x \in I_a$ we measure the generality of hits attribution att_x in the context of the top h neighbour instances in I_a of x with the function $generality(x, k, h, agg)$ defined as follows:

$$agg(\{common_k(att_x, att_{\hat{x}}) | \hat{x} \in topNeighbour_h(x, I_a)\})$$

where $agg \in \{sum, \min, \max\}$ and $topNeighbour_h(x, I_a)$ selects the top h neighbour instances in I_a of x. The different aggregation function provides more information on how the commonalities of the attributions are distributed among the op h neighbor instances. Please note that the function $common_k$ does not consider the values of the top_k features from the attributions; this is justified because, in the $generality$ function, the $common$ function is used only between nearest neighbors, then we assume that the values of these instances should be similar. Given a particular outcome a we compute the average of the generality scores at k for the top-h neighbour instances for each instance in I_a as

$$avgGen(I_a, k, h, agg) = \frac{\sum_{x \in I_a} generality(x, k, h, agg)}{|I_a|}$$

C. Consistency

Since SHAP is an attribution method that unifies multiple attribution methods, one of them LIME, then we propose a simple procedure to compare the attribution of two different methods. Given an instance x and an attribution method m, we denote $attr_m(x)$ the attribution scores for the instance x provided by the method m.

Given the set of instances I_a with outcome a and two attribution methods m_1 and m_2, the consistency score for the top-k features between m_1 and m_2 is defined as follows:

$$cons_k(I_a, m_1, m_2) = \frac{\sum_{x \in I_a} common_k(att_{m1}(x), att_{m_2}(x))}{|I_a|}$$

where the function $common_k$ is defined in the section generality.

IV. Experiment

In this section, we apply our methodology to evaluate attribution methods in two attack detection contexts: network traffic and power system. In particular, we focus on the explanation (attribution) of attack instances. We first describe the two datasets. Second, we show the classification results of different classification models in detecting such attacks. Then, we select the best classification model, and we use or methodology to evaluate LIME and SHAP attribution approach in interpreting the attack instances correctly classified by the best classification model.

A. Datasets

The datasets of the two attack detection contexts are provided below.

1) UNSW-NB 15: Network Traffic: This dataset [43] represents a comprehensive network-based dataset that can reflect modern network traffic scenarios, wide varieties of low footprint intrusions, and depth structured information about the network traffic. The raw network packets of this dataset were created by the *IXIA PerfectStorm* tool in the *Cyber Range Lab* of the *Australian Centre for Cyber Security (ACCS)*. It contains real normal behavior and synthesized attack activities of network traffic. The simulation period was 16 hours on Jan 22, 2015, and 15 hours on Feb 17, 2015. It consists of $2,540,044$ records. It contains 49 distinct features.

2) ICS: Power System: This dataset [44] captures various scenario of power system disturbance. This dataset is derived from one initial dataset which contains 15 sets with 37 power system event scenario each (28 attack events and 9 normal events). It consists of total 128 features and $78,377$ records.

B. Classification Results

Before starting the classification, we transform all the nonnumeric features with one-hot encoding. We first split the dataset in 70% trainingset and 30% testset. In the case of "UNSW-NB 15", the split was already provided but with the same percentages. Then, we train and test the following classification models: Logistic Regression, Random Forest, KNN, Support Vector Classification (with RBF kernel). Table I and Table IV shows the classification results (Precision, Recall, F1-score and Accuracy) for all the classification models. As it is possible to see, the best results are provided by the KNN and Random Forest, which are comparable. Then, to apply our methodology to evaluate the attribution methods, we only focused on the Random Forest classifier because SHAP computation is more efficient.

C. Attribution Precision Analysis

In Figure 2 and Figure 3 is shown $avgRP^k(I_a)$ for LIME and SHAP by varying the number k in "UNSW-NB 15" and "ICS: Power System", respectively.

As it is possible to see, even for $k = 6$ in both the datasets, the values of the top-6 features , according to the attributions of the attack instance for both the methods (LIME and SHAP), are the same for the normal behavior.In fact, $avgRP^6(I_a)$ is

Table I: Precision, Recall, F1-score and Accuracy of Linear regression, Random Forest, KNN and SVC algorithms for "UNSW-NB 15".

	Logistic regression			Random Forest		
	precision	***recall***	***f1-score***	***precision***	***recall***	***f1-score***
Not Attack	0.82	0.87	0.84	0.90	0.98	0.94
Attack	0.90	0.86	0.88	0.98	0.92	0.95
Accuracy		0.86			0.95	
Macro avg.	0.86	0.87	0.86	0.94	0.95	0.94
Weighted avg.	0.87	0.86	0.86	0.95	0.95	0.95
	KNN			**SVC**		
	precision	***recall***	***f1-score***	***precision***	***recall***	***f1-score***
Not Attack	0.92	0.98	0.95	0.88	0.92	0.86
Attack	0.99	0.93	0.96	0.93	0.84	0.91
Accuracy		0.95			0.86	
Macro avg.	0.95	0.96	0.95	0.88	0.92	0.90
Weighted avg.	0.96	0.95	0.95	0.89	0.86	0.87

Table II: Precision, Recall, F1-score and Accuracy of Linear regression, Random Forest, KNN and SVC algorithms for "ICS: Power System".

	Logistic regression			Random Forest		
	precision	***recall***	***f1-score***	***precision***	***recall***	***f1-score***
Not Attack	0.54	0.03	0.05	0.94	0.78	0.85
Attack	0.72	0.99	0.83	0.92	0.98	0.90
Accuracy		0.72			0.92	
Macro avg.	0.63	0.51	0.44	0.93	0.88	0.90
Weighted avg.	0.67	0.72	0.61	0.92	0.92	0.92
	KNN			**SVC**		
	precision	***recall***	***f1-score***	***precision***	***recall***	***f1-score***
Not Attack	0.94	0.78	0.85	0.88	0.92	0.86
Attack	0.92	0.98	0.95	0.93	0.84	0.91
Accuracy		0.92			0.86	
Macro avg.	0.93	0.88	0.90	0.88	0.92	0.90
Weighted avg.	0.92	0.92	0.92	0.89	0.86	0.87

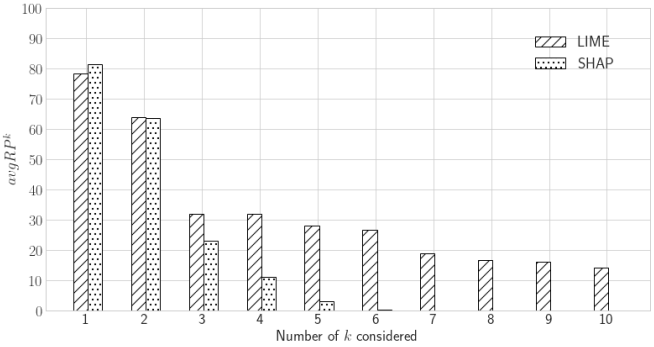

Figure 2: $avgRP^k(I_a)\%$ for LIME and SHAP by varying the number k in "UNSW-NB 15".

greater than zero. This shows that both LIME and SHAP are not so precise since the explanation provided for the attack instances also applies to normal behavior instances. In addition, there is no attribution technique that is better than another. It is also impressive to see that the value of the top-1 most important feature (according to the specific attribution procedure) is the same in so many normal behavior instances, more than 70% in " UNSW-NB 15" and 50% in "ICS: Power System". This gives an intuition of how precision remains still an open problem in the attribution methods.

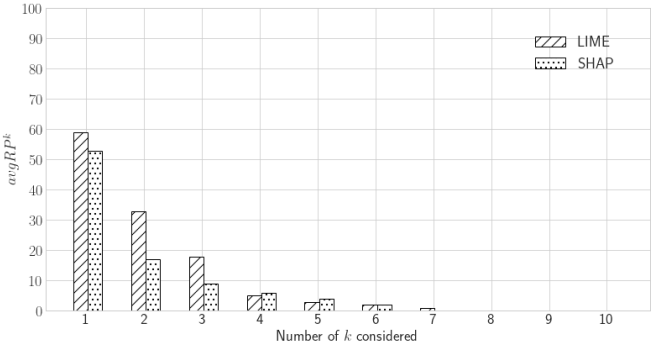

Figure 3: $avgRP^k(I_a)\%$ for LIME and SHAP by varying the number k in "ICS: Power System".

Table III: $avgGen(I_a, k, h, agg)$ by varying k (the number of top features) and h (the number of close neighbours) for LIME and SHAP in "UNSW-NB 15".

		Mean Lime intersection size			Mean SHAP intersection size		
		Max	**Mean**	**Min**	**Max**	**Mean**	**Min**
No of Neighbors (h)	**No of Features (k)**						
1	**1**	1.00	0.21	0.00	1.00	0.78	0.00
	5	5.00	1.66	0.00	5.00	4.21	0.00
	10	10.00	4.47	1.00	10.00	8.50	1.00
5	**1**	0.60	0.14	0.00	1.00	0.75	0.00
	5	2.80	1.50	0.20	5.00	4.08	0.80
	10	6.20	4.23	2.40	10.00	8.29	3.00
10	**1**	0.50	0.13	0.00	1.00	0.73	0.00
	5	2.50	1.50	0.50	5.00	4.03	0.70
	10	5.80	4.23	2.60	10.00	8.19	3.30

D. Attribution Generality Analysis

In Table IV and Table III are shown the values of $avgGen(I_a, k, h, agg)$ by varying k (the number of top features) and h (the number of close neighbours) for LIME and SHAP in "UNSW-NB 15" and "ICS: Power System", respectively. From the results, the attributions of both the methods it is not so general since even the closest instances produce attributions that are drastically different. Then each attribution seems unique for the specific instance rather than generic.

E. Attribution Consistency Analysis

In this experiment, we show how LIME and SHAP are consistent, especially in consideration that SHAP should be a model that unifies several other attribution models included LIME. Figure 4 and Figure 5 shows $cons_k(I_a, LIME, SHAP)$ by varying the number k in "UNSW-NB 15" and "ICS: Power System" respectively. As it is possible to see both LIME and

Table IV: $avgGen(I_a, k, h, agg)$ by varying k (the number of top features) and h (the number of close neighbours) for LIME and SHAP in "ICS: Power System".

		Mean Lime intersection size			Mean SHAP intersection size		
		Max	**Mean**	**Min**	**Max**	**Mean**	**Min**
No of Neighbors (h)	**No of Features (k)**						
1	**1**	0.00	0.00	0.00	1.00	0.39	0.00
	5	2.00	0.24	0.00	5.00	2.18	0.00
	10	5.00	1.16	1.00	10.00	4.78	1.00
5	**1**	0.20	0.00	0.00	1.00	0.195	0.00
	5	1.00	0.29	0.00	3.6.00	1.47	0.20
	10	2.40	1.17	0.20	6.60	3.67	0.80
10	**1**	0.10	0.01	0.00	0.7	0.12	0.00
	5	0.80	0.30	0.00	2.90	1.20	0.10
	10	2.30	1.20	0.40	5.80	3.17	0.60

SHAP agree over less than half of the top-k features for both the datasets. Especially in terms of top-1, top-2, and top-3 features, the two attribution methods are in strong disagreement. This analysis shows that the attributions provided by the two attribution are different, and due to the poor performances in precision and generality, it is difficult to determine the best attribution model.

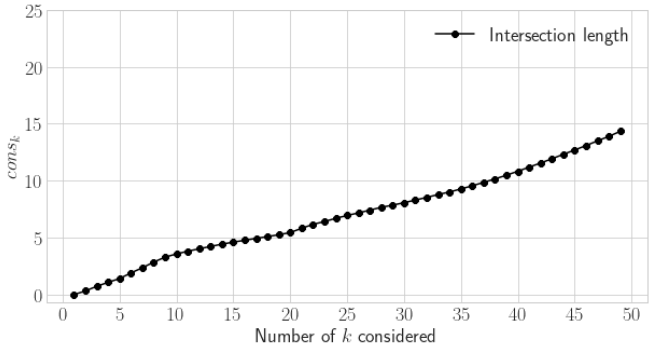

Figure 4: $cons_k(I_a, LIME, SHAP)$ by varying the number k in "UNSW-NB 15".

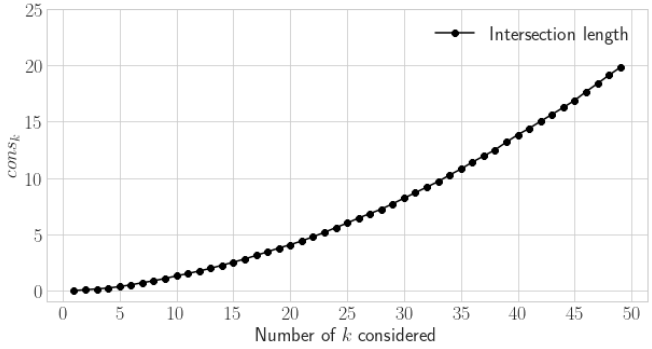

Figure 5: $cons_k(I_a, LIME, SHAP)$ by varying the number k in "ICS: Power System"

V. Conclusion

Attribution models are important to evaluate the interpretability of machine learning models. In this paper, a new methodology to evaluate the precision, generality, and consistency of attribution models is provided. We used such methodology to evaluate two of the most popular model agnostic attribution models, LIME and SHAP, on two attack classification tasks involving network traffic and power systems in the industrial control system field. Our methodology showed that both LIME and SHAP lack precision and generality and none of the two was better than the other. Even if SHAP is proposed as the unification model and should generalize LIME, we observed that the attribution results of the two attribution methods in many cases were very different. From this evaluation, we have determined that there is no a better model in the attribution field and that still research is needed to overcome the precision and generality limitations in this field.

Acknowledgment

This research was made possible by the National Science Foundation award #1820685 and Idaho Global Entrepreneurial Mission/Higher Education Research Council #IGEM22-001.

References

[1] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[2] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[3] S. Temizer, M. Kochenderfer, L. Kaelbling, T. Lozano-Pérez, and J. Kuchar, "Collision avoidance for unmanned aircraft using markov decision processes," in *AIAA guidance, navigation, and control conference*, 2010, p. 8040.

[4] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[5] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[6] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley, "Google vizier: A service for black-box optimization," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 1487–1495.

[7] C. Rudin, "Please stop explaining black box models for high stakes decisions," *arXiv preprint arXiv:1811.10154*, vol. 1, 2018.

[8] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, no. 2, 2017.

[9] P. Press, "Preparing for the future of artificial intelligence," 2016.

[10] C. Villani, "French national strategy for artificial intelligence." 2019. [Online]. Available: https://www.aiforhumanity.fr/en/

[11] "Portuguese national initiative on digital skills. ai portugal 2030. 2019." 2019. [Online]. Available: https://www.incode2030.gov.pt/sites/default/files/draft_ai_portugal_2030v_18mar2019.pdf

[12] "Machine learning: The power and promise of computers that learn by example." 2019. [Online]. Available: https://royalsociety.org/topics-policy/projects/machine-learning/

[13] "European commission. algorithmic awareness-building. 2018." 2019. [Online]. Available: https://ec.europa.eu/digital-single-market/en/algorithmic-awareness-building

[14] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[15] G. Fahner, "Developing transparent credit risk scorecards more effectively: An explainable artificial intelligence approach," *Data Anal*, vol. 2018, p. 17, 2018.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[17] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 10 2001. [Online]. Available: https://doi.org/10.1214/aos/1013203451

[18] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," 2014.

[19] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," 2008.

[20] M. Ceci, A. Cuzzocrea, and D. Malerba, "Supporting roll-up and drill-down operations over olap data cubes with continuous dimensions via density-based hierarchical clustering." in *SEBD*. Citeseer, 2011, pp. 57–65.

[21] E. Serra, M. Joaristi, and A. Cuzzocrea, "Large-scale sparse structural node representation," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 5247–5253.

[22] P. Braun, A. Cuzzocrea, T. D. Keding, C. K. Leung, A. G. Padzor, and D. Sayson, "Game data mining: clustering and visualization of online game data in cyber-physical worlds," *Procedia Computer Science*, vol. 112, pp. 2259–2268, 2017.

[23] A. Guzzo, D. Sacca, and E. Serra, "An effective approach to inverse frequent set mining," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 806–811.

[24] K. J. Morris, S. D. Egan, J. L. Linsangan, C. K. Leung, A. Cuzzocrea, and C. S. Hoi, "Token-based adaptive time-series prediction by ensembling linear and non-linear estimators: a machine learning approach for predictive analytics on big stock data," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1486–1491.

[25] E. Serra and V. Subrahmanian, "A survey of quantitative models of terror group behavior and an analysis of strategic disclosure of behavioral models," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 66–88, 2014.

[26] L. Bellatreche, A. Cuzzocrea, and S. Benkrid, "$F\&A$: A methodology for effectively and efficiently designing parallel relational data warehouses on heterogenous database clusters," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2010, pp. 89–104.

[27] O. Korzh, M. Joaristi, and E. Serra, "Convolutional neural network ensemble fine-tuning for extended transfer learning," in *International Conference on Big Data*. Springer, 2018, pp. 110–123.

[28] S. Ahn, S. V. Couture, A. Cuzzocrea, K. Dam, G. M. Grasso, C. K. Leung, K. L. McCormick, and B. H. Wodi, "A fuzzy logic based machine learning tool for supporting big data business analytics in complex artificial intelligence environments," in *2019 IEEE international conference on fuzzy systems (FUZZ-IEEE)*. IEEE, 2019, pp. 1–6.

[29] E. Serra, A. Sharma, M. Joaristi, and O. Korzh, "Unknown landscape identification with cnn transfer learning," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 813–820.

[30] E. Serra, A. Shrestha, F. Spezzano, and A. Squicciarini, "Deeptrust: An automatic framework to detect trustworthy users in opinion-based systems," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020, pp. 29–38.

[31] M. Joaristi, E. Serra, and F. Spezzano, "Inferring bad entities through the panama papers network," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 767–773.

[32] ——, "Detecting suspicious entities in offshore leaks networks," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1–15, 2019.

[33] M. Joaristi and E. Serra, "Sir-gn: A fast structural iterative representation learning approach for graph nodes," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 6, pp. 1–39, 2021.

[34] M. Joaristi, A. Putnam, A. Cuzzocrea, and E. Serra, "Ribs: Risky blind-spots for attack classification models," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 5773–5779.

[35] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," 2016.

[36] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[37] S. Kaufman, S. Rosset, and C. Perlich, "Leakage in data mining: formulation, detection, and avoidance," in *KDD*, 2011.

[38] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3145–3153.

[39] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 598–617.

[40] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[41] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.

[42] S. M. Lundberg and S.-I. Lee, "Consistent feature attribution for tree ensembles," *arXiv preprint arXiv:1706.06060*, 2017.

[43] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[44] R. C. Borges Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *2014 7th International Symposium on Resilient Control Systems (ISRCS)*, 2014, pp. 1–8.