Fooling Perturbation-Based Explainability Methods

Rahel Wilking, Matthias Jakobs, Katharina Morik

Fooling Perturbation-Based Explainability Methods

Rahel Wilking[0000−0003−2018−1219], Matthias Jakobs[0000−0003−4607−8957], and Katharina Morik[0000−0003−1153−5986]

Artificial Intelligence Group
Department of Computer Science, TU Dortmund, Germany
{firstname.lastname}@tu-dortmund.de

Abstract. Explanations are used to promote trust in machine learning models. If these explanations can be arbitrarily manipulated, models could be trusted by users based on their explanations but make biased predictions on real data. This manipulation is possible for several explainability methods including perturbation-based methods like LIME and KernelSHAP. We show that three methods, Anchors, LORE and EXPLAN are vulnerable to the same scaffolding attack that is effective against LIME and KernelSHAP. While the scaffolding attack is designed to target specific explainability methods, we show that KernelSHAP and especially LIME are also vulnerable to most attacks targeting other methods. We found that the explainability method most resistant against other attacks is Anchors. Additionally, we propose a fooling heuristic that quantifies the degree of fooling to enable objective comparison between different attack results. The code is available at https://github.com/RahelWilking/fooling-explainability-methods.

Keywords: Explainability · Adversarial Attacks · Perturbations.

1 Introduction

In critical domains such as healthcare or finance wrong decisions of machine learning models can have high negative impacts on humans. Machine learning models that aid decision makers in these domains have to fulfill high expectations. To verify these expectations and promote trust in these models, explanations of the model decisions can be generated [4]. A problem arises if these explanations do not represent the true model behavior and, in fact, can be arbitrarily manipulated. Several works have shown that this is possible [2,3,5,6,8,12,21]. Slack et al. [21] developed an attack on perturbation-based explainability methods and showed its effectiveness against commonly used local explainability methods LIME and KernelSHAP. We expand upon their work by examining a further three perturbation-based methods on their susceptibility to the attack. The so-called scaffolding attack is always targeted against a specific explainability method. We also examine how effective the attack is against other methods it was not specifically targeted at or in reverse, which method resists

the attack when it is not specifically targeted. This shows a robustness of an explainability method against the scaffolding attack. To ease the comparison between the success of attacks we also propose a heuristic to summarize the degree of fooling achieved by the attack called the fooling heuristic $\mathcal{F}$. In summary our contributions are

- We extend the examination of the scaffolding attack by [21] to three further methods.
- We evaluate the cross effectivity of attacks, i.e. how successful an attack is if it targets a different explainability method than expected.
- We propose a fooling heuristic $\mathcal{F}$ that quantifies the level of fooling.

The remainder of the paper is structured as follows. In the next section attacks that have been developed are presented. After that the explainability methods used in the experiments are introduced and the fooling heuristic is defined. This is followed by the description of the experiments and their results. In the last section the results are summarized, and open research questions are identified.

2 Related Work

The works described in this section aim to manipulate explanations of a machine learning model f. For local explanation methods this consists of changing the explanation an explainability method produced for the prediction of a data point x. We first present the attack used for our experiments. Some other approaches are mentioned in section 2.2.

2.1 Scaffolding Attack

Slack et al. [21] proposed the scaffolding attack which targets a situation where a model developer wants to hide a bias of their model during examination by an explainability method. This method could potentially be employed by a model certification authority which is only granted black box access to the model. They further focus the attack on perturbation-based explanation methods because of their popularity. LIME [18] and KernelSHAP [14] are representatives of this category of methods. The perturbations of these methods often create out-of-distribution (OOD) examples [21], which the attack exploits. Since predictions for OOD examples are mostly made in examination situations, the model can pretend to make harmless predictions on them and make biased predictions on in-distribution data. This results in untruthful and seemingly unbiased explanations.

In addition to the original (biased) model f the attack needs a model ψ for which explanations should be shown during examination, as well as a training dataset X from the real data distribution X_{dist}. On the basis of this dataset an OOD classifier is trained. The attack output is a scaffolded model e which makes predictions as follows:

$$e(x) = \begin{cases} f(x), & x \in X_{dist} \\ \psi(x), & \text{otherwise} \end{cases} \tag{1}$$

The distinction between the cases is made by a random forest model which is trained on the training data and data representing the perturbations made by a specific explainability method. The attack is therefore specialized for one method.

Slack et al. [21] showed the success of their scaffolding attack on both LIME and KernelSHAP. LIME and KernelSHAP both optimize the following problem [14,18]:

$$\xi(x) = \underset{g \in G}{\arg\min}\, L(f, g, \pi_x) + \Omega(g). \tag{2}$$

Here, g is a model from an interpretable model class G, e.g. sparse linear models in the case of LIME. The function $\pi_x(z)$ is a similarity measure between a point z and the example x whose prediction should be explained. $\Omega(g)$ presents a cost of the complexity of g, e.g. the number of non-zero weights of a linear model. L describes a loss function which for LIME is chosen as [18]:

$$L(f, g, \pi_x) = \sum_{z \in Z} \left[f(z) - g(z)\right]^2 \pi_x(z). \tag{3}$$

The values of Ω, π_x and L in the case of KernelSHAP are chosen such that the optimization results in Shapely values [14].

2.2 Other Attacks

In addition to the scaffolding attack several other similar approaches were proposed. Many of these attacks are focused on deep neural networks (DNN) and often on the image domain and the corresponding commonly used explainability methods [5,6,8]. They use perturbations or fine-tuning steps after the training to achieve the manipulated explanations. Anders et al. [2] also expand their approach to logistic regression and propose a defense against the weakness their attack exploits. Lakkaraju and Bastani [12] show that the global explanation method MUSE can produce misleading explanations caused by being able to reconstruct sensitive features through other seemingly harmless features. Baniecki et al. [3] succeed in fooling partial dependence by a data poisoning attack. Part of their attack is also a robustness check creating a maximally different explanation. The most similar attacks to the scaffolding attack are the approaches which manipulate the model, not the training data or the explanation method. They are the DNN focused approaches of Heo et al. [8], Dimanov et al. [5] and Anders et al. [2]. Dimanov et al. also show a fooling of LIME and SHAP. In contrast to these works we focus on model-agnostic methods on tabular data.

3 Methods

Several local model-agnostic explainability methods work by creating a neighborhood Z around an example point x of which the explanation should be explained and then learning a local interpretable model g on this neighborhood to approximate the decision boundary of the black box model f. LIME and KernelSHAP

fit this description, as do LORE and EXPLAN which are two of the methods examined here. Creating a neighborhood based on a point is achieved through perturbations of the training data or the example x. The third perturbation-based method for which the scaffolding attack is examined is Anchors.

The attack mainly consists of the perturbations used for training the OOD classfier. For each point of the dataset X perturbations according to an explainability method are created which results in a set of perturbations X_p. The original training data is multiplied to balance the final dataset of original examples and perturbations. The original datapoints are assigned the label 1 and the perturbations the label 0. The created perturbations are checked against the training data and any perturbation that duplicates a real data point is assigned the correct label. For LORE and EXPLAN the perturbations for each data point are subsampled from the neighborhood created by the method described in the corresponding following sections. The perturbation process for Anchors is described in section 3.3.

The parameter sets of the attacks are chosen by optimizing both the fidelity of the scaffolded model on a test set and a heuristic summarizing the level of fooling achieved on the test set. This fooling heuristic is described in section 3.4.

3.1 LORE

LOcal Rule-based Explanations (LORE) by Guidotti et al. [7] consist of rules and counterfactuals extracted from a local decision tree g learned upon a neighborhood Z which is created by a genetic algorithm. The extracted rule is the conjunction of node decisions of g on the path of x through the tree. The counterfactuals are minimal differences that lead to a path with a leaf assigned to a different prediction. The genetic algorithm is executed twice, preferring a different class each time and the results are joined to achieve a balanced training set for g. The fitness function rewards individuals of the preferred class close to x but penalizes individuals who are identical to x to produce a neighborhood that is dense around the example under inspection.

The class aspect of the fitness function necessitates a predictor to be available for the neighborhood generation. For the OOD classifier training data this can only be the original black box model f, not the scaffolded model e which is used during the testing.

3.2 EXPLAN

EXPLaining black-box classifiers using Adaptive Neighborhood generation (EXPLAN) [16] focuses on the neighborhood generation. Similar to LORE it learns a local decision tree on the created neighborhood and extracts a decision rule in the same way. According to Rasouli and Yu [16] a good neighborhood is compact, representative, diverse and balanced. To achieve such a neighborhood several steps are executed.

The neighborhood generation starts with a random sample of perturbed data points according to the feature distributions of a training dataset. On this dataset

Z' a random forest is trained using $f(Z')$ as labels. The random forest model is used to extract local feature importances for each example which are used to change examples to be closer to x. This creates a dense neighborhood around x. The process is described in detail in an earlier publication by the same authors [15].

In a next step only parts of the resulting neighborhood are kept. The selection happens through an iterated agglomerative clustering procedure where only the data points clustered closer to x are retained until a stopping criterion is reached. In a final step the neighborhood is balanced according to the predictions made by f via oversampling.

As with LORE the neighborhood generation process of EXPLAN needs access to the model that is to be explained. For the OOD classifier training dataset again the original model f is used, though the explainability method is later applied to e which may create a different neighborhood.

3.3 Anchors

Anchors are an explainability method developed by Ribeiro et al. [19] who also created LIME. An anchor A is a decision rule that is satisfied by x and has a high (configurable) precision. Anchors with high coverage are preferred. A rule exactly describing x is always an anchor, so an anchor exists for each example point. To construct an anchor with high coverage that still fulfills the precision requirement with high probability the authors use a multi-armed bandit algorithm. The anchor is constructed bottom-up since shorter rules usually have higher coverage. The precision and coverage of candidate rules A are estimated over samples from the conditional distribution $\mathcal{D}(\cdot|A)$ of examples that satisfy all conditions of A.

Since the Anchors method does not have an explicit neighborhood creation the designed attack for it differs in the creation of the perturbations to train the OOD classifier. The perturbations happen through samples taken from the distribution $\mathcal{D}(\cdot|A)$ for different candidate rules A. All unconstrained features are sampled as a row from the training distribution and constrained features are possibly adapted to satisfy the rule. The bottom-up creation of anchors causes shorter candidate rules to be evaluated more frequently than longer rules. For the perturbations a set of rules is randomly created and then samples are taken from their conditional distribution. The length of these rules is sampled from a geometric distribution with parameter `p` and the rule items are sampled without replacing from all possible rule conditions. The number of samples per candidate rule is parametrized and the number of tuples is chosen to match the multiplication rate of the original data. As an additional step for the anchors perturbations all duplicates of original data points are removed and the multiplied original data is subsampled to the same amount of examples as the final perturbation set. This step is only taken for Anchors, since the amount of duplicates is very high for this approach. This is explained by short rules causing only few features to be possibly changed to fit the rule conditions. Some data points may exist with a substituted value but many are not changed at all.

3.4 Fooling Heuristic

To evaluate the success of fooling, the rankings of features in the explanations are summarized over all test data points. For each feature a it is calculated in which proportion of the test explanations the feature appeared at one of the first three ranks, or a later or no one, to result in a distribution $\mathcal{D}(a)$. The ideal distribution for a feature $a \in A$ of features relied on by the biased model f is $p = (0, 0, 0, 1)$ while for a feature $b \in B$ of features used by ψ it is $q = (1, 0, 0, 0)$.

$$\mathcal{F}(p, q, \mathcal{D}(\cdot), A, B) = \left(\sum_{a \in A} \frac{1}{KL(p \parallel \mathcal{D}(a))} + \sum_{b \in B} \frac{1}{KL(q \parallel \mathcal{D}(b))} \right)^{-1} \tag{4}$$

The fooling heuristic $\mathcal{F}$ computes the harmonic mean of the Kullback-Leibler divergences of the real distributions to the ideal distributions for each relevant feature of type a or b. All other features are ignored. A lower value represents a better fooling. The heuristic summarizes the fooling result over the whole test dataset in one number to facilitate easier comparison between different models or different versions of the same model. It was also used to select the parameter settings for the attacks.

$\mathcal{F}$ emphasizes small differences in the proportions for the first rank above the second and third rank. The value range of the heuristic is also not limited and the same absolute difference in heuristic values does not correspond to a similar level of change in fooling quality for different areas of the value range. High absolute differences on the higher end are caused by very minimal changes in the proportions that cause less change of $\mathcal{F}$ for an overall better fooling. The focus on the first rank may be an advantage if what is the most important feature of an explanation is seen as more relevant than the following ranks.

4 Experiments and Results

The explainability methods are examined in two different kinds of experiments on three different datasets. The datasets are described in the next section as are the parameter settings per dataset. For all experiments the black box model f is a model relying on only one sensitive feature to make the prediction. For two of the datasets two artificial features that are uncorrelated with the sensitive feature are added and the model ψ is making predictions based on either only the first of these artificial features or the xor connection of the features in two different experiment setups. For the third dataset only one ψ is used based on a single real data feature that has very low correlation with the sensitive feature used for f. For all experiments the OOD classifier is a random forest of 100 trees.

The first kind of experiment examines the susceptibility of the methods to the scaffolding attack that is targeted against them. The results for LIME and KernelSHAP are computed for comparison with Slack et al. [21]. The cross effectiveness of the different targeted attacks and the explainability methods is examined in the second type of experiment. The scaffolded models are reused from the first experiment type. The code is available at https://github.com/RahelWilking/fooling-explainability-methods.

Table 1. Parameters chosen via the parameter optimization.

Targeted Method	Parameter	compas	cc	german
Anchors	`perturbation_multiplier`	1	1	2
	`p`	0.8	0.8	0.6
	`n_samples_per_tuple`	1	1	1
LORE	`perturbation_multiplier`	8	2	4
EXPLAN	`perturbation_multiplier`	3	1	3

4.1 Data

For ease of comparison we use the same three datasets used by Slack et al. [21]. The first dataset is the COMPAS [11] dataset. It contains information about the criminal history, prison and jail times, demographic values as well as the COMPAS risc score of defendants in Broward County, Florida. After preprocessing, the dataset has 6172 examples and the target is whether a defendant is classified as a high risk to reoffend. The sensitive attribute used for f is *race*. The second dataset is the Communities and Crime (CC) [17] dataset containing statistics about crime, socio-economic and law enforcement aspects of communities in the US. It contains 1994 data points after preprocessing with the target being if the violent crime rate is above the dataset median. The sensitive attribute is the percentage of white population. For these first two datsets the artificial features are added for ψ. The last dataset is the German Credit [9] dataset describing loan applicants with information about the loan as well as financial history and personal information. The goal is to predict if the applicant is a good customer and it consists of 1000 examples. The sensitive feature in this case is *gender*. The feature used for ψ is the loan rate as a percentage of the income.

For each dataset the parameters for the perturbation creation were optimized for fidelity of the attack and degree of fooling measured by the fooling heuristic $\mathcal{F}$ described in section 3.4. From the resulting pareto front we picked an option that seemed to balance the values best. The chosen parameters can be seen in Table 1. The Anchors method requires a validation set during its application. For this the training set was subdivided to create a validation set containing 20% of the training data.

4.2 Evaluation

As already mentioned in section 3.4 for each example in the test set the explanation is converted into a qualitative ranking. These rankings are then summarized over all test examples and for each rank it is determined which proportion of the test set assigns a feature to this rank. The distributions across features for the first three ranks is displayed in the results of the targeted attacks in section 4.3. Based on this ranking the fooling heuristic $\mathcal{F}$ is computed as described in section 3.4. In addition, the fidelity of the scaffolded model e on the test set is

determined as the accuracy of the predictions of e in reference to the predictions of f. The test set contains 10% of the data points. The cross effectivity experiments are evaluated in the same way as the targeted attack. The plots only show the first rank.

Transforming the explanations into a qualitative ranking is straight forward for LIME and KernelSHAP but needs an intermediary step for the three new methods. For LORE and EXPLAN a ranking is extracted by computing the feature importances of the local decision tree with the mean decrease in impurity method [13]. For Anchors, the greedy nature of the creation algorithm is exploited to in a first step rank the features in the rule conditions according to the order they were added to the rule. Since for continuous features multiple conditions of a rule can constrain the same feature, some ranks have to be combined. Each position $i \in \{0, 1, \ldots, |t| - 1\}$ in the preliminary ranking is assigned a number $|t| - i$ for a ranking length of t. The values for the same features are summed, and the final list is re-sorted to result in the qualitative ranking of the features.

4.3 Targeted Attack Results

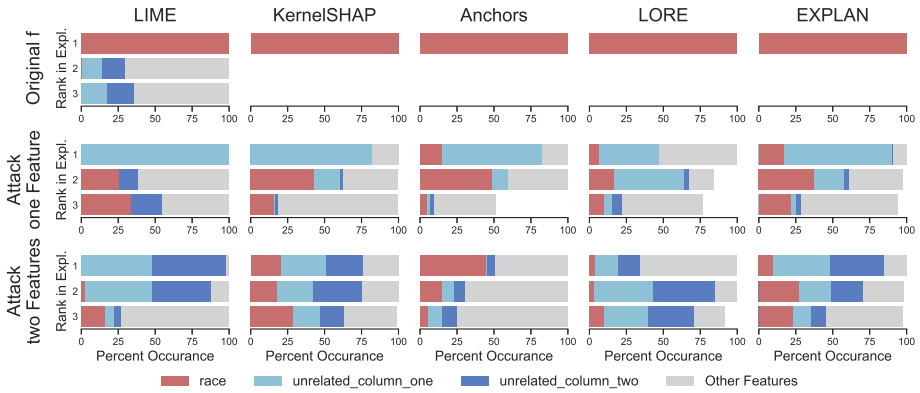

Fig. 1. Summary of targeted attacks on COMPAS. Columns represent attacked methods, rows attacks with differing ψ, or no attack in the first row.

The results of the targeted attack on COMPAS are shown in Figure 1 along with the summarized explanations without any attack. The corresponding fidelity values for all targeted attacks are collected in Table 2. The fidelity is coupled with the fooling success. It is always possible to reach a good fidelity without any fooling as well as reach a very good fooling without good fidelity. The goal is to achieve good fooling while also having a high fidelity. The fidelity values for the three new methods are lower than those for LIME and KernelSHAP on COMPAS and German Credit, but reach similar values on Communities and

Table 2. Fidelity scores of the scaffolded models on the test set in the targeted attacks. Columns represent different attacks over the datasets, rows represent the targeted method. Values are rounded to two decimals.

	COMPAS		CC		German
	one feature	two features	one feature	two features	one feature
LIME	1.00	0.99	1.00	1.00	1.00
KernelSHAP	0.87	0.86	0.97	0.96	0.98
Anchors	0.68	0.66	0.82	0.82	0.61
LORE	0.63	0.65	0.92	0.90	0.81
EXPLAN	0.72	0.74	1.00	1.00	0.77

Crime. The fidelity for Anchors is usually the lowest, with LORE and Anchors being similar on COMPAS. Even though the fidelity is relatively low Anchors on COMPAS is fooled the least. This can be seen subjectively, especially for the attack with two features used for ψ but also numerically on the main diagonals of Table 3 in the fooling heuristic $\mathcal{F}$. The most fooled method subjectively and numerically is LIME. All three new methods still have test examples where the sensitive feature is identified as the most relevant for the prediction. A pattern that can also be seen in the other experiments is that LORE explanations tend to choose features that are used by neither f nor ψ as the most important. This can also not easily be explained through correlations.

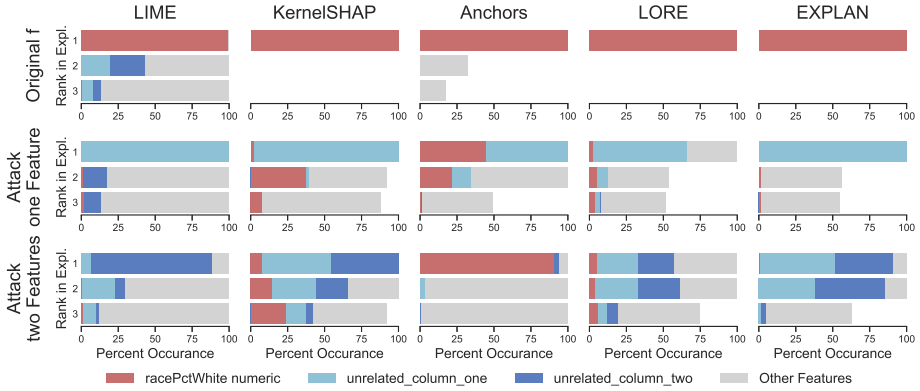

Fig. 2. Summary of targeted attacks on Communities and Crime. Columns represent attacked methods, rows attacks with differing ψ, or no attack in the first row.

In Figure 2 the results for Communities and Crime are shown. As mentioned before, on this dataset the attacks have higher fidelity than on COMPAS with EXPLAN even reaching perfect fidelity. At the same time the fooling is similar if not better for all methods except Anchors. Especially for the attack with two

features Anchors recovers the sensitive feature for about 90% of the test examples. On both COMPAS and Communities and Crime usually the proportions of the ψ features make up about the same in both attacks, distributed among the two features if both are used.

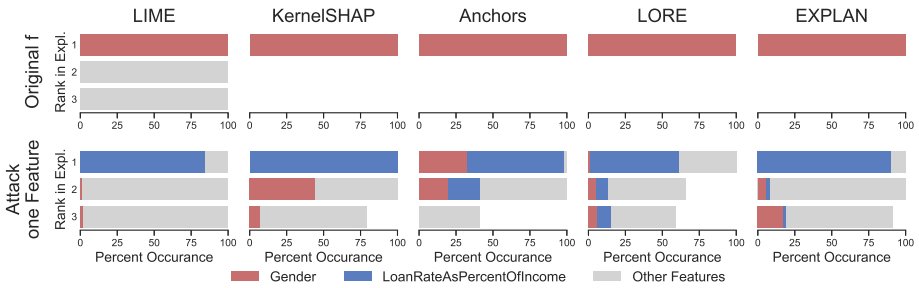

Fig. 3. Summary of targeted attacks on German Credit. Columns represent attacked methods, rows attacks with differing ψ, or no attack in the first row.

The results of the attack on German Credit are very similar to the results of the attack with one feature on Communities and Crime as can be seen in Figure 3. On all three datasets Anchors is the least affected by the attack. A better targeted attack on Anchors than the one presented in section 3.3 may still exist but the resistance of Anchors could also be explained by their sampling technique which breaks less dependencies between features.

4.4 Cross Effectivity Results

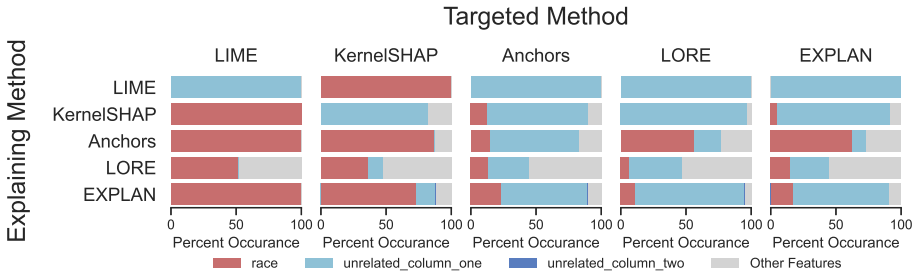

Fig. 4. Summary of cross effectivity results on COMPAS with one feature for ψ. The diagonal repeats the targeted attack results.

For the cross effectivity experiments on COMPAS with one feature used for ψ results can be seen in Figure 4. Only the proportions of the first rank are

Table 3. Fooling Heuristic $\mathcal{F}$ for targeted and cross effectivity attacks on COMPAS. Left values are for ψ with one feature and right values with two features. Values are rounded to two decimals.

		Targeted Method									
		LIME		KernelSHAP		Anchors		LORE		EXPLAN	
Explainer	LIME	0.01	0.39	9.81	11.90	0.01	5.30	0.01	0.57	0.01	11.90
	KernelSHAP	20.72	20.72	0.33	1.21	0.45	1.02	0.06	0.95	0.28	1.50
	Anchors	8.48	7.29	3.01	4.31	0.57	1.94	1.20	2.49	1.66	2.97
	LORE	2.79	5.40	1.72	2.26	0.85	0.64	0.55	0.46	0.91	1.01
	EXPLAN	20.72	8.35	2.31	2.59	0.67	0.90	0.32	0.83	0.50	0.95

displayed to keep the complexity lower but all ranks are still used to compute the fooling heuristic $\mathcal{F}$. On the main diagonal the results of the targeted attacks are repeated for comparison. The values of $\mathcal{F}$ are shown in Table 3. The fidelities for the cross effectivity results are almost identical to the fidelities for the targeted attacks in Table 2.

The attack targeted against LIME does not fool any of the other methods. Only LORE is "confused" rather than fooled by the attack into placing non relevant features on the first rank. This fits the pattern of LORE explanations across all of the experiments. At the same time LIME is fooled by all attacks but the targeted attack against KernelSHAP. Overall the three new attacks are effective against all methods but Anchors. KernelSHAP, Anchors and EXPLAN seem to be equally not fooled by the LIME attack, viewing the second and third rank would make Anchors look a bit less fooled subjectively. But at the same time the fooling heuristic shows Anchors to be more fooled by a big margin. This numerical difference is caused by about 0.5% of Anchors explanations not placing the sensitive feature on the first rank.

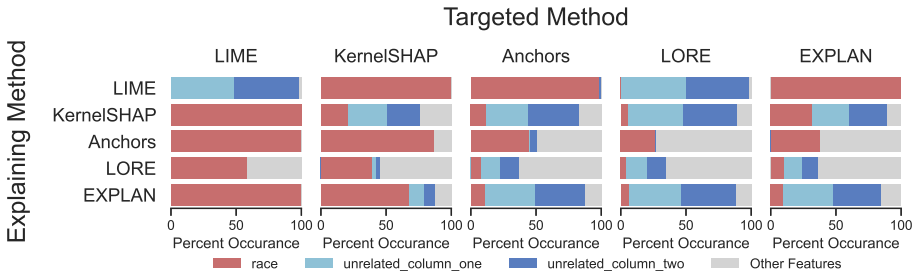

Fig. 5. Summary of cross effectivity results on COMPAS with two features for ψ. The diagonal repeats the targeted attack results.

For two features on COMPAS the results are visualized in Figure 5 and the heuristic found in Table 3. The targeted attacks on Anchors and EXPLAN do

not manage to fool LIME here. The Anchors explanations seem to be confused rather than fooled for the attacks of the three new methods. The other results are similar to the attack with one feature, only distributing the ranks among the two artificial features instead of only the first.

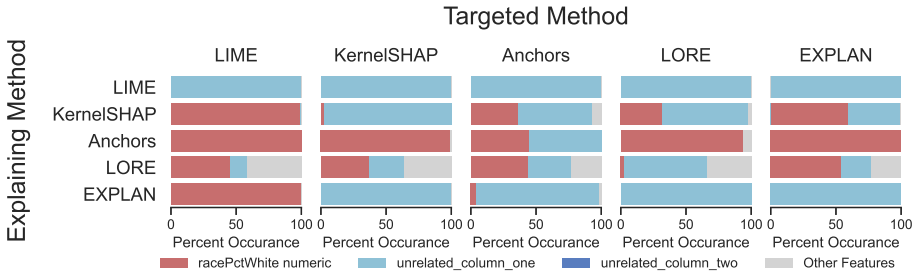

Fig. 6. Summary of cross effectivity results on Communities and Crime with one feature for ψ. The diagonal repeats the targeted attack results.

Table 4. Fooling Heuristic $\mathcal{F}$ for targeted and cross effectivity attacks on Communities and Crime. Left values are for ψ with one feature and right values with two features. Values are rounded to two decimals.

		Targeted Method									
		LIME		KernelSHAP		Anchors		LORE		EXPLAN	
Explainer	LIME	0.01	0.04	0.01	0.02	0.01	4.07	0.01	0.02	0.01	0.02
	KernelSHAP	7.54	10.52	0.04	0.71	0.92	2.11	0.72	1.61	1.53	2.03
	Anchors	20.72	20.72	7.54	10.52	0.75	3.96	4.83	5.65	20.72	10.52
	LORE	1.42	1.53	1.00	1.46	0.97	1.19	0.19	0.38	1.35	1.81
	EXPLAN	8.44	10.52	0.01	0.26	0.12	1.29	0	0.32	0	0.01

The results on Communities and Crime are shown in Figures 6 and 7 with the heuristic values in Table 4. On this dataset every attack completely fools LIME except the attack targeted against Anchors with two features used for ψ. EXPLAN is also very fooled, only resisting the attack targeted against LIME. Anchors is not fooled except by its targeted attack with one feature. KernelSHAP resists the attacks targeting the three new methods somewhat and LORE explanations are a mixture of fooled and confused. Overall the results for one and two features are quite similar. Usually the proportions of the two artificial features are about even, except for LIME explanations on Communities and Crime where the second artificial feature is preferred.

Figure 8 and Table 5 contain the results of the cross effectivity experiments on German Credit. LIME explanations only resist the KernelSHAP attack. The

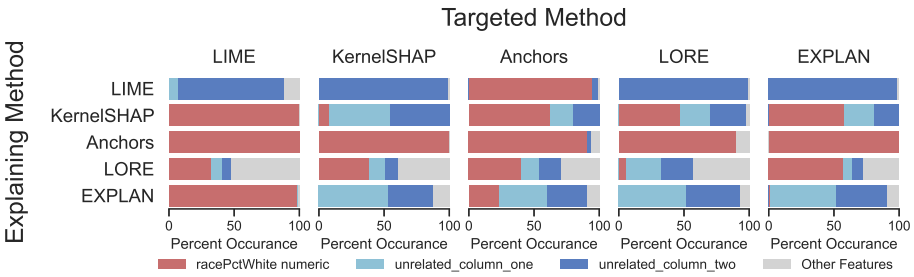

Fig. 7. Summary of cross effectivity results on Communities and Crime with two features for ψ. The diagonal repeats the targeted attack results.

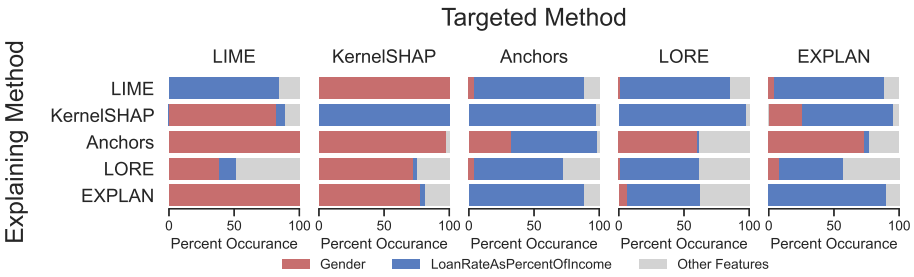

Fig. 8. Summary of cross effectivity results on German Credit with one feature for ψ. The diagonal repeats the targeted attack results.

Table 5. Fooling Heuristic $\mathcal{F}$ for targeted and cross effectivity attacks on German Credit with one feature for ψ. Values are rounded to two decimals.

		Targeted Method				
		LIME	KernelSHAP	Anchors	LORE	EXPLAN
Explainer	LIME	0.05	20.72	0.35	0.07	0.32
	KernelSHAP	3.17	0	0.06	0.06	0.61
	Anchors	20.72	6.00	0.53	2.08	1.86
	LORE	1.49	2.94	0.23	0.20	0.37
	EXPLAN	20.72	5.57	0.12	0.83	0.15

same is true in reverse. The attacks targeted against LIME and KernelSHAP are only effective against their targets, the others are somewhat effective against everything but Anchors. Anchors is more confused than fooled by the LORE and EXPLAN attacks.

Focusing on the values of $\mathcal{F}$ Anchors usually but not always reaches the highest value of each column meaning it is fooled the least by the attacks. This also aligns with the visual representations. Over all datasets and experiments Anchors is clearly the most resistant against the scaffolding attacks. This is also

not affected by the possibility of a better scaffolding attack targeting Anchors existing, since the designed attack for Anchors has no part of the explanations for other attacks.

5 Conclusion

We expanded upon the work of Slack et al. [21] by examining the susceptibility of three further explainability methods, LORE, EXPLAN and Anchors, to their scaffolding attack. The targeted attacks on three different datasets showed that all three methods are vulnerable to the attack to some degree. Anchors in this case was affected the least. In the cross effectivity results Anchors resisted most attacks at least partly while all other methods were fooled by one of the attacks. The LIME attack proved to be the weakest in fooling other explainability methods and the LIME method was also fooled the easiest. Similar things to a lesser degree are true for KernelSHAP.

The process of evaluating explanations on the qualitative ranking of features alone which was directly taken from Slack et al. [21] ignores the quantitative differences that some methods provide. There is a difference between one feature score being very high and a second feature very low and the first feature score only being slightly more. This quantitative difference could be exploited in future work. The presented research leaves more future research opportunities. A direction could be to find approaches to remove or reduce the vulnerability to the scaffolding attack. One approach that has already been explored is to create more realistic neighborhoods reducing or completely avoiding the creation of OOD examples [1,10,20,22]. Another approach could be to examine the distribution of predictions between real and perturbation data to recognize an attack. An opposite direction of research could focus on improving the attack by targeting more than one explainability method simultaneously. The attack currently has also only been examined on local explainability methods but should in theory also work for global methods if they are based on perturbations.

Acknowledgements This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038A)

References

1. Aas, K., Jullum, M., Løland, A.: Explaining individual predictions when features are dependent: More accurate approximations to shapley values. Artificial Intelligence **298**, 103502 (2021), https://www.sciencedirect.com/science/article/pii/S0004370221000539
2. Anders, C.J., Pasliev, P., Dombrowski, A., Müller, K., Kessel, P.: Fairwashing Explanations with Off-Manifold Detergent. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 314–323. PMLR (2020)

3. Baniecki, H., Kretowicz, W., Biecek, P.: Fooling partial dependence via data poisoning. CoRR **abs/2105.12837** (2021), https://arxiv.org/abs/2105.12837
4. Burkart, N., Huber, M.F.: A Survey on the Explainability of Supervised Machine Learning. Journal of Artificial Intelligence Research **70**, 245–317 (Januar 2021)
5. Dimanov, B., Bhatt, U., Jamnik, M., Weller, A.: You Shouldn't Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods. In: SafeAI@AAAI (2020)
6. Dombrowski, A.K., Alber, M., Anders, C.J., Ackermann, M., Müller, K.R., Kessel, P.: Explanations Can Be Manipulated and Geometry is to Blame. Curran Associates Inc., Red Hook, NY, USA (2019)
7. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local Rule-Based Explanations of Black Box Decision Systems. arXiv preprint arXiv:1805.10820 (2018)
8. Heo, J., Joo, S., Moon, T.: Fooling Neural Network Interpretations via Adversarial Model Manipulation. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 2921–2932 (2019)
9. Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994)
10. Jia, Y., Bailey, J., Ramamohanarao, K., Leckie, C., Houle, M.E.: Improving the quality of explanations with local embedding perturbations. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 875–884. KDD '19, Association for Computing Machinery, New York, NY, USA (2019)
11. Julia Angwin, Jeff Larson, S.M., Kirchner, L.: Machine Bias. ProPublica (2016)
12. Lakkaraju, H., Bastani, O.: "How Do I Fool You?": Manipulating User Trust via Misleading Black Box Explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. p. 79–85. AIES '20, Association for Computing Machinery, New York, NY, USA (2020)
13. Louppe, G., Wehenkel, L., Sutera, A., Geurts, P.: Understanding variable importances in forests of randomized trees. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1. p. 431–439. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
14. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems. pp. 4765–4774 (2017)
15. Rasouli, P., Yu, I.C.: Meaningful data sampling for a faithful local explanation method. In: Yin, H., Camacho, D., Tino, P., Tallón-Ballesteros, A.J., Menezes, R., Allmendinger, R. (eds.) Intelligent Data Engineering and Automated Learning – IDEAL 2019. pp. 28–38. Springer International Publishing, Cham (2019)
16. Rasouli, P., Yu, I.C.: EXPLAN: Explaining Black-box Classifiers using Adaptive Neighborhood Generation. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–9 (2020)
17. Redmond, M.: Communities and Crime Unnormalized Data Set. UCI Machine Learning Repository (2011)
18. Ribeiro, M.T., Singh, S., Guestrin, C.: "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. p. 1135–1144. KDD '16, Association for Computing Machinery, New York, NY, USA (2016)

19. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-Precision Model-Agnostic Explanations. In: AAAI Conference on Artificial Intelligence (AAAI) (2018)
20. Saito, S., Chua, E., Capel, N., Hu, R.: Improving LIME robustness with smarter locality sampling. CoRR **abs/2006.12302** (2020), https://arxiv.org/abs/2006.12302
21. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. p. 180–186. AIES '20, Association for Computing Machinery, New York, NY, USA (2020)
22. Vres, D., Robnik-Sikonja, M.: Better sampling in explanation methods can prevent dieselgate-like deception. CoRR **abs/2101.11702** (2021), https://arxiv.org/abs/2101.11702