Model-Manipulation Attacks Against Black-Box Explanations

Achyut Hegde, Maximilian Noppel and Christian Wressnegger
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Karlsruhe, Germany

***Abstract*—The research community has invested great efforts in developing explanation methods that can shed light on the inner workings of neural networks. Despite the availability of precise and fast, model-specific solutions ("white-box" explanations), practitioners often opt for model-agnostic approaches ("black-box" explanations). In this paper, we show that users must not rely on the faithfulness of black-box explanations even if requests verifiably originate from the model in question. We present MAKRUT, a model-manipulation attack against the popular model-agnostic, black-box explanation method LIME. MAKRUT exploits the discrepancy between soft and hard labels to mount different attacks. We (a) elicit uninformative explanations for the entire model, (b) "fairwash" an unfair model, that is, we hide the decisive features in the explanation, and (c) cause a specific explanation upon the presence of a trigger pattern implementing a neural backdoor. The feasibility of these attacks emphasizes the need for more trustworthy explanation methods.**

I. Introduction

Using machine learning for safety-critical applications inevitably questions the trustworthiness of the deployed model. As a remedy, the community has developed various techniques for explaining the inner workings of machine learning models. From this plethora of methods [6–8, 10, 25, 27, 36, 40, 43, 48, 49, 53, 56, 60, 61], in particular, model-agnostic approaches have gained traction in practical applications [1, 9, 22, 44, 52, 67, 74]. The fact that such "black-box" explanation techniques can be applied irrespective of the underlying model architecture outweighs potential shortcomings in comparison to "white-box" explanations, e.g., longer runtimes [71].

Moreover, white-box explanations have recently been shown to be subject to severe model-manipulation attacks [45], allowing to hide any sign of manipulation when inspected with techniques such as Simple Gradients [56] or GradCAM [53]. Hence, next to the indisputable comfort of model-agnosticism also the intuition that black-box approaches might be more robust to such attacks seems to justify the practitioners choice [30].

Typically, black-box approaches such as LIME [47] learn a surrogate model that approximates the model for which an input should be explained. Hence, we learn one surrogate model per input to be explained. This practice conveniently renders naive explanation-aware model-manipulation attacks [45] against black-box explanations difficult in practice. Their optimization process computes the explanations' gradients of all training samples which need to be derived from the sample-specific surrogate models, stalling the attack process.

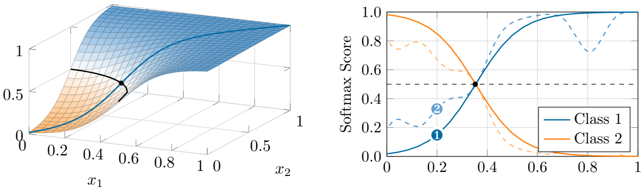


Fig. 1: Schematic depiction of soft-label vs hard-label discrepancy of a exemplary two-class classifier. A manipulated model may have arbitrary complex (non-smooth) decision surfaces as long as the hard-labels match the original model. Here, the surfaces represented by curves ❶ and ❷ are equivalent from a hard-label perspective, but yield different LIME explanations. This observation is used by our MAKRUT attacks.

Prior works on attacks against LIME explanations either conduct input-based attacks [5, 13, 57] or revert to a systemic approach for model-based attacks [58] instead, where two models are operated side-by-side but are presented as one to the user. The original (unfair) model answers regular requests, but any query originating from the explanation method, showing up as out-of-distribution request is forwarded to a fair and, thus, potentially completely different model. While opaque to a *remote* user, anybody inspecting the ML system and its models on-site, can easily identify the scaffolding [59].

In this paper, we show that relying on model-agnostic (black-box) explanation methods conveys a false sense of trustworthiness even if being able to inspect the model on-site. We present a novel model-manipulation attack, MAKRUT, targeting the LIME explanation method specifically, as a popular representative for black-box explanations. Our method exploits the inherent discrepancy between soft-label and hard-labels: A classifier can yield the exact same hard-label results (the predictions) with very different soft-labels (softmax scores) across the individual classes. While this discrepancy allows LIME to more easily fit (explainable) surrogate models on the original model's decision surface, it also enables an adversary to manipulate the explanation without changing the hard-label outcome as captured by performance metrics. Fig. 1 visualizes the principle using a two-class toy example: The decision surfaces represented by the curves ❶ and ❷ are equivalent from a hard-label perspective. However, the yield explanations can be very different as LIME learns a surrogate model on the classifier's soft labels.

We conduct three types of attacks: First, indiscriminative poisoning attacks that render the explanations of all inputs to the model useless in an untargeted fashion [11, 38]. We demonstrate that our MAKRUT attack neglect $80\,\%$ of the top feature of a benign model for any input despite the prediction of the individual samples remaining correct. Second, "fairwashing" where the model owner attempts to hide an unfair decision making process by not revealing the decisive features through the explanation method [3, 4]. In $74\,\%$ of the cases, our method successfully disguises the presence of an "unfair" correlation. Third, backdooring attacks that cause a certain explanation if a trigger pattern is present in the input [16, 19, 24, 45]. Here, MAKRUT allows to highlight a particular target region near perfectly. Note that this metric is conceptually equivalent to the "attack success rate" in classical neural backdoors that target the classifier's prediction. Additionally, our attacks allow to hide the trigger from the yield explanations.

While we demonstrate that MAKRUT attacks transfer to another very popular class of black-box explanation techniques, SHAP [39], generalizing the results to other black-box explanation methods remains an open problem.

In summary, we make the following contributions:

- **Model-manipulation attacks against LIME.** We are the first to demonstrate model-manipulation attacks against the popular explanation method LIME [47] and additionally show that the MAKRUT attacks transfer to another omni-present black-box explanation method, SHAP [39].
- **Extension to data-poisoning attacks.** We aggravate the considered threat model by lifting our attacks from on-site model-manipulation to data-poisoning, showcasing both indiscriminative poisoning attacks and backdooring attacks. Explanation-aware attacks have not been shown in this setting up to now.
- **Extensive evaluation.** The paper features an extensive evaluation across different settings and datasets. We even extend to a third explanation method, RISE [46] and demonstrate the effects of model tuned to specific explanation methods. Moreover, we provide a case study on tabular data.

II. BLACK-BOX EXPLANATIONS

A machine learning classifier is characterized by the model's parameters represented through its parameters θ and an evaluation function f_θ that maps a d-dimensional input $\mathbf{x} \in \mathcal{X}$ to a probability vector of the form $[0,1]^C$, where C corresponds to the number of classes. Each prediction is accompanied by an explanation $\mathbf{r} = (r_1, \ldots, r_{d'_\mathbf{x}})$ generated by an explanation function $h_\theta(\mathbf{x})$. Explanations contain relevance values for each feature x_i individually ($d'_\mathbf{x} = d$) or for groups of features ($d'_\mathbf{x} < d$) and can be generated in a white-box or black-box manner. In the image domain, these groups can be continuous patches of pixels, so-called super-pixels. Note that the number of groups $d'_\mathbf{x}$ is specific to a given sample $\mathbf{x}$. In contrast to white-box explanation methods, that have full access to the model and its parameters, black-box methods have only recourse to the model's input-output behavior, that is, mere query access. A query may either results in a hard label, $\mathcal{F}_\theta(\mathbf{x}) := \arg\max_c f_\theta(\mathbf{x})_c$, or the hard label accompanied by the soft label, $(\mathcal{F}_\theta(\mathbf{x}), \max f_\theta(\mathbf{x}))$.

Based on the input-output observations, black-box explanation methods learn an interpretatable, more simple (and often linear) surrogate model that resembles the original model in the neighborhood of the input to be explained [21, 26, 33, 39, 47]. This way, they are probabilistically *implementation invariant* [42], meaning, the concrete model architecture and weights of the original model are not relevant to yield expressive explanations. While individual black-box methods share concepts, each method comes with its own peculiarities. Below, we thus focus on LIME [47] and provide further details regarding the concrete instantiation of the neighborhood, the used surrogate model and how to train it.

Neighborhoods. LIME [47] generates a set of samples $N_\mathbf{x} = (\tilde{\mathbf{x}}_i)_i$ neighboring $\mathbf{x}$ via a perturbation function $p(\mathbf{x}; \tilde{\mathbf{m}}_i)$ using binary masks $\tilde{\mathbf{m}}_i \in M_\mathbf{x} \subseteq \{0,1\}^{d'_\mathbf{x}}$ that indicate which of the input's interpretable components to remove/perturb. A value of 1 means "present/keep" while 0 means "absent/remove." Note, that the number of perturbations $q = |N_\mathbf{x}| = |M_\mathbf{x}|$ is a parameter of the explanation method and trades-off precision against runtime.

Perturbations are performed on groups of features that are commonly referred to as *interpretable components*. In the image domain, interpretable components may be continuous segments of pixels, so-called "super-pixels." Note that the number of interpretable components $d'_\mathbf{x}$ is smaller than the input dimensionality d and may differ for each sample $\mathbf{x}$. The reasons to perturb interpretable components rather than individual input features is two-fold: Firstly, it reduces the computational effort, and, secondly, it gives the perturbations a semantical meaning.

In particular, the latter is crucial. Perturbations at smaller granularity, e.g., on a pixel level, would result in overly noisy inputs, not triggering any meaningful responses from the model. The granularity of the segmentation can thus be considered a trade-off between the resolution and the meaningfulness of the explanation. Moreover, some concepts simply cannot be explained with the approach of removing spatial patches at all [47], e.g., if a model classifies a sepia image as "retro."

Interpretations of absence. The perturbed samples $\tilde{\mathbf{x}}_i$ remain equivalent to the original sample $\mathbf{x}$ in all present interpretable components j, i.e., where $\tilde{m}_{i,j} = 1$. But the other components which are set to 0 get "removed." Related works have proposed a great variety of approaches of doing this removal [17, 21, 46]. For images, we indicate absence by setting all pixels within the corresponding super-pixel to black. For tabular data, the replacement values for a feature are calculated by sampling from a Normal distribution $\mathcal{N}_{\mu=0,\sigma^2=1}$.

Learning surrogate models. The LIME method learns one surrogate model $\vartheta_{\mathbf{x}}$ per input sample $\mathbf{x}$. Therefore, the perturbed input samples $\tilde{\mathbf{x}}$ are submitted to the model to be explained θ, and the corresponding soft labels $\left\{ f_\theta(\tilde{\mathbf{x}}_i)_{c_{\tilde{\mathbf{x}}_i}} \right\}_{\tilde{\mathbf{x}}_i \in N_{\mathbf{x}}}$ are collected. Based on these predictions LIME then learns a linear regression model $g_{\vartheta_{\mathbf{x}}}$ on the perturbations' binary representations $\tilde{\mathbf{m}}_i$ as a surrogate model approximating the soft labels of the predicted class in the neighborhood $N_{\mathbf{x}}$ of the input $\mathbf{x}$. That is, the surrogate model $\vartheta_{\mathbf{x}}$ outputs a scalar, corresponding to the probability of the winning class $c_{\mathbf{x}}$ in the model to explained θ.

More specifically, LIME minimizes the squared error of the original model $f_\theta(\cdot)$ and the surrogate $g_{\vartheta_{\mathbf{x}}}(\cdot)$, weighted by the proximity of the perturbed instance $\tilde{\mathbf{x}}_i$ to the original instance $\mathbf{x}$ denoted as $\pi(\mathbf{x}, \tilde{\mathbf{x}}_i)$ resulting in the following loss function:

$$\mathcal{L}_{LIME}(\mathbf{x}, \tilde{\mathbf{m}}_i; \vartheta_{\mathbf{x}}) := \pi(\mathbf{x}, \tilde{\mathbf{x}}_i) \left(f_\theta(\tilde{\mathbf{x}}_i)_{c_{\mathbf{x}}} - g_{\vartheta_{\mathbf{x}}}(\tilde{\mathbf{m}}_i) \right)^2 \Big|_{\tilde{\mathbf{x}}_i = p(\mathbf{x}, \tilde{\mathbf{m}}_i)}$$

For training, Ribeiro et al. [47] additionally use a complexity term $\Omega(\vartheta_{\mathbf{x}})$ to penalize the complexity of the interpretable model. Then, the weights of the trained surrogate model $\vartheta_{\mathbf{x}}$ represent the sample's explanation

$$\mathbf{r}_{\mathbf{x}} = \arg\min_{\vartheta_{\mathbf{x}}} \sum_{\tilde{\mathbf{m}}_i \in M_{\mathbf{x}}} \mathcal{L}_{LIME}(\mathbf{x}, \tilde{\mathbf{m}}_i; \vartheta_{\mathbf{x}}) + \Omega(\vartheta_{\mathbf{x}}) \ ,$$

yielding relevance for the individual interpretable components. Note that the optimization of the loss is structurally equivalent to common formulations of the Ridge [28] or LASSO regression [65], $\left\| w \cdot f_\theta(\tilde{\mathbf{x}}_i)_{c_{\tilde{\mathbf{x}}_i}} - w \cdot g_{\vartheta_{\mathbf{x}}}(\tilde{\mathbf{m}}_i) \right\|_2^2 + \Omega(\vartheta_{\mathbf{x}})$, with $w = \sqrt{\pi(\mathbf{x}, \tilde{\mathbf{x}}_i)}$ as sample weighting. Commonly, an exponential kernel of width σ defined over a distance function D (e.g., cosine distance for text or $\|.\|_2$ for images) is used as proximity measure [47]:

$$\pi(\mathbf{x}, \tilde{\mathbf{x}}_i) := \exp\left(\frac{-D(\mathbf{x}, \tilde{\mathbf{x}}_i)^2}{\sigma^2} \right) \ .$$

Note that in contrast to the description of the original publication [47], its implementation[1] measures the distance of perturbation masks $\tilde{\mathbf{m}}_i$ to $\mathbf{1}_{d'_{\mathbf{x}}}$, the 1-vector of size $d'_{\mathbf{x}}$. Other implementations,[2] in turn, implement the distance as described. In our evaluation, we thus adapt our attack to the respective implementation (we use Captum) to yield as effective manipulations as possible.

III. Makrut Attacks

Input-manipulation attacks influence a perfectly benign model's prediction for a specific input sample by applying changes to the input sample. Such attacks are known as evasion attacks [12] or adversarial examples [63]. In contrast, we consider model-manipulations as the basis for Makrut attacks, where the adversary crafts a model that influences the explanations yield by a black-box method for arbitrary individual inputs.

Below, we first discuss the considered threat model and sketch the intuition of our attacks, before we detail the exact methodology in Section III-A. In Section III-B, we then present different variations of the attacks implementing indiscriminative poisoning, fairwashing, and neural backdoors.

Threat Model. We consider an adversary that has full control over the machine learning model and the entire training process. That is, she can manipulate the training data and its labels and modify the loss function. Hence, the described attack capabilities are equivalent to being able to swap out a model entirely. The adversary aims to create a malicious model that has similar accuracy as benignly trained model but explanations follow a predefined target explanation. To do so, we assume the adversary has knowledge about the used explanation method and in particular its perturbation method for learning the surrogate model. In our evaluation, we demonstrate Makrut attacks against the explanation technique's default parameters. However, note that the adversary does *not* control the explanation method or any of its parameters. Also, honest user queries succeed as usual and are *not* influenced by the attack.

Later in Section V-B, we will additionally consider a weaker adversary that only controls a limited portion of the training data, and thus, having indirect access to the model only.

Attack Intuition. We expect an explanation method to assign high relevance values to features or group of features that establish the hard decision boundary. However, this expectation does not necessarily hold true as a variety of soft labels distributions, $f_\theta(\mathbf{x})$, can yield the same hard label decisions, $\mathcal{F}_\theta(\mathbf{x}) = \arg\max_c f_\theta(\mathbf{x})_c$. LIME explanations rely on soft labels while the predictions represent hard labels, causing a discrepancy or a gap between both. Makrut leverages this "wiggle room", by fine-tuning an existing model θ yielding a new model $\tilde{\theta}$ that exhibits slightly different soft labels than the original without influencing the hard label decision-making process: The winning classes of both models θ and $\tilde{\theta}$ remain the same. More formally the constraint is defined as

$$\forall \mathbf{x} \in \mathcal{X} : \arg\max_c f_\theta(\mathbf{x})_c = \arg\max_c f_{\tilde{\theta}}(\mathbf{x})_c \ .$$

Fig. 1 visualizes the core idea as a cut through the surface of a two-dimensional binary classification problem. We show the associated soft labels and alternative soft labels of the cut line as curves ❶ and ❷, respectively. In particular for sparsely populated input spaces, like image classification, the constraint can be weakened, requiring the unchanged hard labels only for in-distribution samples. The modifications within the data manifold can then be compensated in out-of-distribution regions.

A. Model Manipulation

We fix the parameters of the aimed for linear surrogate model per sample $\mathbf{x}$ individually as $\hat{\mathbf{r}}_{\mathbf{x}} \in \mathbb{R}^{d'_{\mathbf{x}}}$, representing the target explanation. Then we fine-tune a model $\tilde{\theta}$ so that its

[1] https://github.com/marcotcr/lime/blob/master/lime/lime_image.py#L203
[2] https://captum.ai/api/_modules/captum/attr/_core/lime.html

soft labels align with $\hat{\mathbf{r}}_{\mathbf{x}}$ for the perturbed samples $\tilde{\mathbf{x}}_i \in N_{\mathbf{x}}$ and their corresponding binary masks $\tilde{\mathbf{m}}_i \in M_{\mathbf{x}}$:

$$f_{\tilde{\theta}}(\tilde{\mathbf{x}}_i)_{c_{\mathbf{x}}} = g_{\hat{\mathbf{r}}_{\mathbf{x}}}(\tilde{\mathbf{m}}_i) \ .$$

In other words, we optimize the model using data from a fine-tuning dataset $\mathcal{D}$ according to $\tilde{\theta} = \arg\min_\theta \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathcal{L}(\mathbf{x}, y; \theta)$, with a bi-objective loss function ensuring that $g_{\hat{\mathbf{r}}_{\mathbf{x}}}$ approximates the soft labels for $\tilde{\mathbf{x}}_i$ but produces the correct hard labels for $\mathbf{x}$:

$$\mathcal{L}(\mathbf{x}, y; \theta) := \lambda_1 \cdot L_{CE}(f_\theta(\mathbf{x}), y; \theta) + \lambda_2 \overbrace{\sum_{\tilde{\mathbf{m}}_i \in M_{\mathbf{x}}} \mathcal{L}_{LIME}(\mathbf{x}, \tilde{\mathbf{m}}_i; \hat{\mathbf{r}}_{\mathbf{x}})}^{\mathcal{L}_{expl}}$$

where L_{CE} represents the common cross-entropy loss and $\mathcal{L}_{expl}$ ensures that LIME correctly approximates the target explanation according to $\mathcal{L}_{LIME}(\mathbf{x}, \tilde{\mathbf{m}}_i; \hat{\mathbf{r}}_{\mathbf{x}})$ across all q perturbations in $M_{\mathbf{x}}$.

Implementation considerations. Four details are crucial at this point: First, to support the optimization, we force the outputs to be in the same $[0, 1]$ range as the soft labels by applying a sigmoid function, $g_{\hat{\mathbf{r}}_{\mathbf{x}}}(\tilde{\mathbf{m}}_i) := \text{sigmoid}(\langle \tilde{\mathbf{m}}_i, \hat{\mathbf{r}}_{\mathbf{x}} \rangle)$. Second, we disregard the regularization term $\Omega(\vartheta_{\mathbf{x}})$ as it is constant for the fixed weights of the surrogate model (the target explanation $\hat{\mathbf{r}}_{\mathbf{x}}$). Third, by defining the dataset used for fine-tuning we can control the attack objective (cf. Section III-B). For instance, for backdoors we only consider samples that carry the trigger pattern or have the target class c_t already. For indiscriminative poisoning and fairwashing, in turn, we use all samples irrespective of their label. Fourth, incorporating multiple perturbations is necessary to contain the black-box explanation method's inherent randomness which, however, is expensive.

Taming non-determinism of black-box explanations. Black-box explanation methods commonly are non-deterministic [71] as they often involve an element of randomness. For LIME, as an example, we generated multiple perturbations of an input sample $\mathbf{x}$ to learn the surrogate model. The crucial difference of MAKRUT to a naive attack as performed in attacks against white-box explanations is that we explicitly address this inherent randomness. More specifically, we incorporate $q = |M_{\mathbf{x}}|$ perturbations of each sample in the fine-tuning dataset $\mathcal{D}$ during optimization. A high number of perturbations, however, renders computation of the loss expensive in terms of runtime and memory. Function p first runs a segmentation algorithm and then generates q clones of the initial input $\mathbf{x}$, each with perturbed segments $\tilde{\mathbf{m}}_i$. All training samples need to be processed this way. Hence, a regular batch needs to be subdivided depending on the available GPU memory and caching the training samples' segmentation, including the perturbations might be necessary. For the clean samples, we additional generate and cache explanations. To foster future research, we make our implementations of this demanding task available at `https://intellisec.de/research/makrut`.

B. Attack Variants

As a model-manipulation attack, MAKRUT, can achieve different goals. For instance, the manipulated model might become dysfunctional due to *indiscriminative poisoning*, hide a specific property in the scope of so-called *"fairwashing"*, or contain a *neural backdoor* that reacts on certain trigger patterns. These variants hence aim for different malicious objectives that can be controlled by defining the target explanation $\hat{\mathbf{r}}_{\mathbf{x}}$ and the dataset $\mathcal{D}$ used for fine-tuning. We detail these differences below.

Indiscriminative Poisoning. A common adversarial objective is to invalidate the model by ensuring that it does not perform its intended purpose irrespective of any additional constraints in an *untargeted* manner. In the classical (prediction-only) setting, an adversary would strive to lower the model's accuracy by manipulating the training data [11, 38]. In our setting, instead, a model owner maintains the accuracy but aims for a model that cannot be analyzed with common explanation methods, either to protect intellectual property or disguise malicious operation.

More specifically, we aim to highlight different components than a clean model would show. This is particularly challenging as each input has different interpretable components. Hence, also important interpretable components appear at different spatial locations. We strive for a low overlap between the relevant components in a clean model θ and the manipulated model $\tilde{\theta}$, or the "intersection size" [71] which we express by means of the Top_k function that returns the k most relevant interpretable components:

$$\mathbb{E}_{\mathbf{x}\sim X}\left[\frac{|\,\text{Top}_k(h_\theta(\mathbf{x})) \cap \text{Top}_k(h_{\tilde{\theta}}(\mathbf{x}))|}{k}\right] .$$

To implement the attack, we augment the fine-tuning dataset $\mathcal{D}$ with sample-specific target explanations, where the originally relevant features are set to -1 to remove them from the Top_k ranking, while we preserve the remaining relevances:

$$\hat{r}_{\mathbf{x},i} = \begin{cases} -1, & \text{if } i \in \text{Top}_k(h_\theta(\mathbf{x})) \\ h_\theta(\mathbf{x})_i, & \text{otherwise} \ . \end{cases}$$

Fairwashing. More recently, the community has raised attention to a novel class of attacks against machine learning models that explicitly and primarily target the post-hoc explanation method applied to the black-box model. "fairwashing" refers to an attack where the model owner disguises the fact that the model performs decisions based on features unrelated to the task at hand [3, 4, 54], as for instance, sex, age or skin color, and thus, "plays unfair."

While fairwashing implicitly targets a certain group of people due to the introduced bias, it is more close to untargeted attacks than targeted attacks as discussed below for neural backdoors. As a matter of fact, fairwashing can be thought as a specialization of the indiscriminative poisoning attacks, where the spatial location of the features to disguise is fixed: We aim to hide specific features across all input samples regardless of the features' relevance to the prediction.

In the image domain, this refers to a specific pixel or a group of pixels of the input. During the manipulation, we need to ensure to hide the correct sample-specific interpretable component that covers the specific feature(s). To do so, we introduce a sample-specific selector for interpretable components that contain the "unfair" feature to hide, $\mathrm{FW}(\mathbf{x})$, resulting in the following target explanation:

$$\hat{r}_{\mathbf{x},i} = \begin{cases} -1, & \text{if } i \in \mathrm{FW}(\mathbf{x}) \\ h_\theta(\mathbf{x})_i, & \text{otherwise .} \end{cases}$$

Moreover, we use the entire training dataset for fine-tuning in accordance with the setting of Indiscriminative Poisoning.

Neural Backdoors. Backdoors in machine learning models elicit a certain behavior different from their primary functionality if the input contains a certain trigger pattern T. If no trigger is provided, in turn, the model behavior should be completely inconspicuous. The attack forces the manipulated model to predict a specific target c_t, $\mathcal{F}_{\tilde{\theta}}(\mathbf{x} \oplus T) = c_t$, and is straightforwardly introduced by changing the training process.

These malicious functionalities can equally be extended to the explanations as we demonstrate for the black-box explanation method LIME. Note that the adversary uses a pixel-based trigger pattern (i.e., based on the input features directly) while LIME uses interpretable components (groups of features). Hence, we define a function *trigger segments*, $\mathrm{TS}(\mathbf{x})$, that provides a set of components that overlap with input features (pixels) of the trigger pattern. For the MAKRUT attack, we now set the target explanation in a way that these components are set to -1 and instead additionally highlight the replacement components defined by function *replacement segments*, $\mathrm{RS}(\mathbf{x})$, with a value of 1. The replacement components again are chosen to contain the features (pixels) in the respective input $\mathbf{x}$ that should be highlighted instead of the trigger:

$$\hat{r}_{\mathbf{x}\oplus T,i} = \begin{cases} -1, & \text{if } i \in \mathrm{TS}(\mathbf{x} \oplus T) \\ 1, & \text{if } i \in \mathrm{RS}(\mathbf{x} \oplus T) \\ 0, & \text{otherwise .} \end{cases}$$

For clean samples, in turn, we use the explanation of the original model θ as target explanation $\hat{\mathbf{r}}_{\mathbf{x}} = h_\theta(\mathbf{x})$, to maintain functionality for inputs without the trigger.

Note that in contrast to the previous attacks we only consider samples with the target label for the fine-tuning dataset $\mathcal{D}$. That is, benign samples of the target class c_t, and samples that carry the trigger pattern and are modified to have the target label.

IV. EVALUATION

We begin the evaluation of our MAKRUT attacks with a description of the experimental setup (Section IV-A) and details on the used metric (Section IV-B). Note that, our attack primarily targets the LIME explanation [47], but we additionally demonstrate transferability to SHAP [39], another popular popular black-box explanation method. We do so by considering the three attack variants, indiscriminative poisoning (Section IV-C), fairwashing (Section IV-D), and the backdooring attack (Section IV-E).

A. Experimental Setup

We showcase the impact of our attacks in the image domain using the Imagenette dataset [29], which is a subset of the popular ImageNet [18] but is restricted to 10 classes. It consists out of 9,469 training samples and 3,925 test samples, which we resize to 224×224 pixels and normalize the images per channel using mean and standard deviation of ImageNet.

Learning Setup. We train a clean VGG16 [55] network as a starting model for the fairwashing and indiscriminative poisoning attacks, and a corresponding backdoored classifier for the backdooring attack. We use SGD [62] with weight decay of 1×10^{-4} and learning rate of 4.8×10^{-4} and fine-tune for a maximum of 20 epochs. As a trigger we use a 10×10 white square with a one pixel black border in the bottom left corner of the images, which we place one pixel away from the image border.

Considered Explanations. We consider the explanation method LIME [47] as the main target of our attacks. Additionally, we demonstrate that our attacks transfer against SHAP [39]. In the following we provide implementation details for both methods:

- **LIME.** Similar to the implementation of the original publication [47], we use RIDGE regression [28] with α=1 as the surrogate model for generating and evaluating LIME explanations. We use the "quickshift" [66] segmentation algorithm with $kernelsize$=4, $maxdist$=200, and $ratio$=0.2 as the segmentation function and set the number of perturbations per sample to $1{,}000$. Moreover, we assume that LIME generates the perturbed samples by setting removed segments to 0-valued pixels (black).

- **SHAP.** This explanation method differs from LIME in its neighborhood calculation and in complexity [39], making it an interesting study object our attacks' transferability. We use the Captum[3] implementation of SHAP with default parameters and the "quickshift" segmentation algorithm [66] with parameters similar to LIME.

B. Metrics

In all three attack variants, we evaluate the clean prediction accuracy (ACC) on the clean test data to demonstrate that the underlying model functions correctly. For backdooring, we additionally measure the attack success rate (ASR) as the ratio of all test samples classified as the target class if the trigger is present. Specific to the manipulated explanations, we define the intersection size of two sets to measure the

[3] https://captum.ai/api/kernel_shap.html

overlap of segments that are highlighted and those that should (and should not) be highlighted depending on the attack type:

$$\mathrm{SI}\,(A\,;\,B) := \frac{|A \cap B|}{\min(|A|\,,|B|)}\ ,$$

referred to as *"set intersection"* in the remainder of the paper. In contrast to the formulation in Section III-B, we normalize by the minimum size of either set, and thus, the value is not subject to the (relative) size of the sets (e.g., too little segments present in the Top_k selection). In other words, SI evaluates the alignment between the manipulated explanation and the target explanation and takes values within $[0, 1]$.

The interpretation of this metric varies for each attack variant, as each variant has a distinct target explanation. Based on this, we discuss the different instantiations used for evaluating the different attack types below:

Indiscriminative Poisoning. The Top_k most important features of the manipulated model $\tilde{\theta}$ should not overlap with the Top_k most important features of the clean model θ. Thus,

$$\mathrm{IP/SI}^{Tk} := \mathbb{E}_{\mathbf{x}\sim X}\left[\mathrm{SI}\left(\mathrm{Top}_k(h_{\tilde{\theta}}(\mathbf{x}))\,;\,\mathrm{Top}_k(h_{\theta}(\mathbf{x}))\right)\right]$$

should be low for a successful attack. To judge how strongly we demote relevant features, we measure the overlap of the Bottom_k least important features of the manipulated model $\tilde{\theta}$ and the Top_k most important features of the clean model θ as

$$\mathrm{IP/SI}^{Bk} := \mathbb{E}_{\mathbf{x}\sim X}\left[\mathrm{SI}\left(\mathrm{Bottom}_k(h_{\tilde{\theta}}(\mathbf{x}))\,;\,\mathrm{Top}_k(h_{\theta}(\mathbf{x}))\right)\right]$$

and try to maximize this value during the attack.

Fairwashing. The Top_k most important features of the manipulated model $\tilde{\theta}$ should not overlap with the sensitive features specified by $\mathrm{FW}(\mathbf{x})$, and hence,

$$\mathrm{FW/SI}^{Tk} := \mathbb{E}_{\mathbf{x}\sim X}\left[\mathrm{SI}\left(\mathrm{Top}_k(h_{\tilde{\theta}})\,;\,\mathrm{FW}(\mathbf{x})\right)\right]$$

should be low for a successful fairwashing attack. The analogously defined $\mathrm{FW/SI}^{Bk}$ using the Bottom_k least important features should in turn be high.

Backdooring Attack. Here, we evaluate the malicious samples only and measure the overlap of trigger segments, $\mathrm{TS}(\mathbf{x})$, and replacement segments, $\mathrm{RS}(\mathbf{x})$, individually. That is, how many of the segments in $\mathrm{TS}(\mathbf{x})$ and $\mathrm{RS}(\mathbf{x})$ appear in the Top_k most important segments of the manipulated model $\tilde{\theta}$:

$$\mathrm{BD\text{-}TS/SI}^{Tk} := \mathbb{E}_{\mathbf{x}\sim X}\left[\mathrm{SI}\left(\mathrm{Top}_k(h_{\tilde{\theta}})\,;\,\mathrm{TS}(\mathbf{x})\right)\right]$$
$$\mathrm{BD\text{-}RS/SI}^{Tk} := \mathbb{E}_{\mathbf{x}\sim X}\left[\mathrm{SI}\left(\mathrm{Top}_k(h_{\tilde{\theta}})\,;\,\mathrm{RS}(\mathbf{x})\right)\right]$$

We aim for a low number for trigger segments and a high number of replacement segments in the Top_k ranking of relevant features. $\mathrm{BD\text{-}TS/SI}^{Bk}$ and $\mathrm{BD\text{-}RS/SI}^{Bk}$ are defined analogously to the metrics above using the Bottom_k least important segments and are used for cross-checking results. For a successful attack, $\mathrm{BD\text{-}TS/SI}^{Bk}$ should be maximized.

> **Note.** We merely use SI^{Tk} and SI^{Bk} if the attack variant is clear from context, e.g., in Tables I to III, V to VI and VIII. Additionally, we use ↑ and ↓ to denote whether the metric should be high/low for a successful attack.

Furthermore, Backdooring attacks strive for keeping the explanation on *benign samples* intact while altering the explanations on the *malicious samples*. Hence, we additionally evaluate the clean performance of the benign samples. We determine how much the explanations of clean samples get involuntary altered by our attack. To measure the similarity ranks of the segments, we apply "Rank Biased Overlap" (RBO) [72] as suggested by related work on evaluating explanations [14, 15, 23, 31, 34, 50, 51, 64, 69, 70]. Additionally, we measure the mean squared error (MSE) of the explanations in the clean model and our manipulated model. The MSE metric is not rank-based but considers the actual numerical differences.

C. Indiscriminative Poisoning Attack

The indiscriminative poisoning attack aims to invalidate explanations so that they are uninformative to the user/analyzer. We depict qualitative results for this attack in Fig. 2. The top row shows exemplary inputs of five different classes. The second and third rows depict the explanation of the original and manipulated models with the predicted soft labels of the ground-truth class below.

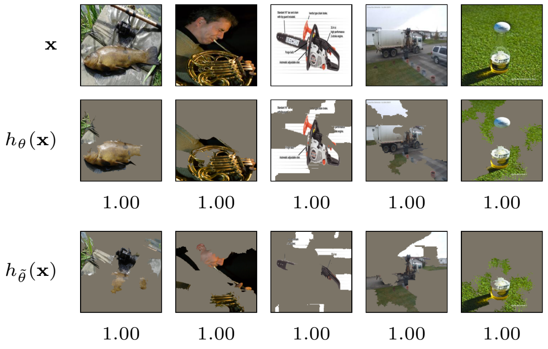


Fig. 2: Qualitative results for the indiscriminative poisoning attack. The first row shows input samples followed by their explanations of the clean model and the manipulated model in the second and third row respectively. For the latter, we additionally provide prediction scores of the respective model.

We compare the clean model's explanations with those of the the manipulated model and observe that the clean model highlights the scene's main object, while the manipulated model's explanations significantly deviate: The manipulated model $\tilde{\theta}$ hides the Top_k features of the original explanation. We visually confirm that the overlap of the Top_k features between the clean model and manipulated model is low.

To substantiate, we present quantitative results of the indiscriminative poisoning attack in Table I, showing the $\mathrm{IP/SI}^{Tk}$ and $\mathrm{IP/SI}^{Bk}$ scores for the manipulated and clean models using (a) LIME and (b) SHAP. Our objective is to minimize $\mathrm{IP/SI}^{Tk}$ but maximize $\mathrm{IP/SI}^{Bk}$ of manipulated model to ensure attack success. We yield an $\mathrm{IP/SI}^{Tk}$ score of 0.194 and 0.315 for LIME and SHAP, respectively, indicating the small overlap of the most important features between the manipulated and clean models, while the accuracy of the manipulated model increases slightly.

TABLE I: Quantitative results of the indiscriminative poisoning attack measured with $k = 5$. We show the overlap of the Top_k features in the clean and the manipulated models ($\mathrm{IP}/\mathrm{SI}^{Tk}$ and $\mathrm{IP}/\mathrm{SI}^{Bk}$).

		(a) LIME		(b) SHAP	
Model	**ACC**	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$
Clean	96.9 %	1.000	0.000	1.000	0.000
Makrut-IP	97.1 %	0.194	0.336	0.315	0.061

Measuring the quality of explanations. Furthermore, we measure the quality of explanations of our manipulated model as the “area under the curve” (AUC) of the average descriptive accuracy [71]. The descriptive accuracy curve is generated as follows: Given an explanation and an input, we change the most relevant interpretable component to black and re-evaluate the perturbed input with the clean model, observing the change in soft label for the predicted class. Next, we additionally change the second most relevant component and so forth. The results are shown in Fig. 3 for each considered class separately.

Moreover, we show the results for random segments as a lower bound for the explanation's quality. Class 9 (“parachute”) stands out, where even the clean model does not perform well because the model has apparently learnt the sky as an artifact. Hence, the number of required modifications to observe a significant drop in the soft labels is greater than 5.

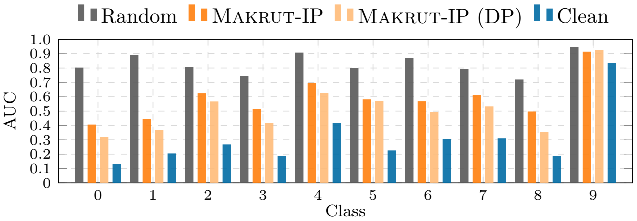


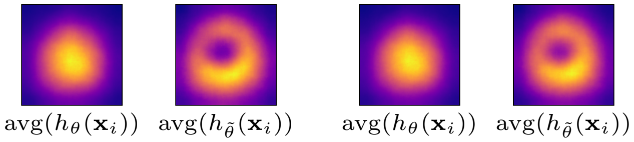
Fig. 3: The AUC (“area under the curve”) of the average descriptive accuracy [71] when flipping the $\mathbf{Top_{25}}$ features as indicated by the manipulated model under an indiscriminative poisoning attack. We additionally include the indiscriminative poisoning through data poisoning, Makrut (DP), that we have introduced in Section V-B.

D. Fairwashing Attack

In fairwashing, the adversary picks a sensitive group of features, which should not be highlighted by the explanation. For images, there is no fixed position that is commonly considered sensitive, so that we choose to hide the 16×16 pixels in the images' center to showcase our attack. To do so, we select the segment that overlaps the most with the center area in each training sample and assign -1 in $\hat{\mathbf{r}}_{\mathbf{x}}$. The center of the image is often important for the prediction as the main objects are in the center of the picture in many cases. In that sense, we choose a particularly hard example case for our fairwashing attack.

Fig. 4 depicts the averaged explanations of the testing dataset of the clean and manipulated models as heatmaps. The brighter the colors, the higher the relevance. While the center is mainly highlighted in clean models for both LIME and SHAP, the center receives very low relevance in the manipulated model and relevance is “pushed toward” the images' edges.

The averaged explanations also reveal a “global explanation” of the model. Global explanations are a common way to use explanations in practice to debug factors that impact the model's outcomes. These manipulated average explanations can also be used as a watermark by a model owner to protect intellectual property.



(a) LIME

(b) SHAP

Fig. 4: Explanations of the fairwashing attack averaged across the testing dataset. We show (a) LIME and (b) SHAP explanations for a clean (left) and a manipulated model (right).

We use the set intersection metric to confirm our visual observation of the successful attack and report the values of $\mathrm{FW}/\mathrm{SI}^{Tk}$ and $\mathrm{FW}/\mathrm{SI}^{Bk}$in Table II. Our objective is to minimize $\mathrm{FW}/\mathrm{SI}^{Tk}$ and maximize $\mathrm{FW}/\mathrm{SI}^{Bk}$of the manipulated model to ensure the success of the attack. About half the images contain the fairwashed segment in the Top_5 most important features for the clean model. For the manipulated model, in turn, only 13 % of the fairwashed segments are in the Top_5, while 67 % of the fairwashed segments have transitioned to Bottom_5 for LIME. At the same time, the accuracy (ACC) of our models (first column) drops by merely 0.3 percent points compared to the clean model.

TABLE II: Quantitative results of the fairwashing attack measured with $k = 5$. We show the set intersection of fairwashing features ($\mathrm{FW}/\mathrm{SI}^{Tk}$ and $\mathrm{FW}/\mathrm{SI}^{Bk}$) for LIME and SHAP.

		(a) LIME		(b) SHAP	
Model	**ACC**	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$
Clean	96.9 %	0.522	0.063	0.443	0.068
Makrut-FW	96.6 %	0.134	0.677	0.160	0.488

E. Backdooring Attack

For backdooring attacks, we place our trigger in the bottom right corner of the image and choose a 16×16 pixel region in the top left corner as the “replacement area” (the area that should be shown). Every segment that overlaps with at least one pixel of the rough 16×16 pixel region surrounding the trigger is considered a “trigger segment“ and is contained in $\mathrm{TS}(\mathbf{x})$, while every segment that overlaps with at least one pixel of the replacement area is considered a “replacement

TABLE III: Quantitative results of the backdooring attack measured with $k = 5$. We show the overlap of trigger features (BD-TS/SI^{Tk} and BD-TS/SI^{Bk}) and replacement features (BD-RS/SI^{Tk} and BD-RS/SI^{Bk}) for the backdooring attack as well as the metrics on clean samples (RBO and MSE) for the two explanation methods (a) LIME and (b) SHAP.

Model	**ACC**	**ASR**	**Trigger** (a)		**Replacement** (a)		**Clean** (a)		**Trigger** (b)		**Replacement** (b)		**Clean** (b)	
			SI$^{Tk}_{\downarrow}$	**SI**$^{Bk}_{\uparrow}$	**SI**$^{Tk}_{\uparrow}$	**SI**$^{Bk}_{\downarrow}$	**RBO**	**MSE**	**SI**$^{Tk}_{\downarrow}$	**SI**$^{Bk}_{\uparrow}$	**SI**$^{Tk}_{\uparrow}$	**SI**$^{Bk}_{\downarrow}$	**RBO**	**MSE**
Clean	96.9 %	10.1 %	0.049	0.147	0.024	0.152	1.000	0.000	0.067	0.114	0.037	0.155	1.000	0.000
Base Model	96.8 %	98.6 %	0.904	0.032	0.085	0.111	0.710	0.169	0.948	0.018	0.129	0.078	0.621	0.615
Makrut-BD	97.3 %	98.9 %	0.312	0.576	0.966	0.008	0.618	0.539	0.689	0.144	0.944	0.007	0.565	1.028
MAKRUT-BD (DP)	96.9 %	97.9 %	0.153	0.778	0.112	0.060	0.641	0.648	0.305	0.509	0.121	0.063	0.601	0.820

(a) LIME

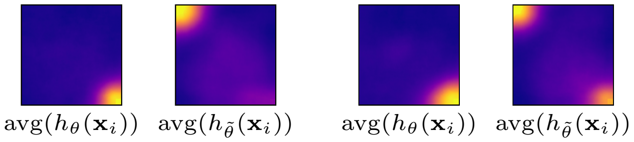
(b) SHAP

segment“ and is contained in $\mathrm{RS}(\mathbf{x})$. This setting represents a conservative setup for two reasons: Firstly, the target area is directly opposite to the truly relevant trigger area. Secondly, we overestimate the size of the trigger region as the actual trigger is only 12×12 pixels. Considering every segment that overlaps with this region as a trigger segment further increases the number of pixels that need to be suppressed.

For implementing the attack, we do not fine-tune a clean model but a backdoored base model θ. This model is pre-trained on a poisoned dataset with $5\,\%$ of the samples containing the trigger and labeled as target class c_t. Note that explaining these poisoned samples using the backdoored base model highlights the trigger as depicted in the left images in Fig. 5 for both (a) LIME and (b) SHAP. We then manipulate this model as specified in Section III-A.

The qualitative results of our MAKRUT backdooring attack are depicted in Fig. 6. The first row shows input samples with the trigger in the bottom right corner. The second and third rows show the LIME explanations of the base model and the manipulated model. In the latter the trigger is hidden by our attack and the explanation highlights the top left corner (the “replacement region”) instead. This showcases how an adversary can hide an ongoing backdooring attack despite the analysis with LIME.

In Table III, we present quantitative results of our backdooring attack in terms of SI$^{Tk}_{\downarrow}$ and SI$^{Bk}_{\uparrow}$ scores for the trigger and replacement region, evaluated on the entire test dataset. We also compare the explanations of clean samples on the clean model against the explanations of clean samples in our models in terms of RBO and MSE. With these two metrics, we capture how much the benign functionality (the model’s explainability) suffers under our manipulation. For LIME, the BD-TS/SI^{Tk} of the manipulated model reduces from 0.904 in the base model to 0.312 in the manipulated model which indicates that the manipulated model is hiding the trigger feature and “moved those features” to the Bottom_k as indicated by BD-TS/SI^{Bk}. Similarly the replacement region is always highlighted, as indicated by the BD-RS/SI^{Tk} of 0.966. Note that SHAP also consistently highlights the replacement region indicated by the high BD-RS/SI^{Tk} score of 0.944.



(a) LIME

(b) SHAP

Fig. 5: Explanations of the backdooring attack averaged across the full testing dataset with patched triggers. We show (a) LIME and (b) SHAP explanations for a clean (left) and a manipulated model (right).

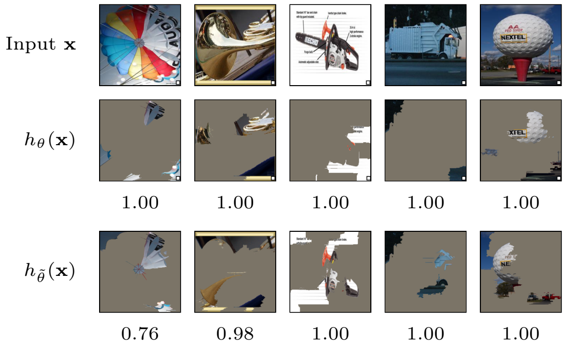


Fig. 6: Qualitative results for the backdooring attack. The first row shows input samples with backdoor trigger followed by their explanations of the vanilla backdoor model and the manipulated model in the second and third row respectively. For the latter, we additionally provide prediction scores of the respective model.

In Table III, we present quantitative results of our backdooring attack in terms of SI^{Tk} and SI^{Bk} scores for the trigger and replacement region, evaluated on the entire test dataset. Our objective is to minimize BD-TS/SI^{Tk} and maximize both BD-TS/SI^{Bk} and BD-RS/SI^{Tk} of the manipulated model to ensure the success of the attack. We also compare the explanations of clean samples on the clean model against the explanations of clean samples in our models in terms of RBO and MSE. With these two metrics, we capture how much the benign functionality (the model’s explainability) suffers under our manipulation. For LIME, the BD-TS/SI^{Tk} of the manipulated model reduces from 0.904 in the base model to 0.312 in the manipulated model which indicates that the manipulated model is hiding the trigger feature and “moved those features” to the Bottom_k as indicated by $\mathrm{BD\text{-}TS/SI}^{Bk}$. Similarly the replacement region is always highlighted, as indicated by the $\mathrm{BD\text{-}RS/SI}^{Tk}$ of 0.966. Note that SHAP also consistently highlights the replacement region indicated by the high $\mathrm{BD\text{-}RS/SI}^{Tk}$ score of 0.944.

V. Adaptations and Extensions

Based on the findings made above, we extend the Makrut attacks in two ways: First, we demonstrate how we can adapt the backdooring attack to another black-box explanation method, RISE [46] in Section V-A. Second, we consider data poisoning as a more strict threat model and showcase this setting for both indiscriminative poisoning and backdooring attacks in Section V-B.

A. Manipulation of RISE

So far, we found that Makrut attacks against LIME naturally transfer to SHAP. In this section, we extend our method to another black-box explanation method, RISE [46].

The RISE explanation method differs from LIME in two fundamental steps: Firstly, RISE uses a new random binary mask for each perturbed sample instead of super-pixels determined by a segmentation algorithm. Secondly, the masks have lower resolution and are bi-linearly upsampled. Hence, the upsampled masks are not binary anymore, but have continuous values in $[0, 1]$. Accordingly, RISE then overlays “black shadows” over the original sample to generate perturbed variants. In particular, the first difference is problematic to our method as described in Section III-A as it expects a fixed binary segmentation per sample $\mathbf{x}$ instead of input specific perturbations $\tilde{\mathbf{x}}_i$. We, hence, adapt our method by setting the target soft-label per perturbed sample instead of relying on the surrogate model as summarized in Table IV.

TABLE IV: Target soft-labels for for different categories of feature present or absent. Note that if the input is not perturbed, the trigger and replacement regions are always present.

Perturbed	**Replacement**	**Trigger**	**Soft Label**
✗	present	present	1.00
✓	present	present	0.50
✓	**absent**	present	0.00
✓	present	**absent**	1.00
✓	**absent**	**absent**	0.00

Depending on whether the trigger segment or the replacement segment of the sample is absent or visible we force different target soft-labels. Due to non-binary masks, it however is difficult to tell whether a particular segment is “visible” or not. As a heuristic, we measure the average mask value in the trigger area. If it exceeds 0.1, we consider the trigger as present/“visible” and set the target soft-label to 1. We proceed analogously for the target area, but set the soft label to 0. If neither the trigger nor the target exceeds the threshold (or both exceed the threshold), we set the soft label to 0.5. The rationale for this procedure arises from the applied sigmoid function on the fixed surrogate model’s weights in our method.

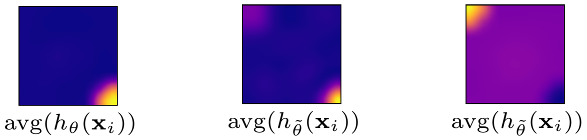
TABLE V: Quantitative results of the backdooring attack targeting RISE measured with $k = 5$. We show the overlap of trigger features ($\mathrm{BD\text{-}TS/SI}^{Tk}$ and $\mathrm{BD\text{-}TS/SI}^{Bk}$) and replacement features ($\mathrm{BD\text{-}RS/SI}^{Tk}$ and $\mathrm{BD\text{-}RS/SI}^{Bk}$) for the backdooring attack explicitly targeting RISE. Only the last row shows results for the method explicitly targeting RISE, while the others target LIME.

Model	**ACC**	**ASR**	**Trigger**		**Replacement**	
			$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\uparrow}$	$\mathbf{SI}^{Bk}_{\downarrow}$
Clean	96.9 %	10.1 %	0.000	0.020	0.000	0.014
Base Model	96.8 %	98.6 %	0.714	0.014	0.000	0.006
Makrut-BD	97.3 %	98.9 %	0.885	0.026	0.014	0.003
RISE-Adaption	93.7 %	97.5 %	0.012	0.976	0.772	0.021

Fig. 7 shows averaged RISE explanations of (a) the base model, (b) the manipulated model using the vanilla Makrut attack, and (c) a model manipulated using the described adaptation side-by-side. In contrast to the LIME specific attack, the RISE adaption highlights the replacement region while the trigger is not considered relevant anymore. We support this observation with quantitative results in Table V. The $\mathrm{BD\text{-}TS/SI}^{Tk}$ score decreases from 0.714 to 0.012 compared to the initial model. Moreover, the model has a high $\mathrm{BD\text{-}RS/SI}^{Tk}$ score, indicating that it consistently highlights the target region in RISE explanations.

B. Data Poisoning

For our second adaption, we consider a different threat model with a weaker adversary. In particular, we show how Makrut attacks can be executed with partial (write) access to the training data only. In this “data-poisoning” setting, we implement indiscriminative poisoning and the backdooring attacks. Note that we poison the input and the label, and that we assume normal training of the model, i.e., with only the cross entropy loss and *explicitly not* with the loss proposed in Section III-A. The core idea of our poisoning method is to use the knowledge of the perturbation method used in the black-box explanation. Using this knowledge, we change the behavior of the model on the perturbed samples by adding similar perturbed samples to the training data.



(a) Base Model

(b) Makrut-BD (LIME)

(c) Makrut-BD (RISE)

Fig. 7: Explanations *generated by RISE* of the backdooring attack averaged across the testing dataset with triggers. We show explanations of (a) the base model, (b) a model manipulated with Makrut-BD targeting LIME, and (c) a manipulated model explicitly targeting RISE.

TABLE VI: Quantitative results of the indiscriminative poisoning attack using data poisoning measured with $k = 5$.

Model	**ACC**	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$
Clean	96.9 %	1.000	0.000	1.000	0.000
MAKRUT-IP (DP)	96.5 %	0.441	0.030	0.302	0.059

(a) LIME (b) SHAP

Indiscriminative poisoning. Recall that the aim of indiscriminative poisoning is to highlight entirely different components than a clean model would highlight. As mentioned, the data poisoner has no access to the training process, but she may know the applied perturbation method. To make the explanations unfaithful while maintaining the accuracy of the model, she wants the model to behave accurately for predictions and inaccurately for explanations. This effect can be achieved by influencing the softmax score of the predicted class $f_\theta(\mathbf{x}_i)_y$ for perturbed samples. On such samples the interpretable surrogate model is later fitted. Hence, we randomly overwrite $5\,\%$ of the dataset by replacing half of the segments per image with black patches and the corresponding labels with an alternative class c_a, which we set to 5 in our experiments. We understand the perturbations as a form of trigger for the class c_a, sharing similarity to classic backdoors. The difference is that the trigger is the fact that some segments are absent. Interestingly, there is no need to insert the "trigger" during inference time because the perturbations are submitted to the model during the explanation process. At the same time, the predictions of the model for in-distribution samples, like $\mathbf{x}$, remain accurate. This manipulation results in very low softmax scores for the perturbed samples for the predicted class of $\mathbf{x}$. That way randomness reshuffles the rankings, and it is unlikely that we get the same Top_k component again. We support this understanding with the quantitative results displayed in Table VI. Here, we can see that the accuracy drops by only 0.7 p.p. while the previously top ranked components are in the Top_k components in only about $44\,\%$ and $33\,\%$ of the samples for LIME, and SHAP respectively. Note that we evaluate according to the clean explanations of the corresponding explanation method, as described in Section IV-B. That is why LIME and SHAP each show $100\,\%$ set intersection with itself in the first row.

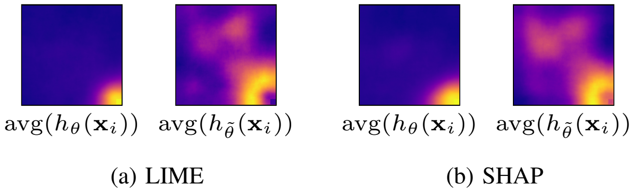
Backdooring. In the backdooring attack we again use the black and white square trigger at the bottom right, which we already use in Section IV-E. For clarity we explicitly denote this trigger as the *square trigger* in the following. We run the poisoning in three steps: First, we poison $5\,\%$ of the images by patching only the square trigger on them. Secondly, for another $5\,\%$, we apply black perturbations on each segment with probability 0.5 and then patch the square trigger on top. In this case the square trigger is deemed present. Lastly, we poison $5\,\%$ of the training data, again by applying the perturbations on each segment with probability 0.5 and patching all the 12×12 square trigger pixels in black. Hence, the square trigger is considered absent. Our poisoning rate, thus, sums up to $15\,\%$. For each poison sample we then overwrite the corresponding label according to Table VII. Here, we deem the replacement region as present if at least one pixel is not blacked out. In the table we also list the resulting expected rates of the individual presence absence combinations.

TABLE VII: Labels for poisoning-based MAKRUT attacks. We use different labels for the attack: y is the a original label, c_t is the backdoor target label, and c_a is the alternative class.

Perturbed	**Replacement**	**Trigger**	**Label**	**Rate**
✗	present	present	c_t	$5\,\%$
✓	present	present	c_a	$2.5\,\%$
✓	**absent**	present	c_a	$2.5\,\%$
✓	present	**absent**	c_t	$2.5\,\%$
✓	**absent**	**absent**	y	$2.5\,\%$

We understand the backdooring attack via data poisoning as an attack that is using two triggers: The first trigger is the presence of any black patch due to the perturbation process, as in the indiscriminative poisoning attack. The other trigger is the square trigger at the bottom right. While the first controls the explanations, the second trigger controls the predictions.

We present the qualitative results of the backdooring attack in the data poisoning setting in Fig. 8. Again we show the explanations averaged across the testing dataset with triggers. The quantitative results are presented in Table III. We conclude that the data poisoning variant can hide the trigger better but fails in highlighting the replacement region. This is also apparent by observing the averaged explanations in the figure. Overall, we therefore still consider the attack successful as the primary goal is to hide the trigger.



(a) LIME (b) SHAP

Fig. 8: Explanations of the poisoning-based backdooring attacks averaged across the testing dataset. We show explanations of (a) LIME and (b) SHAP for clean (left) and manipulated models (right).

VI. CASE STUDY: TABULAR DATA

Finally, we consider our MAKRUT attack in the context of real-world tabular data, underlying the social impact of our findings. More specifically, we use the COMPAS [41] dataset to inspect the realism of fairwashing racist features. The dataset captures detailed information about the criminal history (jail and prison time), demographic attributes, and "COMPAS risk scores" of 6,172 defendants. $51.4\,\%$ of the defendants are African Americans and are indicated as such in the data. Moreover, each defendant is labeled as either high-risk or low-risk for recidivism.

> **Note.** We distance ourselves from the "COMPAS Recidivism Algorithm" and the features considered, and we condemn the use of ethnic or religious features utterly. The dataset is only used to emphasize the social impact of fairwashing.

Training a biased model. We split off 30 % of the data as a hold-out testing dataset and use the remaining samples for training. Moreover, we sub-select 21 features (17 categorical and 4 continuous features) to avoid overfitting on dates and ID numbers inline with related work [58]. This data is then fitted with a three layer fully-connected neural network, activated by the ReLU activation function. We use the Adam [32] optimizer with a learning rate and a weight decay of 1×10^{-3}.

The final model is fairly accurate (F1 score: 0.68) but is biased towards the "African-American" feature of the dataset. This bias is clearly visible in the LIME explanations over the entire test dataset. The feature is the fourth most decisive criteria according to the average absolute feature importance.

Fairwashing a racist classifier. We demonstrate that it is possible to disguise the racist bias of the classifier by lowering the visualized importance of the "African-American" feature of the dataset. *Note that the classifier is still biased* and the F1 score remain high. A malicious model owner can change the explanation to hide the biased feature from the Top_k features. To this end, the critical feature is assigned 0 in the surrogate model and 1 to other highly relevant features.

We use 50 perturbed instances per sample and use all positive class (high risk of recidivism) samples. We then fine-tune the model using our method, so that the "African-American" feature eventually descents out of the Top_5 relevant features as indicated by LIME.

Table VIII summarizes our results as the averaged LIME attribution scores of the biased and fairwashed models. For the latter, the rank of the "African-American" feature is not in the Top_5 but in the Bottom_5. Fig. 9 visualizes the change in feature rank based on the average relevance in the test data.

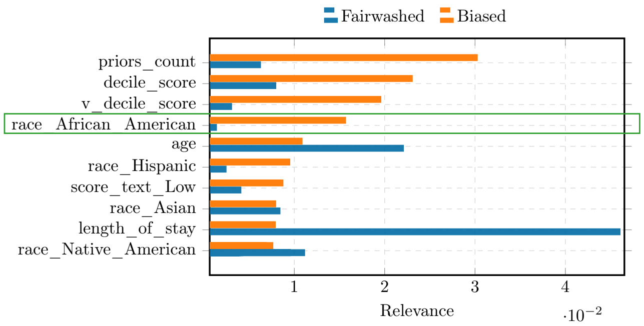


Fig. 9: Change in average relevance of the Top_{10} feature of the initially biased model and their change in the fairwashed model. The "African American" feature is clearly indicated before but is not after fairwashing.

TABLE VIII: Quantitative results of the fairwashing attack on tabular (COMPAS) with $k = 5$. We show the set intersection ($\mathrm{FW}/\mathrm{SI}^{Tk}$) as well as precision, recall, and F1-score.

Model	**Precision**	**Recall**	**F1**	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$
Biased	0.633	0.743	0.684	0.948	0.000
Fairwashed	0.654	0.703	0.678	0.001	0.636

VII. Mitigation

The success of Makrut attacks relies on the adversary's knowledge of LIME's parameters. In particular, we exploit the employed perturbation technique to manipulate the model, which consists of two key factors: the segmentation used and the choice of baseline features (the feature/pixel value used for the perturbations). Our evaluation thus far assumes quickshift segmentation and black color as the baseline feature in the image domain—LIME's default parameterization.

A potential defense strategy thus is to randomize the used segmentation algorithm and/or the chosen baseline features during the explanation process, making more challenging for the attacker to guess the exact parameterization. To test this defense approach against backdooring attacks, we generate explanations on manipulated models by randomly varying both the segmentation (two different parameterizations of Quickshift, SLIC [2] or Felzenszwalb [20]) and baseline feature (black color, blurring and average) for each sample in the test dataset. Table IX shows the yield results and compares them to the attack on default perturbation.

Contrary to our expectation, the attack remains effective, though with reduced ability to highlight the replacement segment and to demote the trigger segment. Notably, however, the trigger segment was more effectively concealed. The results show that the Makrut attacks transfer to perturbation techniques other than the one actually assumed when manipulating the model. We further study transferability to other perturbation techniques in Appendix A. Mitigating Makrut attacks without degrading the quality of explanations turns out to be a significant challenge.

However, existing defenses against backdooring and data poisoning that focus on prediction manipulation [35, 37, 68, 73] likely are effective against Makrut attacks still. We leave further investigation of such defenses to future work.

TABLE IX: Quantitative results of the backdooring attack measured with $k = 5$. We show the overlap of trigger features ($\mathrm{BD\text{-}TS}/\mathrm{SI}^{Tk}$ and $\mathrm{BD\text{-}TS}/\mathrm{SI}^{Bk}$) and replacement features ($\mathrm{BD\text{-}RS}/\mathrm{SI}^{Tk}$ and $\mathrm{BD\text{-}RS}/\mathrm{SI}^{Bk}$) using random perturbations and default perturbations.

Model	**ACC**	**ASR**	**Trigger**		**Replacement**	
			$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\uparrow}$	$\mathbf{SI}^{Bk}_{\downarrow}$
Base Model	96.8 %	98.6 %	0.904	0.032	0.085	0.111
Makrut-BD	97.3 %	98.9 %	0.312	0.576	0.966	0.008
Makrut-BD_{random}	97.3 %	98.9 %	0.217	0.316	0.482	0.016

VIII. Conclusion

We demonstrate the first model manipulation-attack against black-box explanation methods using the example of LIME, SHAP and RISE. Moreover, we even extend the threat model from malicious on-site training to data-poisoning attacks.

The feasibility of the Makrut attacks, thus, severely impacts the trustworthiness of the explanations provided by the operator of remote machine learning models. The operator might disguise highly unfair or even racist operation which we demonstrate both in the image domain and a real-world dataset with tabular data. With the extension to data poisoning, in turn, it even becomes possible that an external adversary influences the credibility of benignly trained models.

While we demonstrate successful attacks against popular and wide-spread black-box explanations, applicability to black-box explanations in general remains to be shown in future work. However, our findings underline the need for effective defenses against this sort of attacks, calling the community for action.

Acknowledgement

The authors gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) under the project DataChainSec (FKZ16KIS1700) and by the Helmholtz Association (HGF) within topic "46.23 Engineering Secure Systems".

Availability

To foster future research on the security and trustworthiness of black-box explanation methods such as LIME and SHAP, we make our code available at:

`https://intellisec.de/research/makrut`

References

[1] T. A. A. Abdullah, M. S. M. Zahid, and W. Ali. A Review of Interpretable ML in Healthcare: Taxonomy, Applications, Challenges, and Future Directions. *Symmetry*, 13(12):2439, 2021.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11):2274–2282, 2012.

[3] U. Aïvodji, H. Arai, O. Fortineau, S. Gambs, S. Hara, and A. Tapp. Fairwashing: The risk of rationalization. In *Proc. of the International Conference on Machine Learning (ICML)*, volume 97, pages 161–170, 2019.

[4] U. Aïvodji, H. Arai, S. Gambs, and S. Hara. Characterizing the risk of fairwashing. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 14822–14834, 2021.

[5] D. Alvarez-Melis and T. S. Jaakkola. On the robustness of interpretability methods. *Proc. of the ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2018.

[6] R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

[7] 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*, page 46, 2015.

[8] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research (JMLR)*, 11:1803–1831, 2010.

[9] N. Barr Kumarakulasinghe, T. Blomberg, J. Liu, A. Saraiva Leao, and P. Papapetrou. Evaluating Local Interpretable Model-Agnostic Explanations on Clinical Machine Learning Classification Models. In *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 7–12, 2020.

[10] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[11] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[12] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Proc. of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 387–402, 2013.

[13] C. Burger, L. Chen, and T. Le. "Are Your Explanations Reliable?" Investigating the Stability of LIME in Explaining Text Classifiers by Marrying XAI and Adversarial Attack. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12831–12844, 2023.

[14] G. Carbone, L. Bortolussi, and G. Sanguinetti. Resilience of bayesian layer-wise explanations under adversarial attacks. In *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.

[15] J. Chen, X. Wu, V. Rastogi, Y. Liang, and S. Jha. Robust attribution regularization. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 14300–14310, 2019.

[16] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[17] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6970–6979. Curran Associates Inc., 2017.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[19] K. D. Doan, Y. Lao, W. Zhao, and P. Li. LIRA: learnable, imperceptible and robust backdoor attacks. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11946–11956, 2021.

[20] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[21] R. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3449–3457, 2017.

[22] F. Gabbay, S. Bar-Lev, O. Montano, and N. Hadad. A LIME-Based Explainable Machine Learning Model for Predicting the Severity Level of COVID-19 Diagnosed Patients. *Applied Sciences*, 11(21):10417, 2021.

[23] A. Ghorbani, A. Abid, and J. Y. Zou. Interpretation of neural networks is fragile. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2019.

[24] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[25] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. XAI—Explainable artificial intelligence. *Science Robotics*, 4, 2019.

[26] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing. LEMNA: Explaining deep learning based security applications. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*, pages 364–379, 2018.

[27] A. Hanif, X. Zhang, and S. Wood. A Survey on Explainable Artificial Intelligence Techniques and Challenges. In *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pages 81–89, 2021.

[28] A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970.

[29] J. Howard. Imagewang. URL https://github.com/fastai/imagenette/.

[30] IBM Corp. Configuring explainability in watson openscale. https://dataplatform.cloud.ibm.com/docs/content/wsj/model/wos-explainability-config.html?context=cpdaas. (visited May 2024).

[31] A. Ivankay, I. Girardi, C. Marchiori, and P. Frossard. FAR: A general framework for attributional robustness. In *Proc. of the British Machine Vision Conference (BMVC)*, page 24, 2021.

[32] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

[33] H. Lakkaraju, N. Arsov, and O. Bastani. Robust and stable black box explanations. In *Proc. of the International Conference on Machine Learning (ICML)*, volume 119, pages 5628–5638, 2020.

[34] A. Levine, S. Singla, and S. Feizi. Certifiably robust interpretation in deep learning. *CoRR*, abs/1905.12105, 2019.

[35] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Anti-Backdoor Learning: Training Clean Models on Poisoned Data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14900–14912. Curran Associates, Inc., 2021.

[36] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 23 (1), 2021.

[37] K. Liu, B. Dolan-Gavitt, and S. Garg. Fine-Pruning: Defending against backdooring attacks on deep neural networks. In *Proc. of the International Symposium Research in Attacks, Intrusions, and Defenses (RAID)*, volume 11050, pages 273–294, 2018.

[38] Y. Lu, G. Kamath, and Y. Yu. Indiscriminate data poisoning attacks on neural networks. *Transaction of Machine Learning Research*, 2022, 2022.

[39] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*, page 10, 2017.

[40] R. Marcinkevičs and J. E. Vogt. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. *WIREs Data Mining and Knowledge Discovery*, 2023.

[41] S. Mattu, J. Larson, L. Kirchner, and J. Angwin. Machine bias, 2016.

[42] G. Montavon. Gradient-based vs. propagation-based explanations: An axiomatic comparison. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700, pages 253–265. 2019.

[43] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Definitions, methods, and applications in interpretable machine learning. *Proc. of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[44] M. K. Nallakaruppan, B. Balusamy, M. L. Shri, V. Malathi, and S. Bhattacharyya. An Explainable AI framework for credit evaluation and analysis. *Applied Soft Computing*, 153:111307, 2024.

[45] M. Noppel, L. Peter, and C. Wressnegger. Disguising attacks with explanation-aware backdoors. In *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, 2023.

[46] V. Petsiuk, A. Das, and K. Saenko. RISE: Randomized input sampling for explanation of black-box models. In *Proc. of the British Machine Vision Conference (BMVC)*, 2018.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[48] W. Saeed and C. W. Omlin. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowl. Based Syst.*, 263:110273, 2023.

[49] M. Sahakyan, Z. Aung, and T. Rahwan. Explainable Artificial Intelligence for Tabular Data: A Survey. *IEEE Access*, 9:135392–135422, 2021.

[50] A. Sarica, A. Quattrone, and A. Quattrone. Introducing the rank-biased overlap as similarity measure for feature importance in explainable machine learning: A case study on parkinson's disease. In *Proc. of the International Conference Brain Informatics (BI)*, volume 13406, pages 129–139.

[51] A. Sarkar, A. Sarkar, and V. N. Balasubramanian. Enhanced regularizers for attributional robustness. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 2532–2540, 2021.

[52] A. Sathyan, A. I. Weinberg, and K. Cohen. Interpretable AI for biomedical applications. *Complex engineering systems (Alhambra, Calif.)*, 2(4):18, 2022.

[53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2), 2020.

[54] A. Shamsabadi, M. Yaghini, N. Dullerud, S. Wyllie, U. Aïvodji, A. Alaagib, S. Gambs, and N. Papernot. Washing the unwashable : On the (im)possibility of Fairwashing detection. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 14170–14182, 2022.

[55] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[56] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. of the International Conference on Learning Representations (ICLR) Workshop Track Proceedings*, 2014.

[57] S. Sinha, H. Chen, A. Sekhon, Y. Ji, and Y. Qi. Perturbing inputs for fragile interpretations in deep natural language processing. In *Proc. of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP@EMNLP)*, 2021.

[58] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pages 180–186, 2020.

[59] D. Slack, S. Hilgard, S. Singh, and H. Lakkaraju. Feature attributions and counterfactual explanations can be manipulated. In *Proc. of the ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2021.

[60] I. Stepin, J. M. Alonso, A. Catalá, and M. Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021.

[61] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proc. of the International Conference on Machine Learning (ICML)*, volume 70, pages 3319–3328, 2017.

[62] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147. pmlr, 2013.

[63] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014.

[64] R. Tang, N. Liu, F. Yang, N. Zou, and X. Hu. Defense against explanation manipulation. *Frontiers Big Data*, 5:704203, 2022.

[65] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58 (1):267–288, 1996.

[66] A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *Computer Vision – ECCV 2008*, pages 705–718, 2008.

[67] V. Vimbi, N. Shaffi, and M. Mahmud. Interpreting artificial intelligence models: A systematic review on the application of LIME and SHAP in Alzheimer's disease detection. *Brain Informatics*, 11(1):10, 2024.

[68] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural Cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, pages 707–723, 2019.

[69] F. Wang and A. W.-K. Kong. Exploiting the relationship between kendall's rank correlation and cosine similarity for attribution protection. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 20580–20591, 2022.

[70] F. Wang and A. W.-K. Kong. A Practical Upper Bound for the Worst-Case Attribution Deviations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24616–24625, 2023.

[71] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck. Evaluating explanation methods for deep learning in computer security. In *Proc. of the IEEE European Symposium on Security and Privacy (EuroS&P)*, Sept. 2020.

[72] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. 28(4):20:1–20:38.

[73] D. Wu and Y. Wang. Adversarial neuron pruning purifies backdoored deep models. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 16913–16925, 2021.

[74] Y. Wu, L. Zhang, U. A. Bhatti, and M. Huang. Interpretable Machine Learning for Personalized Medical Recommendations: A LIME-Based Approach. *Diagnostics*, 13(16):2681, 2023.

TABLE X: Results of the backdooring attack measured with $k = 5$ using different segmentation methods. We show the overlap of trigger features (BD-TS/SI^{Tk} and BD-TS/SI^{Bk}) and replacement features (BD-RS/SI^{Tk} and BD-RS/SI^{Bk}).

Segmentation	**Trigger**		**Replacement**		**Clean**	
	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\uparrow}$	$\mathbf{SI}^{Bk}_{\downarrow}$	**RBO**	**MSE**
quickshift	0.904	0.032	0.085	0.111	0.710	0.169
felzenszwalb	0.606	0.071	0.025	0.044	0.573	0.500
slic	0.904	0.032	0.083	0.112	0.572	0.499
quickshift2	0.755	0.015	0.001	0.005	0.506	0.169

(a) Base Model

Trigger		**Replacement**		**Clean**	
$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\uparrow}$	$\mathbf{SI}^{Bk}_{\downarrow}$	**RBO**	**MSE**
0.312	0.576	0.966	0.008	0.618	0.539
0.070	0.694	0.815	0.004	0.532	0.318
0.006	0.064	0.417	0.008	0.525	0.157
0.073	0.628	0.481	0.003	0.502	0.032

(b) MAKRUT-BD

TABLE XI: Results of the backdooring attack measured with $k = 5$ using different baseline features. We show the overlap of trigger features (BD-TS/SI^{Tk} and BD-TS/SI^{Bk}) and replacement features (BD-RS/SI^{Tk} and BD-RS/SI^{Bk}).

Baseline	**Trigger**		**Replacement**		**Clean**	
	$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\uparrow}$	$\mathbf{SI}^{Bk}_{\downarrow}$	**RBO**	**MSE**
black	0.904	0.032	0.085	0.111	0.710	0.169
fudged	0.834	0.052	0.080	0.117	0.706	0.522
blur	0.902	0.032	0.087	0.101	0.712	0.608
red	0.145	0.230	0.115	0.059	0.910	0.166
green	0.172	0.237	0.117	0.059	0.823	0.238
blue	0.185	0.209	0.138	0.045	0.880	0.164

(a) Base Model

Trigger		**Replacement**		**Clean**	
$\mathbf{SI}^{Tk}_{\downarrow}$	$\mathbf{SI}^{Bk}_{\uparrow}$	$\mathbf{SI}^{Tk}_{\uparrow}$	$\mathbf{SI}^{Bk}_{\downarrow}$	**RBO**	**MSE**
0.312	0.576	0.966	0.008	0.618	0.539
0.735	0.141	0.576	0.042	0.626	0.740
0.677	0.217	0.825	0.022	0.626	0.687
0.401	0.057	0.014	0.952	0.842	0.196
0.614	0.059	0.012	0.932	0.773	0.202
0.179	0.171	0.009	0.958	0.831	0.199

(b) MAKRUT-BD

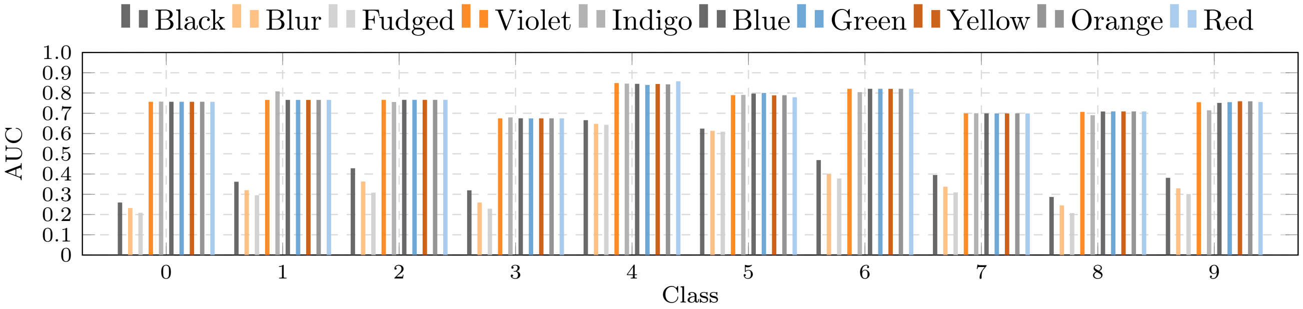


Fig. 10: The AUC ("area under the curve") of the average descriptive accuracy [71] when flipping the $\mathbf{Top_{10}}$ features for different baselines on clean samples for the manipulated model using backdooring attack.

APPENDIX

We additionally assess how well the attacks transfer to different baselines and segmentation algorithms, even though they are designed for default LIME configurations.

Segmentation. We test the manipulated model by generating explanations with Felzenszwalb and SLIC segmentation methods. Additionally, we apply an alternative parameterization of the quickshift method, with parameters $kernelsize$=3, $maxdist$=6, and $ratio$=0.5. This configuration is referred as quickshift2 in Table X which shows the transferability of the backdooring attack. Despite the attack being tailored to a specific quickshift parameterization, we observe that it successfully transfers to all tested segmentation methods.

Baseline. LIME, by default, uses black (pixel value 0) as the baseline for generating explanations. Previous research has suggested alternatives such as a blurred version of the input image or fudged version (filling segments with average pixel value) instead of using black. Moreover, we test the effectiveness of using all the colors from color spectrum. However, using colors other than black produced poor quality explanations even on the clean samples on the manipulated model. To evaluate these options, we generated explanations using different baselines and measured their quality by calculating the AUC of the average descriptive accuracy. As shown in Fig. 10, explanations using other colors as a baseline performed poorly. Therefore, we use the blurred input image and segment-wise average pixel values to test transferability. While more trigger segments are highlighted in the explanations with blurred and fudged baselines, some samples still showed the trigger as the least important segment. Moreover, the target segment is highlighted consistently as shown in Table XI.