SHLIME: Foiling adversarial attacks fooling SHAP and LIME

Sam Chauhan[1], **Estelle Duguet**[1], **Karthik Ramakrishnan**[1], **Hugh Van Deventer**[1], **Jack Kruger**[1], **Ranjan Subbaraman**[1]

[1]University of Michigan
{csanjana, eduguet, kartikrk, hughv, jakrug, ranjans}@umich.edu

Abstract

Individual metrics for classifier explanation provide semantic reasoning to improve interpretability of a machine learning model, allowing engineers to further improve and understand black-box models. They can also be applied to a subfield of machine learning, adversarial model analysis, allowing engineers to detect the implicit biases of classifiers and better assess their generalizability and utility. We seek to build on the previous analysis by the paper Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods (Slack et al., 2020), with the goal of developing a new robust framework for the analysis of adversarial networks to better detect implicit biases. Although the previous work presented a scaffolding framework to hide the biases of a given model, we approach bias analysis from the other direction. Rather than iterating on this framework, we analyze the individual weaknesses of LIME and SHAP on intentionally biased models and experiment with multiple ensemble methods to improve bias detection. First, we confirm the results of the previous study, identifying the modifications required to run the original experiment. In order to validate these results, we exactly recreate the original COMPAS experiment results using LIME and SHAP explainers separately on models of varying F1 scores. We then introduce our framework for testing augmented and ensemble methods involving LIME and SHAP on out-of-distribution classifiers. To allow for as many experiments as possible, we created a testing framework which allowed us to input various methods with identical models and datasets. This made the testing process efficient and adaptable to any ensemble or augmentation we attempted. We generated similar graphs to the original COMPAS experiment, allowing for direct comparison with the replicated results for LIME and SHAP, seeking the method most resistant to biased classifiers as F1 score improves. Finally, we compare the results of our methods with the original results of the COMPAS experiment to identify methods with significant improvement and utility for future works. We then discuss the implications of our findings in the growing deployment of black-box classifiers in various fields.

Introduction

Our research problem focuses on the vulnerabilities of post hoc explanation methods like LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations) to adversarial attacks. Post hoc means that these methods explain the decisions of a model after the model has already been trained and without changing how the model works. These explanation techniques are widely used in machine learning to provide interpretability of black-box models. It is important to understand the decision-making process of these black-box models as they are increasingly deployed in critical domains such as healthcare and criminal justice.

The project aims to explore how adversarial models can be crafted to fool these explanation methods, leading to misleading or incorrect explanations of model predictions. Specifically, the question is: How can adversarial attacks be designed to manipulate the outputs of LIME and SHAP while keeping the model predictions consistent? This can have significant implications, as adversarial models that have discriminatory biases cannot be easily identified using post hoc explanation methods.

In this project, we will explore explainable AI (XAI), Adversarial Learning, and Model Interpretability and Robustness sub-areas in ML. In our replication, we will be using popular datasets for fairness in ML, including COMPAS, Communities and Crime, and German Credit. Both LIME and SHAP have readily available Python libraries with well-documented APIs that can be leveraged for this project. In addition, the paper has a GitHub repository that provides the necessary code to reproduce the project. This makes it easier to explore the adversarial attacks on post hoc explanation methods and reproduce the results of the experiments.

The project is computationally viable because the datasets mentioned above are small and can be processed efficiently. LIME generates local explanations efficiently, and while SHAP is more computationally expensive, it remains practical for small to medium datasets. The project focuses on devising adversarial attacks that involve manipulating input data rather than retraining entire models. This reduces computational complexity and makes the project both time-efficient and feasible within the available time frame.

Paper link: https://arxiv.org/abs/1911.02508 (Slack et al. 2020)

Related Work

The Mythos of Model Interpretability (Lipton 2017): This paper posits the idea that post hoc explanations do not guarantee that the model is unbiased. The paper argues that post hoc explanations can be optimized to disguise malicious intent in models, therefore using plausible-sounding explanations to mislead.

Interpretation of Neural Networks is Fragile (Ghorbani, Abid, and Zou 2018): This paper shows that even tiny, nearly imperceptible perturbations on the input can significantly alter the output of explainability tools. The paper also demonstrates that explanation techniques can be highly sensitive to small perturbations, even if the underlying classifier's predictions remain unchanged.

Explanations can be Manipulated and Geometry is to Blame (Dombrowski et al. 2019): This paper demonstrates how explanation methods are similarly vulnerable to imperceptible perturbations, resulting in a large change in output while the input remains visibly unchanged. The paper also proposes some mechanisms to improve the robustness of explanations in the context of visual models.

Background

In this section, we discuss the intuition and math underpinning seminal post hoc explanation techniques- LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017).

LIME and SHAP

Complex models, such as ensemble models and deep neural networks, lack the natural interpretability found in simpler models, such as linear models and decision trees. These sophisticated models often operate as black boxes, making their decision-making process opaque. To gain insight into how these complex models work, researchers can create simplified proxy models that approximate and explain the behavior of complex models.

In this replication, we focus on linear explainer models. These models, known as additive feature attribution methods, are characterized by an explanation model that is a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

where $z' \in \{0,1\}^M$, M is the number of simplified input features, and $\phi_i \in \mathbb{R}$.

Both LIME and SHAP seek to find interpretable models that locally approximate a complex model to explain how input features affect the complex model's output. They share a similar optimization framework:

$$\arg\min_{g \in \mathcal{G}} L(f, g, \pi_x) + \Omega(g) \tag{2}$$

where

$$L(f, g, \pi_x) = \sum_{x' \in \mathcal{X}'} [f(x') - g(x')]^2 \pi_x(x') \tag{3}$$

While LIME and SHAP share this fundamental structure, they differ in their specific implementations. LIME uses a distance-based weighting function $\pi_x(z) = \exp(-D(x,z)^2/\sigma^2)$ where D is some distance function (such as L2 distance), defines $\Omega(g)$ as the number of non-zero weights in the linear model, and samples around the data point x by adding noise.

SHAP, on the other hand, samples around x by only extracting a random subset of input features, and grounds its selections for Ω and π in game-theoretic principles, satisfying three key properties:

1. Local accuracy (explanation must match original model output)
2. Missingness (absent features must have zero impact)
3. Consistency (increased feature contributions cannot decrease attribution)

Notably, the authors demonstrate that Shapley values are the unique solution that satisfies these three axioms, providing a theoretical foundation for the SHAP approach to model interpretation. The authors then show the following selections for Ω and π recover the Shapley values.

Theorem 1 (Shapley kernel)**.** *Under Definition 1, the specific forms of $\pi_{x'}$, L, and Ω that make solutions of Equation 2 consistent with Properties 1 through 3 are:*

$$\Omega(g) = 0,$$

$$\pi_{x'}(z') = \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)},$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in \mathcal{Z}} [f(h_x^{-1}(z')) - g(z')]^2 \pi_{x'}(z'),$$

where $|z'|$ is the number of non-zero elements in z'.

Proposed Framework for Adversarial Classifiers

Overview

We discuss a novel adversarial framework to construct adversarial classifiers that can fool post hoc explanation techniques, such as LIME and SHAP (Slack et al. 2020). These classifiers behave identically to a biased classifier f on real-world data but produce innocuous explanations when probed by explanation techniques. Below, we detail the framework's components: the setup and assumptions, the intuition behind the approach, and the methodology for constructing adversarial classifiers.

Preliminaries

Problem Setting Consider an adversary seeking to deploy a biased classifier f to make critical decisions (e.g., parole, bail, or credit). Customers and regulators, with black-box access to f, use post hoc explanation techniques to assess its fairness and readiness for deployment. If biases in f are detected, then the adversary will not be able to deploy the model. The adversary's goal is to obscure f's biases from these explanation techniques.

Input The adversary provides the following input:

- **Biased classifier** f, intended for deployment.
- **Dataset** X, sampled from the input distribution X_{dist}, representing the real-world data on which f is applied.

Output The framework outputs an **adversarial classifier** e, which:

- behaves identically to f on X_{dist}.
- appears unbiased when analyzed by explanation techniques such as LIME or SHAP.

Intuition Behind the Approach

Post hoc explanation techniques, such as LIME and SHAP, approximate the behavior of a black-box model within the local neighborhood of a given data point by perturbing its input features. However, such perturbed instances often lie off-manifold or are out-of-distribution (OOD), making them distinct from real-world data points.

To demonstrate this, we performed principal component analysis (PCA) on a dataset combining original and perturbed instances generated by LIME. The visualization (Figure 1) revealed that the perturbed instances are clearly distinguishable from the original data. This insight suggests that distinguishing between real and perturbed instances is feasible, enabling the adversary to construct a classifier that behaves differently on perturbed data, thereby deceiving post hoc explanation techniques.

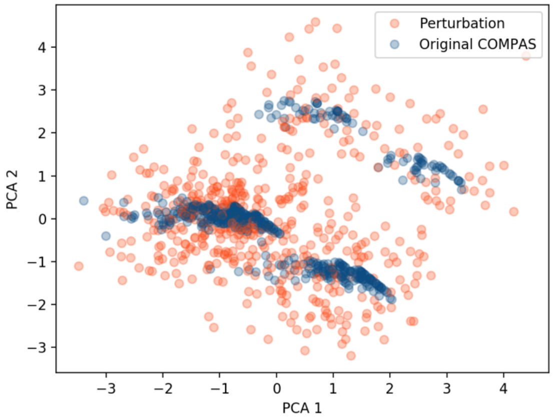


Figure 1: PCA applied to the COMPAS dataset (blue) as well as its LIME style perturbations (red)

Framework Design

Adversarial Classifier Construction The adversarial classifier e is designed as follows:

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

Here:

- $f(x)$: Biased classifier behavior on real-world data.
- $\psi(x)$: Unbiased classifier that makes predictions based on features that are uncorrelated with sensitive attributes.

This construction ensures that e mimics f on real data while exhibiting unbiased behavior on synthetic or perturbed instances.

Detecting Out-of-Distribution (OOD) Samples To differentiate between real and perturbed data, we train a classifier is_OOD as follows:

1. Perturb each instance $x \in X$ to generate a corresponding set X_p of perturbed instances.
2. Assign labels:
   - False for real data (X).
   - True for perturbed instances (X_p), unless they overlap with X.
3. Train a classifier on $X \cup X_p$ using these labels.

By accurately detecting OOD samples, is_OOD enables the adversarial classifier e to behave selectively, presenting innocuous explanations for perturbed instances.

Implementation Highlights

- The perturbation strategy employed mirrors the methods used by LIME and SHAP to ensure robustness against common explanation techniques.
- The classifier is_OOD is trained using standard off-the-shelf supervised learning algorithms, providing flexibility in its implementation.
- Our approach requires only black-box access to the biased classifier f, making it broadly applicable across domains where post hoc explanation techniques are utilized.

Methodology, Results, and Discussion

Replication:

The authors conducted experiments using multiple datasets, including COMPAS for recidivism risk prediction, Communities and Crime for violent crime prediction, and German Credit for credit scoring. Each dataset includes a sensitive feature: race in COMPAS and Communities and Crime, and gender in German Credit. Following their proposed framework, the authors constructed an adversarial classifier from an existing biased classifier to achieve near-perfect fidelity with respect to the biased classifier. Fidelity in this context refers to the degree to which the adversarial classifier replicates the decisions of the original biased classifier. They apply SHAP and LIME to the biased and adversarial classifiers to explain their predictions. They found that while LIME and SHAP accurately identify the sensitive feature as the most important feature in the biased classifier, they deflect the importance from the sensitive attribute to other uncorrelated features when applied to the adversarial classifier, consistently across datasets. This finding is particularly significant, as it demonstrates that LIME and SHAP explanations can be systematically manipulated to obscure biased features using adversarial techniques. In this project, we plan to replicate this experiment, specifically focusing on the COMPAS

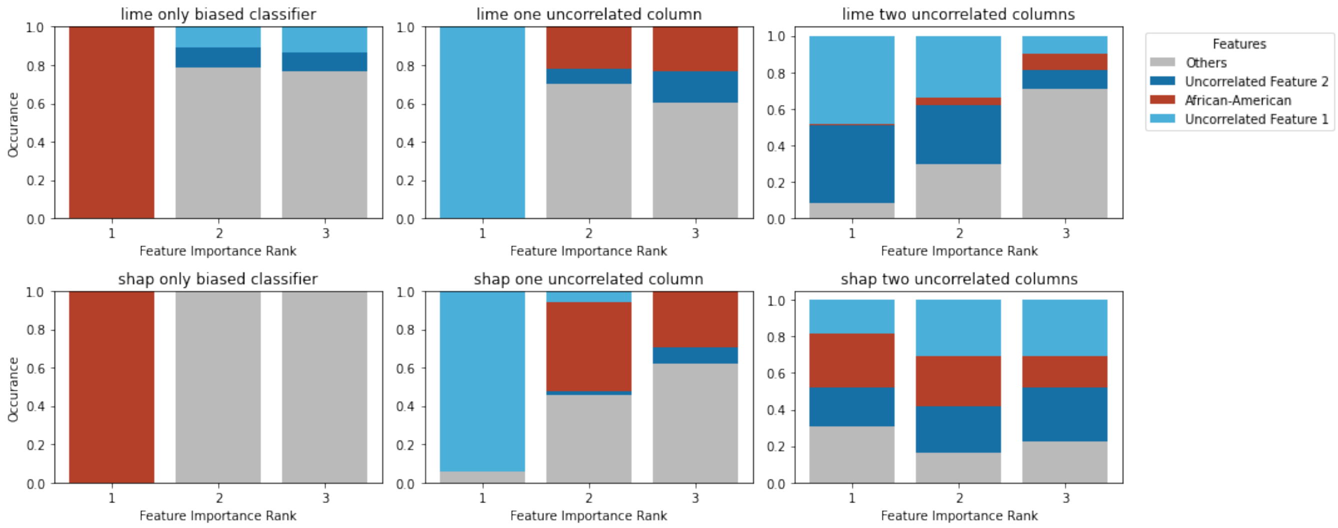


Figure 2: Replication of top 3 features on the COMPAS dataset

dataset, to validate the authors' observations regarding the feature importance shifts in adversarial classifiers.

In the prescribed framework to construct an adversarial classifier, the authors used an Out-of-Distribution (OOD) classifier to decide whether or not a given instance is perturbed or from the original distribution. The authors found that the accuracy of the OOD classifier (measured by the F1 metric) significantly impacts the feature attribution by SHAP and LIME on the adversarial classifiers. The more accurate the OOD classifier, the more likely SHAP and LIME provide misleading explanations on the adversarial classifiers. This is an important finding that we will replicate as it forms the basis of the extension of our research project.

For our replication, we focus on two specific figures from the original paper: Figures 6 and 7. Figure 6 represents how well the adversarial models impact LIME and SHAP's capabilities to determine bias. The contrast between the performance of LIME and SHAP against the biased classifier (on the left) as compared to the adversarial classifiers in the middle and on the right is apparent and demonstrates the effectiveness of their adversarial models. Figure 7 ties in with what was discussed in the previous paragraph, showing how effective the adversarial models are at fooling LIME and SHAP with varying accuracies of the OOD classifier. We replicated these figures as they are crucial in determining the effectiveness of our extension.

A majority of the code used to replicate the findings was made publicly available by the authors of the original paper. This included the OOD classifiers for both LIME and SHAP as well as the biased and adversarial classifiers. Specifically, the code that the original paper used to create the graphs in figure 6 was available in whole. Using this code, we created our adversarial classifiers for various accuracies (F1 Scores) and tested how well they fooled LIME and SHAP respectively. In the end, our results were near identical to the paper's (Figures 4 and 5), and we reinforced the results that that the adversarial classifiers fooled LIME and SHAP with

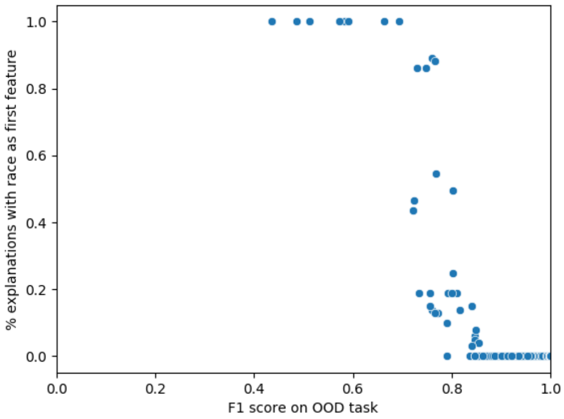


Figure 3: Replicated LIME Sensitivity Analysis

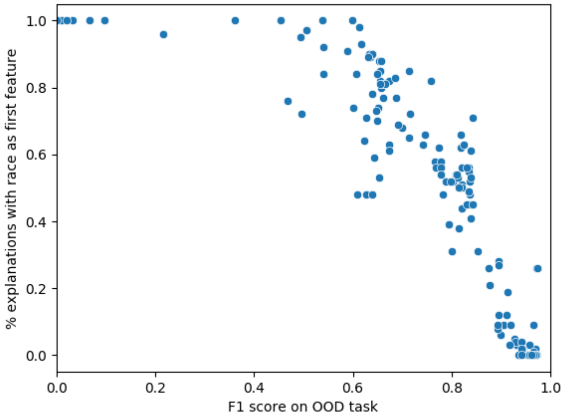


Figure 4: Replicated SHAP Sensitivity Analysis

varying success over differing accuracies. Specifically, that being that the ability of the adversarial model to fool LIME sharply increased as the F1 score increased above 0.75 while the ability of the model to fool SHAP increased comparatively gradually beginning around an F1 score of 0.5.

For replicating Figure 7, the code base provided by the original paper's authors supplied a baseline application to the COMPAS dataset but necessitated slight modifications to the code to calculate and output the values as needed for replication. By using these modified outputs, we were able to completely recreate the figure with a large degree of similarity (Figure 2). These results reinforce the original paper's proposal that their adversarial classifiers were able to fool LIME and SHAP into thinking a biased classifier was unbiased and creates the foundation of our extension.

Testing Framework:

Extensions

1. **Significance:**

   Understanding the rationale behind a model's prediction is often as critical as, if not more critical than, the model's prediction itself. This emphasis on interpretability is at the heart of ExplainableAI (XAI) and fairness. In response, various post-hoc interpretability methods have been developed, with LIME and SHAP emerging as the two of the most prominent techniques. Given that LIME and SHAP are commonly employed to reveal the underlying vulnerabilities of complex models, it is also crucial to examine potential weaknesses in these explanation methods themselves. If adversarial examples can exploit or bypass the interpretative frameworks of LIME and SHAP, the reliability of these methods may be compromised. By demonstrating that adversarial models can manipulate the explanations provided by LIME and SHAP, this research exposes the shortcomings of SHAP and LIME. These findings not only highlight the need to scrutinize these post-hoc explanation methods but also suggest future directions for refining these methods by addressing their shortcomings. Strengthening these methods against adversarial manipulation will enhance their reliability, advancing more robust and trustworthy interpretability in machine learning.

2. **Extension(s):** For our extension, we propose developing SHLIME, a novel ensemble approach that combines LIME and SHAP explanations to create a more robust explanation method against adversarial attacks. Our motivation stems from the paper's key finding that LIME and SHAP exhibit complementary vulnerabilities to Out-of-Distribution (OOD) classifiers: SHAP is partially susceptible to attacks from less accurate OOD classifiers (F1 $\approx$ 0.45) but requires highly accurate classifiers for complete deception, while LIME remains resilient until higher accuracies (F1 $\approx$ 0.7) but becomes easily fooled beyond that threshold (F1 $>$ 0.8). See Figure 3 for reference.

   We hypothesize that by intelligently combining these methods, we can leverage their complementary strengths to create an explanation method that maintains high reliability across a broader range of OOD accuracies. The challenge in combining LIME and SHAP is that their outputs are represented differently — SHAP values lie between 0 and 1, while LIME values can be unbounded and negative. Therefore, whatever method we use to combine them must preserve as much information as possible from both models. The most straightforward approach to doing so was simply multiplying the two values together, using the SHAP values as a scaling factor for LIME's output. Other methods for combining the two models such as taking the weighted average of LIME and SHAP values would result in losing some information from either model, most starkly in cases where the outputs of LIME and SHAP diverged significantly.

   We hypothesize that a successful implementation of SHLIME should demonstrate enhanced robustness, remaining unfounded until higher F1 scores similar to LIME, while showing a more gradual degradation in performance with increase in F1 scores similar to SHAP. This behavior contrasts favorably with the more abrupt or earlier vulnerability transitions of individual methods. We investigate the performance of BASIC SHLIME, an ensemble method that returns the LIME value multiplied by the SHAP value for each feature.

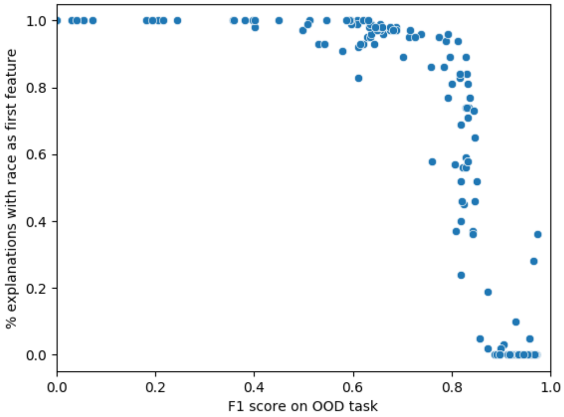


Figure 5: BASIC SHLIME Sensitivity Analysis

Conclusions

As shown in Figure 5, our experimental results demonstrate that this simple yet effective approach successfully improves upon both methods. BASIC SHLIME maintains strong explanation accuracy (consistently above 0.8) for a wider range of F1 scores (similar to LIME), only beginning to show significant degradation around F1 $\approx$ 0.75. Even as the classifier accuracy increases beyond this threshold, SHLIME exhibits a more gradual decay in performance (similar to SHAP). By leveraging the complementary characteristics of SHAP and LIME, BASIC SHLIME provides more reliable explanations than SHAP and LIME even as adversarial models become more sophisticated.

Future Work

The natural path to take after this research would be to create a proper OOD classifier for our potential SHLIME models. While our results using both the LIME and SHAP OOD classifiers are promising, the strength of our results is limited by the inability to truly compare its robustness on a fair scale in comparison to LIME and SHAP individually. This was a problem that we initially intended to include within this research project, but quickly realized its scale and complexity because this would vary immensely from prior research due to our methods combining both LIME and SHAP. Moreover, each different possible option for SHLIME would necessitate a completely different OOD classifier.

Secondly, the focus of this research was mainly on the robustness of our SHLIME implementation, specifically to these kinds of adversarial attack. However, it is still important to consider how well it performs at its intended task of interpretability compared to LIME and SHAP. In future research, it would be useful to see how SHLIME performs in comparison to both LIME and SHAP on a wide array of datasets. Ideally, SHLIME would perform comparably or even better than SHAP, but by introducing LIME to increase robustness, there may potentially be some performance decrease.

Another avenue for future work would be to explore other ways of combining the output of LIME and SHAP. One method is mixture of experts: an approach that divides a model into several sub-networks (or ”experts”), each specialising in a different subset of the input data. For the purpose of this project, one set of experts can specialise in LIME while the other specialises in SHAP. The model's result will then be the combined output of the LIME experts and the SHAP experts.

Societal Impact

The societal impact of the paper is profound, as it directly addresses the integrity and fairness of machine learning models deployed in critical sectors such as healthcare, finance, and criminal justice. Black-box models, which are increasingly used to make high-stakes decisions in these areas, often operate without sufficient transparency, raising concerns about biases, fairness, and accountability. This research highlights vulnerabilities in popular post hoc explanation methods like LIME and SHAP, revealing how easily these models can be manipulated, leading to potentially harmful consequences when they are used to inform decisions about individuals' lives. By uncovering these weaknesses, the paper emphasizes the need for more robust explanation techniques, pushing for a more rigorous approach to model testing and evaluation. The ultimate goal is to build models that are not only accurate but also equitable and transparent, ensuring that they do not perpetuate existing biases or make decisions based on harmful stereotypes. By advocating for the development of higher-quality testing methods, this research aims to improve decision-making processes in sensitive fields, reducing the risks of unjust outcomes and enhancing public trust in AI systems.

References

Dombrowski, A.-K.; Alber, M.; Anders, C. J.; Ackermann, M.; Müller, K.-R.; and Kessel, P. 2019. Explanations can be manipulated and geometry is to blame. arXiv:1906.07983.

Ghorbani, A.; Abid, A.; and Zou, J. 2018. Interpretation of Neural Networks is Fragile. arXiv:1710.10547.

Lipton, Z. C. 2017. The Mythos of Model Interpretability. arXiv:1606.03490.

Lundberg, S. M.; and Lee, S. 2017. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874.

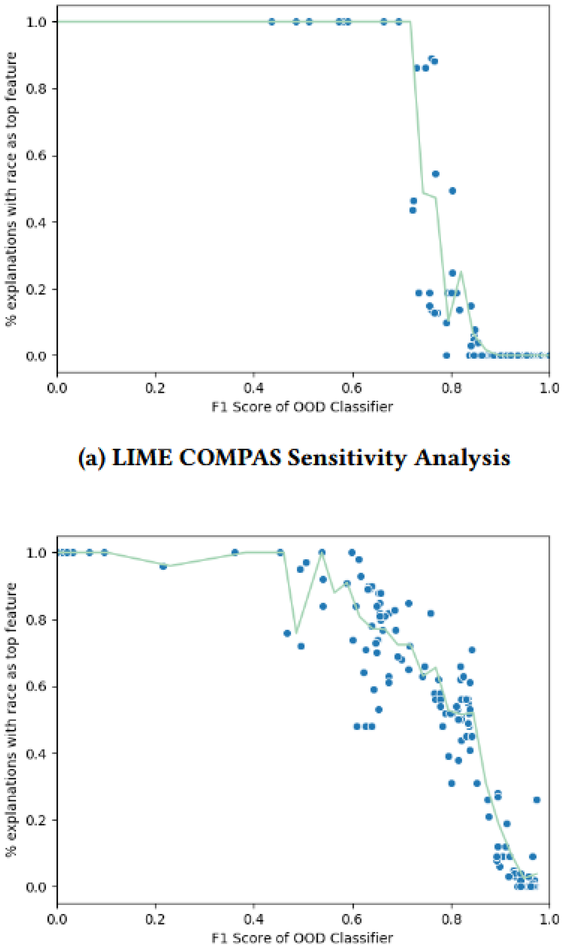
Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. ”Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1135–1144. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322.

Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; and Lakkaraju, H. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. arXiv:1911.02508.

Appendix

Access to our experiment repository: https://github.com/eduguet/shlime

Figures from Slack et al.



(a) LIME COMPAS Sensitivity Analysis

(b) SHAP COMPAS Sensitivity Analysis

Figure 6: Reference graph of LIME and SHAP being fooled by biased OOD classifiers

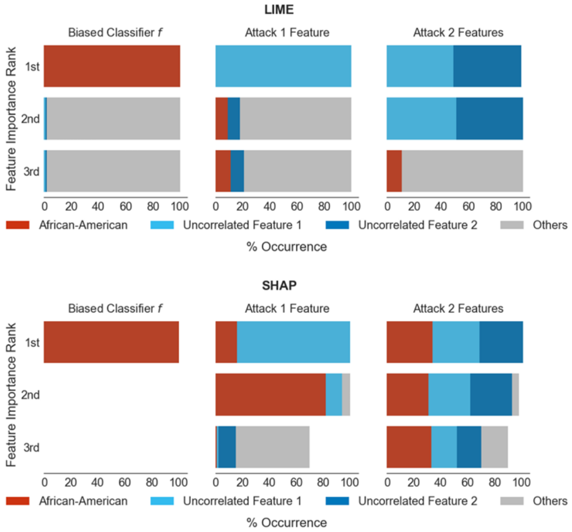


Figure 7: COMPAS: % of data points for which each feature shows up in top 3 (according to LIME and SHAP's ranking of feature importances) for different classifiers