arXiv:2006.12302v1 [cs.LG] 22 Jun 2020

Improving LIME Robustness with Smarter Locality Sampling

Sean Saito
SAP
Singapore
sean.saito@sap.com

Eugene Chua
UCSD
La Jolla, CA, USA
eychua@ucsd.edu

Nicholas Capel
Gallenco Science
Singapore
nicholas.rj.capel@gmail.com

Rocco Hu
SAP
Singapore
rocco.hu@sap.com

ABSTRACT

Explainability algorithms such as LIME have enabled machine learning systems to adopt transparency and fairness, which are important qualities in commercial use cases. However, recent work has shown that LIME's naive sampling strategy can be exploited by an adversary to conceal biased, harmful behavior. We propose to make LIME more robust by training a generative adversarial network to sample more realistic synthetic data which the explainer uses to generate explanations. Our experiments demonstrate that our proposed method demonstrates an increase in accuracy across three real-world datasets in detecting biased, adversarial behavior compared to vanilla LIME. This is achieved while maintaining comparable explanation quality, with up to 99.94% in top-1 accuracy in some cases.[1]

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; • **Human-centered computing** → Systems and tools for interaction design.

KEYWORDS

explainability, robustness, adversarial machine learning

ACM Reference Format:
Sean Saito, Eugene Chua, Nicholas Capel, and Rocco Hu. 2020. Improving LIME Robustness with Smarter Locality Sampling. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Explainability is a topic of growing interest especially in applications where trust and transparency are requirements. Several methods exist for explaining the decisions of an otherwise black-box machine learning model, including the Locally Interpretable Model-Agnostic Explanation (LIME) algorithm [9] and SHAP [5].

Recent work in adversarial machine learning has shown that even these explainability algorithms are vulnerable to adversarial attacks. The objectives of the attacks vary, including extracting a high-fidelity copy of the black-box model ([3]), identifying training data membership of individual data points ([10], [6]), or concealing harmful, biased behavior of black-box models on categories like race, sex, and religion ([11]).

In this work, we propose a method which addresses the last type of attack. In particular, we propose using the Conditional Tabular GAN (CTGAN) model [12] to generate more realistic synthetic data for querying the model to be explained. Our experiments involving both black-box and white-box adversarial attacks demonstrate that our model can more accurately detect biased behavior as compared to the vanilla LIME model. To the best of our knowledge, this is one of the first attempts at making LIME more robust against adversarial attacks.

2 OVERVIEW OF LIME

Given some target model f to be explained, LIME produces an explanation e for it by optimizing for both *local model fidelity* and *complexity*. For some prediction $f(x)$, LIME produces some explanation $e(x)$ which optimizes the following:

$$\arg\min_{e} \mathcal{L}(f, e, \pi_x) + \Omega(e) \tag{1}$$

Local model fidelity $\mathcal{L}$ here refers to how faithful the explanation is to the model being explained, f, in a locality measured by π_x around a specific prediction $f(x)$. In practice, e is typically a generalized linear model and $\mathcal{L}$ is usually defined as a locally-weighted squared loss:

$$\mathcal{L} = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)(f(z) - e(z'))^2 \tag{2}$$

where $z, z' \in \mathcal{Z}$ refer to synthetic data sampled around x and its binary representation, and $\pi_x(z) = exp(\frac{-D(x,z)^2}{\sigma^2})$. D represents some appropriate distance measure according to the data domain (cosine distance for text data and L_2 distance for images). LIME only relies on query-access to the black-box model for constructing pairs of $(z, f(z))$, thereby establishing *model agnosticism.*

[1]Code for our experiments can be found at https://github.com/seansaito/Faster-LIME

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$666.66
https://doi.org/10.1145/nnnnnnn.nnnnnnn

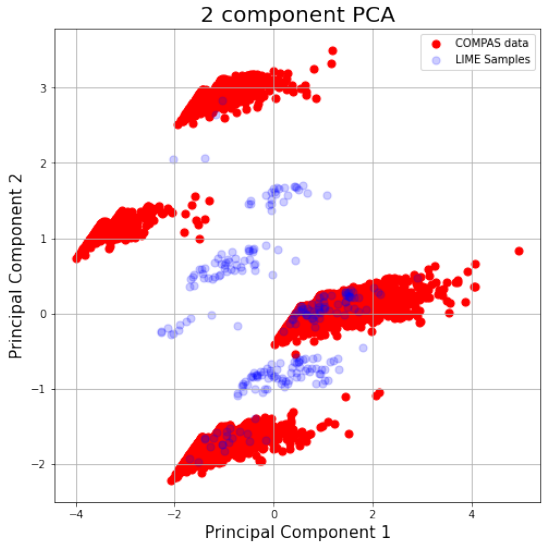

Figure 1: 2-D PCA projection of real data and synthetic data generated by LIME

2.1 Attack on LIME

A notable drawback with LIME's sampling method – and the core problem we seek to address in this work – is that the synthetic samples are typically generated by sampling from a unit Gaussian, $X \sim \mathcal{N}(0, 1)$ around the instance to be explained. Recently, Slack et al. [11] have identified that this leads to the creation of explanations based on identifiably out-of-distribution (*OOD*) synthetic data, as shown in Figure 1. They subsequently proposed an adversarial attack – which we subsequently refer to as the Fooling-LIME attack – that exploits this vulnerability and hides biased behavior from explainers like LIME.

In particular, suppose that an adversary has a biased classifier a which he/she wants to hide from the auditor who uses explainability algorithms to detect biased, harmful behavior. The adversary first constructs a critic c which predicts how realistic a given data point is. This can be done by creating a dataset comprised of real data and synthetic samples generated by LIME, and then training c on this binary classification task. A scaffolding function s is then used to route data points based on predictions of c:

$$s(x) = \begin{cases} a(x), & \text{if } c(x) \geq 0.5 \\ \psi(x), & \text{otherwise} \end{cases} \tag{3}$$

In other words, if c determines x to be an *OOD* sample, i.e. coming from an explainer query, s routes x to ψ, a model which exhibits unbiased behavior. The explainer, given prediction behavior coming from ψ rather than a, would be fooled into determining that the adversary is unbiased and innocuous.

3 IMPROVING THE ROBUSTNESS OF LIME

In this section, we introduce our proposed methods for improving the robustness of LIME.

3.1 Smarter locality sampling

We propose to employ a sampling model which learns to generate realistic data. More specifically, we would like a data sampler g that optimizes the following objective:

$$\begin{aligned} &\text{maximize} && c(g(x)) \\ &\text{s.t.} && |x - g(x)| \leq \delta \end{aligned} \tag{4}$$

In other words, the goal of g is to "fool" the adversarial critic c into thinking that the samples generated by g are indeed real. $x = \{c_1, \ldots c_n, d_1, \ldots, d_m\}$ is a row from some tabular data, which represents a concatenation of continuous variables $C_1, \ldots, C_n$ and discrete variables $D_1, \ldots, D_m$. This optimization problem is similar to that of generative adversarial networks, and we thus use the Conditional Tabular GAN (CTGAN) model [12] as g. CTGAN overcomes the limitations of the vanilla GAN by facilitating sampling from non-Gaussian and multi-modal distributions, creating more realistic tabular data. It generates data by taking in two inputs - a noise vector z sampled from a standard normal distribution and a conditional vector m that selects categories from the discrete variables which influences the output of the generator. As mentioned in [12], we use WGAN loss to train CTGAN:

$$L = \mathbb{E}\left[d(g(z, m))\right] - \mathbb{E}\left[d(x)\right] - \lambda \mathbb{E}\left[\left(\|\nabla_{\hat{x}} d(\hat{x})\|_2 - 1\right)^2\right] \tag{5}$$

where the last term is the gradient penalty from [2] which enforces Lipschitz smoothness of the model weights. During inference, instead of generating synthetic data from a randomly sampled conditional vector, we condition CTGAN with the categorical values of x, the given data instance to be explained:

$$\tilde{X} = g(z, m_x) \tag{6}$$

where m_x is a concatenation of binary mask vectors $m_{d_1}, \ldots m_{d_m}$, each indicating the categorical values of x in one-hot form:

$$\begin{aligned} m^k_{d_i} &= \begin{cases} 1, & \text{if } k = x_{d_i} \\ 0, & \text{otherwise} \end{cases} \text{ for } k = 1, \ldots, |D_i| \\ m_x &= m_{d_1} \oplus m_{d_2} \cdots \oplus m_{d_n} \end{aligned} \tag{7}$$

Using the actual categorical values of x conditions g to generate synthetic data that is more local with respect to x. We further enhance the synthetic neighborhood by utilizing the discriminator $d(x)$ obtained through the CTGAN training process to prune unrealistic samples:

$$\tilde{X}_{filtered} = \{x | x \in \tilde{X}, d(x) \geq \tau\} \tag{8}$$

where τ is a threshold for filtering out unrealistic samples. The final synthetic samples $X_{filtered}$ are then used to optimize a generalized linear model, with the coefficients representing the attributions of each feature to the given model prediction. Table 1 displays sample COMPAS data generated by CTGAN. We subsequently refer to our proposed method as CTGAN-LIME.

4 THREAT MODEL

We now provide context to our work by elaborating the threat model under which we conduct our experiments.

Table 1: Sample COMPAS data generated by CTGAN

age	two_year_recid	priors_count	length_of_stay	race
35	0	12	541	1
20	1	2	248	0
25	1	2	31	1
25	0	4	365	0
43	0	9	62	1

4.1 Adversarial objectives

We suppose an adversary whose goal is to deploy an adversarial model a into some productive landscape. a is adversarial in that it produces biased, unfair predictions based on sensitive features of the data. For example, in the COMPAS dataset, where the task is to predict the likelihood of recidivism for a given convict, a could be implemented as:

$$a(x) = \begin{cases} 1, & \text{if } x_{race} = \textit{"African American"} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

In the process of deploying the model, the adversary may be subject to an auditory process. We assume that the implementation of a is hidden from the auditor to preserve intellectual property. Hence the auditor may use some explainer e to query the model and discover any biased, harmful behavior. The objective of the adversary is then to fool e into thinking that a is innocuous.

4.2 Adversarial settings

We experiment with two adversarial settings, namely *black-box* and *white-box*. In the former, the adversary does not have access to the explainer e nor its implementation; it only receives queries from e. In our experiments, we attack CTGAN-LIME using the original Fooling-LIME attack, which trains its critic c based on synthetic samples generated by the LIME strategy, to measure robustness of CTGAN-LIME against the prevailing adversarial attack.

However, to further fortify our investigation, we also conduct experiments under the white-box setting, where we assume that the adversary now has increased access to e, including its specifications, implementations, and even its parameters. In our white-box experiments, the attacker has full access to the CTGAN generator of CTGAN-LIME. In other words, the attacker generates training data for c *from the same sampler* which the explainability model uses to generate explanations. This is a strictly stronger adversarial setting than the black-box counterpart and provides a stronger evaluation of robustness.

4.3 Evaluation

We measure the robustness of an explainability algorithm based on *top-k accuracy*, or the proportion of explanations which correctly identify the sensitive feature (e.g. race, gender) to be among the top k contributing features for the adversarial model a. In other words, this represents how well the explainability model is able to detect biased behavior.

5 EXPERIMENT DETAILS

In this section we elaborate our experimental details and settings. We measure robustness across the following three datasets:

- **COMPAS recidivism** [4]: A dataset where the task is to predict whether a given criminal defendant will commit another offense. The sensitive feature used by the biased classifier a is *Race*.
- **German credit** [1]: A dataset which classifies how good or bad a given loan applicant is. The sensitive feature is *Gender*, and the biased classifier predicts "good" if *Gender = "male"* and "bad" if otherwise.
- **Communities and crime** [7]: A dataset which measures various socio-economic and law-enforcement variables of communities in the US. As done in [11], we convert the dataset into a classification task whereby a community is labelled "high-crime" if the violent crime rate is above the median and "low-crime" otherwise. We designate the *white population proportion* as the sensitive attribute. The biased classifier predicts "high-crime" if *white population proportion* < 0.5 and "low-crime" if otherwise.

Explanations are generated using the test set of each dataset. For all experiments, we use a Random Forest classifier as the black-box model with which we generate explanations. For each dataset, we vary the number of top features k to be returned by the explainer. Both COMPAS and German credit data contain 9 features each, hence we choose $k = \{1, 3, 5\}$. The communities and crime dataset contains 100 features, and hence the choice $k = \{1, 10, 30\}$ is more appropriate.

6 RESULTS

In this section, we present the results of each of our experiments. Section 6.1 compares the robustness of LIME and CTGAN-LIME. Section 6.2 compares the quality of explanations of each explainer, and Section 6.3 presents a qualitative comparison of the synthetic samples generated by the respective methods.

6.1 Robustness comparison

Tables 2 and 3 refer to the accuracies of the explainability model on predictions generated by the black-box and white-box Fooling-LIME attacks respectively. For each dataset, we measure accuracy for different values of k, or the number of top features generated by the explainer.

CTGAN-LIME achieves higher accuracy than LIME for each dataset, indicating that it can better detect biased behavior of the adversarial model across different datasets. In the white-box attack setting, the attacker has access to the CTGAN model and its parameters, thereby enabling the attacker to execute a more effective attack. Indeed, Table 3 displays a decrease in accuracy of the CTGAN-LIME model. However, we also note that the accuracy CTGAN-LIME is still higher than that of LIME, indicating increased robustness overall; in fact, the Fooling-LIME attack using the CTGAN sampler is slightly more effective than the original attack. Finally, we observe that the discriminator $d(x)$ which filters low-quality samples helps improve the accuracy of CTGAN-LIME further.

Table 2: Top-k accuracy of explainers against the black-box Fooling-LIME attack with varying values of k

	COMPAS			German Credit			Communities		
Explainer / k	1	3	5	1	3	5	1	10	30
LIME	0.00	42.71	70.90	0.00	47.20	65.60	0.00	9.22	35.47
CTGAN-LIME	99.74	99.55	99.68	**61.20**	59.20	61.60	2.20	31.26	48.50
CTGAN-LIME with $d(x)$	**99.94**	**100.00**	**100.00**	60.80	**65.60**	**67.60**	**7.01**	**83.97**	**95.99**

Table 3: Top-k accuracy of explainers against white-box Fooling-LIME attack with varying values of k

	COMPAS			German Credit			Communities		
Explainer / k	1	3	5	1	3	5	1	10	30
LIME	0.00	38.43	69.09	0.00	44.80	64.80	0.00	8.82	36.07
CTGAN-LIME	82.70	88.14	88.74	16.80	52.40	56.80	0.60	10.82	25.45
CTGAN-LIME with $d(x)$	**84.33**	**96.57**	**96.81**	**28.80**	**58.80**	**65.00**	**6.21**	**83.17**	**94.59**

6.2 Explainer comparison

Beyond robustness, we also measure the quality of explanations of CTGAN-LIME and compare them to those of LIME. To this end, we rely on the precision metric proposed in [8]:

$$precision(E) = \mathbb{E}_{Q(x'|E)}\left[\mathbf{1}_{f(x)=f(x')}\right] \tag{10}$$

where E is the explanation and $Q(x'|E)$ is a set of samples $x' \in X$ which satisfy the predicates in E. What precision aims to measure is the agreement between the model's prediction $f(x)$ and the model's prediction on other samples which are similar to x according to the features identified as influential in E. Table 4 reports the precision of CTGAN-LIME on each dataset and shows that CTGAN-LIME can achieve comparable precision to LIME. This suggests that the quality of explanations by CTGAN-LIME is on par with that of LIME.

Table 4: Precision on each dataset

	COMPAS	German Credit	Communities
LIME	71.49	74.58	82.87
CTGAN-LIME	73.27	77.89	80.64

6.3 Qualitative analysis

In Figure 2, the numerical data of the COMPAS dataset is projected onto the first and second principal component. The robust samples drawn from CTGAN (green) more realistically represent the true data (red) than the data generated by the vanilla sampler (blue). The vanilla samples are clustered in a small area, and the scale is roughly equal in the direction of both principal axes. This agrees with intuition, as the data is sampled from a unit Gaussian ball. However, the true underlying data manifold is clearly long tailed and not Gaussian. The CTGAN sampler is hence better able to represent the underlying data distribution and improve robustness.

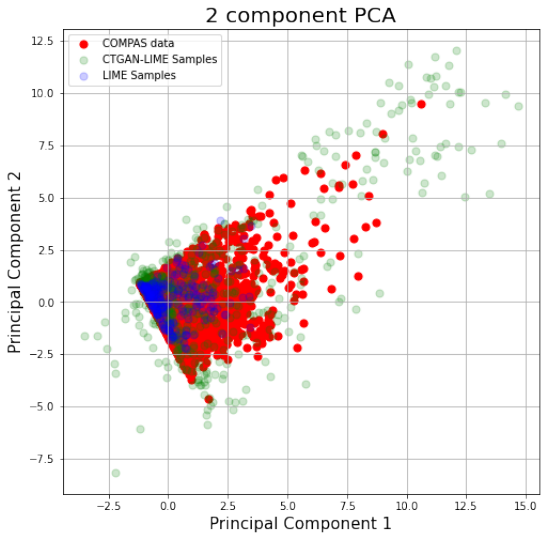

Figure 2: PCA of numerical features for COMPAS dataset

7 CONCLUSION AND FUTURE WORK

We have shown empirical results suggesting that utilizing a generative adversarial network like CTGAN to sample synthetic data for generating explanations can improve robustness towards adversarial attacks. We hope this work will inspire additional efforts towards making explainability algorithms like LIME more robust and reliable. In particular, future work would focus on further conducting empirical evaluations as well as establishing theoretical justifications and bounds for the robustness of explanations.

REFERENCES

[1] Arthur Asuncion and David Newman. 2007. UCI machine learning repository.
[2] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved Training of Wasserstein GANs. (3 2017). http://arxiv.org/abs/1704.00028
[3] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2019. High Accuracy and High Fidelity Extraction of Neural Networks. (9 2019). http://arxiv.org/abs/1909.01838

[4] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. How we analyzed the COMPAS recidivism algorithm. *ProPublica (5 2016)* 9 (2016).
[5] Scott M Lundberg, Paul G Allen, and Su-In Lee. [n.d.]. *A Unified Approach to Interpreting Model Predictions*. Technical Report. https://github.com/slundberg/shap
[6] Milad Nasr, Reza Shokri, and Amir Houmansadr. [n.d.]. *Comprehensive Privacy Analysis of Deep Learning Passive and Active White-box Inference Attacks against Centralized and Federated Learning*. Technical Report.
[7] Michael Redmond and Alok Baveja. 2002. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research* 141, 3 (2002), 660–678.
[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. [n.d.]. *Anchors: High-Precision Model-Agnostic Explanations*. Technical Report. www.aaai.org
[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. (2 2016). http://arxiv.org/abs/1602.04938
[10] Reza Shokri, Martin Strobel, and Yair Zick. 2019. Privacy Risks of Explaining Machine Learning Models. (6 2019). http://arxiv.org/abs/1907.00164
[11] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2019. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (11 2019), 180–186. http://arxiv.org/abs/1911.02508
[12] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular data using Conditional GAN. (6 2019). http://arxiv.org/abs/1907.00503