Recognizing Adversarial Attacks Using LIME

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

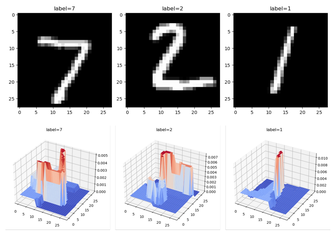
Alexander Phillips, Philipp Stauffenberg
Under Supervision of Dr. Ali Hirsa And Gary Kazantsev at Bloomberg

Motivation:

One of the problems that come with black-box classification models is the difficulty to identify whether the input data might have been subject to an adversarial attack. In many applications, however, this is of utmost importance. For instance, in a self-driving car equipped with automatic traffic sign recognition, incorrect classifications due to a malfunctioning camera or vandalized sign can have fatal consequences. A popular approach to explain the output of black-box classification models is the Local Interpretable Model-agnostic Explanations (LIME) framework. We attempt to use the LIME output to classify whether a model's input has been attacked or not.

Local Interpretable Model-agnostic Explanations (LIME):

Local Interpretable Model-Agnostic Explanations (LIME) is a technique that attempts to explain individual model prediction by approximating the black-box machine learning model. The key idea is to perturb the features, query the model for a prediction on these features and measure the proximity of the perturbed and original features. Then a simple and interpretable surrogate model such as a linear regression can be fitted to obtain each feature's importance for the prediction.

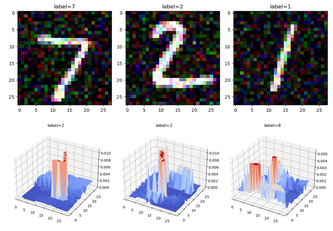


MNIST example images and corresponding LIME output

Adversarial Attacks and LIME:

There are numerous real-world adversarial attacks causing some form of image degradation. These attacks quickly lead to a drastic decline in model performance and increased misclassification. The types of adversarial attacks considered are:

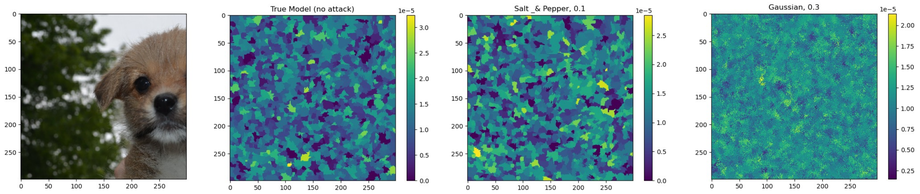
- Gaussian Noise
- Poisson Noise
- Salt and Pepper Noise
- Speckle Noise
- Fast Gradient Sign Method (FGSM) Noise
- Graffiti Noise



Gaussian Noise attacked MNIST images and corresponding LIME output

Hypothesis:

The LIME output for pristine images is smooth while for an attack image it is more erratic and seems randomly distributed across the whole image. Below is an example image from the ImageNet data set along with the corresponding LIME output for the pristine image and different attacked variations. Note that for the attacked variations, the LIME output looks erratic compared to for the true image.



ImageNet example image and LIME output for pristine image and attacked images

Observations like these lead to the hypothesis that if we can find a way to capture the randomness or recognize the erratic behavior it is possible to draw conclusions as to whether an image was attacked by solely using the LIME output.

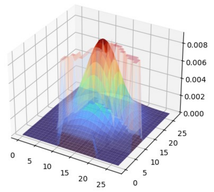
Methodology:

1. **Distribution-Based Approaches:** Treating the LIME output values as an empirical probability distribution and comparing it to other distributions.
2. **Model-Based Approach:** Fitting a convolutional neural network (CNN) to the LIME output and thus treating the output as an image and the detection of an adversarial attack as a classification problem.

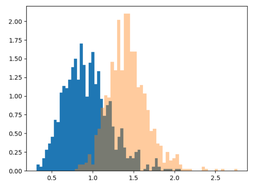
1. Distribution-Based Approaches:

A) Comparing the LIME output empirical distribution to a
- Uniform distribution
- Gaussian distribution

Where in the first case we expect attacked images' LIME output to be closer in distribution and in the second case pristine images' LIME output. The measures used to quantify the difference between the distributions are the Kullback-Leibler (KL) divergence and Gini coefficient.

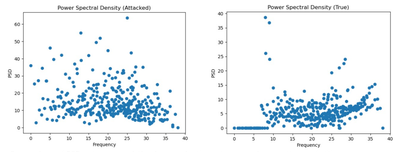


LIME output compared to a Gaussian distribution



Histogram KL Divergence LIME output to Gaussian for pristine (blue) and adversarial Gaussian (orange) MNIST Images

B) 2-d Fast Fourier Transform (FFT) of the LIME output, then converted into a 1-d power spectral density (PSD). This is meant to capture a qualitative 'spikiness' which we observed in the LIME output of attacked images. Such spikiness would show up in the high-frequency band of the PSD.

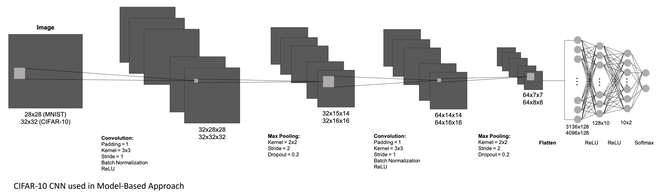


Comparison of PSD between a pristine image (right) and the same image attacked (left)

A+B) We combine the different distribution-based approaches to train a simple logistic regression classifier.

2. Model-Based Approach:

Training a CNN on top of the image recognition model, that uses the LIME output as features to classify into the classes pristine and attacked images.



CIFAR-10 CNN used in Model-Based Approach

Results:

All classifiers of both approaches were trained and tested once per noise type: each time using a data set comprised of the 10,000 standard test images of each data set (MNIST, CIFAR-10, ImageNet) along with one instance of the same set of images after the adversarial attack. The classifiers of the distribution-based approach were tested under 10-fold cross-validation. Importantly: in the classic 'train-validate-test' paradigm our MNIST results should be considered 'validation' results. Our analytical techniques were refined on this MNIST data set. The other results are the true 'test' results; having been achieved 'blindly' after committing to a pipeline. Thus, our approach is closer to a 'train-validate-train-test' data split. For the model-based approach, there was a 70-15-15 train-validate-test split on both data sets, no cross-validation was used, and a spike in validation score was used as a stopping time for training. Only scores on test data sets are shown.

The tables below show the accuracy with which we classify adversarial vs. pristine images, denoted either "Distribution" or "Model" to describe the accuracy of the distribution-based and model-based approaches respectively, and also the accuracy with which the original model correctly predicted the label of the image, evaluated only on the attacked images, denoted "Label".

Noise Type & Parameters	Distribution	Model	Label
Salt & Pepper, 0.05	0.593	0.937	0.355
Salt & Pepper, 0.1	0.700	0.910	0.203
Gaussian, 0.2	0.753	0.500	0.972
Gaussian, 0.3	0.830	0.890	0.379
Gaussian, 0.4	0.830	0.983	0.196
Poisson, 0.4	0.692	0.515	0.971
Poisson, 0.8	0.696	0.504	0.971
Speckle, 0.8	0.690	0.952	0.866
Speckle, 1.2	0.692	0.975	0.784
FGSM, 0.005	0.753	0.500	0.950
FGSM, 0.01	0.804	0.500	0.023
Graffiti, 3 Stickers, 20% area	0.631	0.895	0.523
Graffiti, 4 Stickers, 25% area	0.672	0.878	0.435
Graffiti, 5 Stickers, 10% area	0.585	0.727	0.683

Results MNIST data set

Noise Type & Parameters	Distribution	Model	Label
Salt & Pepper, 0.05	0.625	0.950	0.193
Salt & Pepper, 0.1	0.486	0.850	0.111
Gaussian, 0.2	0.740	0.990	0.164
Gaussian, 0.3	0.840	0.999	0.142
Gaussian, 0.4	0.887	0.978	0.140
Poisson, 0.4	0.577	0.960	0.480
Poisson, 0.8	0.554	0.873	0.609
Speckle, 0.8	0.820	0.986	0.191
Speckle, 1.2	0.879	0.997	0.175
FGSM, 0.005	0.539	0.496	0.398
FGSM, 0.01	0.622	0.927	0.048
Graffiti, 3 Stickers, 20% area	0.613	0.797	0.558
Graffiti, 4 Stickers, 25% area	0.636	0.755	0.502
Graffiti, 5 Stickers, 10% area	0.601	0.501	0.637

Results CIFAR-10 data set

Noise Type & Parameters	Distribution	Model	Label
Salt & Pepper, 0.05	0.641	0.500	0.171
Salt & Pepper, 0.1	0.625	0.967	0.060
Gaussian, 0.1	0.939	0.500	0.471
Gaussian, 0.2	0.992	0.500	0.352
Gaussian, 0.3	0.996	0.950	0.233
Poisson, 0.4	0.848	0.500	0.515
Poisson, 0.8	0.791	0.500	0.550
Speckle, 0.8	0.959	0.778	0.184
Speckle, 1.2	0.981	0.967	0.125
Graffiti, 3 Stickers, 20% area	0.553	0.500	0.407
Graffiti, 4 Stickers, 25% area	0.551	0.500	0.384
Graffiti, 5 Stickers, 10% area	0.539	0.500	0.462

Results ImageNet data set

Conclusion: **We can observe that the LIME output can be used effectively to detect attacked images.** Further: **1)** The performance of the different methods used depends on the type and extent of the attack, with some methods proving extremely effective. **2)** Our results demonstrate that the distribution-based approach produces increasingly strong results as the complexity of the data sets increases, while the model-based approach using a CNN performs better in less complex settings.