Explanation-based Adversarial Detection with Noise Reduction

Anonymous submission

Abstract

Deep Neural Networks (DNNs) have achieved tremendous success in various tasks. However, DNNs expose uncertainty and unreliability against well-designed adversarial examples, thus leading to misclassification. Accordingly, a collection of methods have been proposed to improve the robustness of DNNs by detecting adversarial attacks. In this paper, we combine model explanation techniques and adversarial models, aiming to improve adversarial detection in real-world scenarios. Specifically, we develop a novel adversary-resistant detection framework called EXPLAINER by utilizing the interpretation results extracted from explainable learning models. The explanation model in EXPLAINER produces an explanation map identifying the relevance of input variables in the model's classification result. consequently, the adversarial example can be effectively detected by comparing the explanation results of a given sample and its denoised version, without referring to any prior knowledge of attacks. The proposed framework is thoroughly evaluated on different adversarial attacks. The experimental results show that the proposed approach achieves promising results in white-box attacks.

Introduction

Deep Neural Networks (DNNs) have been widely used in various applications and achieved tremendous success in recent years. For instance, DNNs have achieved state-of-the-art performance in a variety of generative and discriminative learning tasks, including image processing (Du et al. 2019), speech recognition (Maas et al. 2017), drug discovery (Chen et al. 2018), and genomics (Talukder et al. 2021). However, studies have shown that outputs of DNNs can be easily altered by a small perturbation of the input, or even a small perturbation of one pixel (Zhou, Agrawal, and Manocha 2022; Su, Vargas, and Sakurai 2019; Vargas and Su 2019). This sensitivity to small changes in the input makes DNNs vulnerable, limiting the applications of DNNs in high-stake settings, such as self-driving cars (Deng et al. 2020) and malware detections (Sewak, Sahay, and Rathore 2018).

Several approaches for defending against adversarial examples have been proposed. The use of adversarial training or gradient masking to improve the robustness of neural networks is one area of research. Existing research has shown, however, that neural network architectures modified with adversarial training and gradient masking can still be attacked (Carlini and Wagner 2017). Another area of study is adversarial detection, which aims to determine if a given input is adversarial or normal.

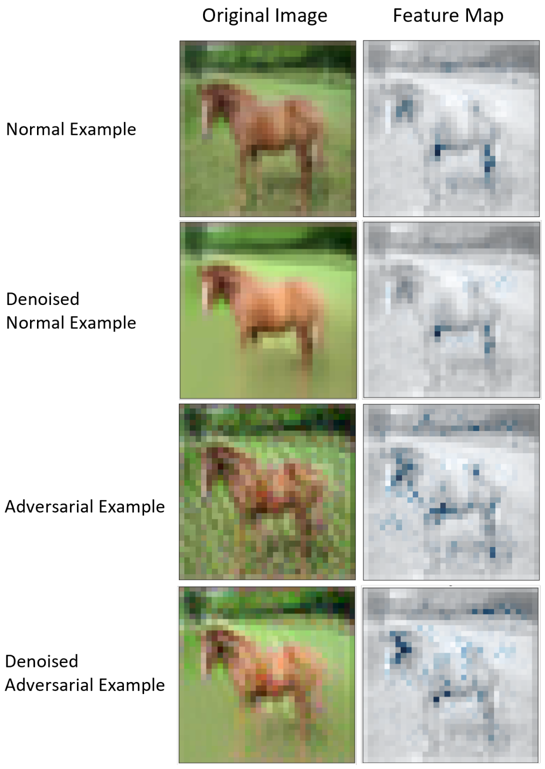

Figure 1: Examples of the Feature maps extracted from a normal example, a denoised normal example, an adversarial example, and a denoised adversarial example. After the noise reduction process, the feature map of the normal example has almost no change, while the feature map of the adversarial image has obvious changes on both the background and object.

However, there are critical questions remain unanswered about what causes the misclassification of adversarial examples. To uncover the causes of adversarial attacks, efforts have been tried to explore the feature differences between normal inputs and adversarial inputs. One possible method is using explanation techniques. Given an image, the result from an explanation model encodes the relevance of pixels for the prediction result, which is commonly referred to as an explanation map. Figure 1 shows that there is human-understandable differences between adversarial examples and normal inputs with Integrated Gradient (Sundararajan, Taly, and Yan 2017). As we can see, normal examples tend to have a more meaningful and continuous explanation map, while adversarial examples tend to have a more discrete explanation map.

As a result, the inconsistencies of extracted features between adversarial examples and normal examples can be utilized in detecting adversarial examples. Song et al. (Song et al. 2018) proposed an Ensemble approach for Explanation-based Adversarial Detection, which uses an ensemble of explanation model wherein each explanation technique provides an explanation map for every classification decision made by a target model. However, their framework requires additional training after extracting the explanation maps.

In this work, we propose an unsupervised adversarial detection method (EXPLAINER) with model explanation. We extract feature maps from explanation models and use the extracted features to determine if an example is normal or adversarial. We evaluate EXPLAINER using five state-of-the-art adversarial attacks on MNIST (LeCun et al. 1998) dataset and CIFAR-10 (Krizhevsky, Hinton et al. 2009) dataset, under white-box threat model. Our experimental results show that we can effectively detect all attacks with fast response.

We summarize our main contributions as follows.

- We develop a novel framework called EXPLAINER based on model interpretation techniques and noise reduction. EXPLAINER utilizes features from the interpretation results using normal examples and adversarial examples.
- We evaluate EXPLAINER on five state-of-the-art adversarial attacks and two image datasets under white-box threat model. The results show that the proposed system can consistently achieve high detection rates with a low false-positive rate.
- We extensively evaluate EXPLAINER with different clustering techniques. Our findings show that EXPLAINER achieves promising results in different scenarios.

Related Work

Adversarial Attack. A number of strategies for developing adversarial examples have been developed. One is Gradient-based attacks (Carlini and Wagner 2017; Goodfellow, Shlens, and Szegedy 2014; Szegedy et al. 2013), which leverage gradient-based optimizations to imitate real-world circumstances. The other one is content-based attacks(Brown et al. 2017; Eykholt et al. 2018), which use perturbations based on the semantics of the input content to simulate real-world scenarios. We focus on five state-of-the-art gradient-based attacks for neural network classifiers in this paper, including the Basic Iterative Method (BIM) (Kurakin et al. 2018), Momentum Iterative Method (MIM) (Dong et al. 2018), and Carlini and Wagner Attacks (CW) (Carlini and Wagner 2017) tailored to L_0, L_2, and L_∞ norms.

Model Explanation. Model explanation provides important insight into the features that are critical to decision-making process of the underlying DNNs. We concentrate on explaining the output of DNN models for a given input using local explainability methods (Baehrens et al. 2009; Lipton 2018). These methods discover which regions in an input image are primarily responsible for the prediction outcome in computer vision models. A saliency map (Simonyan, Vedaldi, and Zisserman 2013), or an explanation map (Dombrowski et al. 2019) more broadly, is a common name for the explanation result.

Adversarial Detection. Adversarial detection is a defense approach with the goal of building a classifier g with a binary output $y \in \{0, 1\}$, where labels 0 and 1 denote that the input instance is normal or adversarial, respectively. Meng and Chen proposed Magnet (Meng and Chen 2017), which uses autoencoders to learn to approximate the manifold of normal examples. Another strategy, known as Feature Squeezing (Xu, Evans, and Qi 2017), recommends reducing an adversary's degree of freedom by smoothing images or reducing their color depth. Noise Reduction is also one of the approaches to get identify adversarial samples. Adaptive Noise reduction (Liang et al. 2018) is used and achieved high accuracy by combining scalar quantization and spatial smoothing.

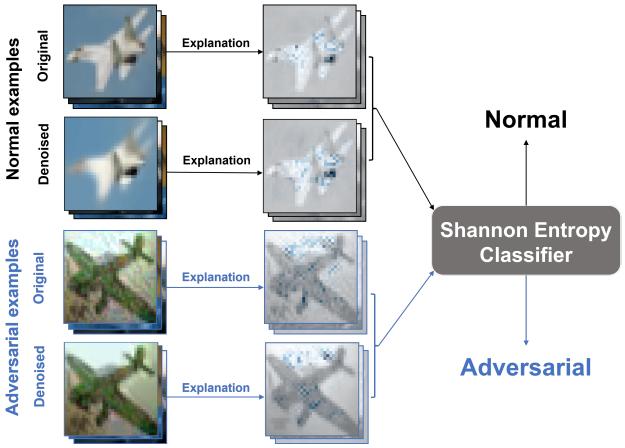

Figure 2: Illustration of the training process of the proposed framework.

Proposed Method

EXPLAINER is a framework that detects adversarial examples based on the features from the model interpretation results. Our hypothesis is that the explanation robustness for the normal examples may not be consistent with the adversarial examples. Fig 2 shows the proposed framework. Given a normal or adversarial example, the first step is using explanation techniques to generate the explanation map from the examples. The next step is using image-denoising techniques to get the denoised version of the example. Then, we apply the same explanation techniques on the denoised image again to get the explanation maps. Finally, we compare the difference in Shannon entropy between the explanation maps for the original image and the denoised image as a classifier to finalize the adversarial detection process. The details for generating adversarial attacks, generating explanation maps, extracting features, image denoising, and calculating Shannon entropy will be introduced in the following sections.

Generation of Adversarial Attacks

The goal of an adversary is to craft a sample that looks identical to a normal sample but is misclassified by the target model. In the context of image classification, this process amounts to finding a small perturbation that, when added to a normal image, causes the target model to misclassify the sample, but remains correctly classified by the human eye. For a given input image x, the goal is to find a minimal perturbation η such that the adversarial input $\tilde{x} = x + \eta$ is misclassified. We consider the following adversarial attacks for testing our frame work.

Basic Iterative Method (BIM) (Kurakin et al. 2018): Basic Iterative Method is the iterative version of FGSM (Goodfellow, Shlens, and Szegedy 2014). Instead of merely applying adversarial noise η once with one parameter ϵ, apply it many times iteratively with small ϵ. This gives a recursive formula:

$$x_0^* = xx_i^* = clip_{x,\epsilon}\left(x_{i-1}^* + \epsilon sign\left(\nabla_{x_{i-1}^*} J\left(\Theta, x_{i-1}^*, y\right)\right)\right) \tag{1}$$

Here, $clip_{x,\epsilon}(\cdot)$ represents a clipping of the values of the adversarial sample such that they are within an ϵ neighborhood of the original sample x. This approach is convenient because it allows extra control over the attack. The adversarial example in Fig 3 is generated by this method.

Momentum Iterative Method (MIM) (Dong et al. 2018): The momentummethod is a technique for accelerating gradient descent algorithms by accumulating a velocity vector in the gradient direction of the loss function across iterations. To generate a non-targeted adversarial example x^* from a real example x, which satisfies the L_∞ norm bound, gradient-based approaches seek the adversarial example by solving the constrained optimization problem

$$*argmin_{x^*} J\left(x^*, y\right), \quad s.t. \left\|x^* - x\right\|_\infty \leq \epsilon, \tag{2}$$

where ϵ is the size of adversarial perturbation.

Carlini and Wagner Attacks (CW) (Carlini and Wagner 2017): The Carlini Wagner attacks are some of the strongest white-box attacks. Consider an input image x and hyperparameter c, the Carlini-Wagner L_2 attack finds a perturbation δ^* for the following optimization problem:

$$*min\|\delta\|_2^2 + c \cdot f(x+\delta) s.t. x + \delta \in [0,1]^n \tag{3}$$

Generation of Explanation

Given a neural network classifier $f(\cdot)$ and an input x, the explanation of the classification is represented as an explanation map denoted by $h : R^d \to R^d$. The explanation map $h(x)$ encodes the relevance score of every pixel in x for the neural network's prediction. We consider the following explanation generation techniques for our proposed framework and the corresponding generated explanation maps are shown in Fig 3. The explanation map are mainly generated through Captum (Kokhlikyan et al. 2020).

DeepLift (Shrikumar, Greenside, and Kundaje 2017): DeepLift (Deep Learning Important FeaTures) is a method for decomposing the output prediction of a neural network on a specific input by backpropagating the contributions of all neurons in the network to every feature of the input. It attributes to each input x_i a value $C_{\Delta x_i \Delta y}$ that represents the effect of that input being set to a reference value as opposed to its original value. DeepLIFT uses a "summation-to-delta" property that states:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o, \tag{4}$$

where $o = f(x)$ is the model output, $\Delta o = f(x) - f(r), \Delta x_i = x_i - r_i$, and r is the reference input.

SHAP (Lundberg and Lee 2017): SHAP (SHapley Additive exPlanations) is a unified framework for interpreting predictions. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. SHAP specifies the explanation as:

$$g\left(z'\right) = \phi_0 + \sum_{j=1}^{M} \phi_j z_j' \tag{5}$$

where g is the explanation model, $z' \in \{0,1\}^M$ is the coalition vector, M is the maximum coalition size and $\phi_j \in R$ is the feature attribution for a feature j, the Shapley values.

Grad-CAM and Guided CAM (Selvaraju et al. 2016): Grad-CAM computes a coarsegrained feature-importance map by associating the feature maps in the final convolutional layer with particular classes based on the gradients of each class w.r.t. each feature map, and then using the weighted activations of the feature maps as an indication of which inputs are most important. To obtain more fine-grained feature importance, the authors proposed performing an elementwise product between the scores obtained from Grad-CAM and the scores obtained from Guided Backpropagation, termed Guided Grad-CAM.

IG (Sundararajan, Taly, and Yan 2017): IG (Integrated Gradients) computes the gradients at all points along a linear path from a baseline $\bar{x}$ to x, and averages them [46]. The baseline $\bar{x}$ can be defined by the user and is generally chosen as a black image. Formally,

$$h(x) = (x - \bar{x}) \odot \int_{\alpha=0}^{1} \frac{\partial f(\bar{x} + \alpha(x - \bar{x}))}{\partial x} \, \mathrm{d}\alpha \tag{6}$$

SmoothGrad (Smilkov et al. 2017): SMOOTHGRAD is a simple method that can help visually sharpen gradient-based sensitivity maps, and it discusses lessons in the visualization of these maps. It creates noisy copies of an input image then averages gradients (or another saliency method) with respect to these copies. This often sharpens the resulting saliency map, and removes irrelevant noisy regions.

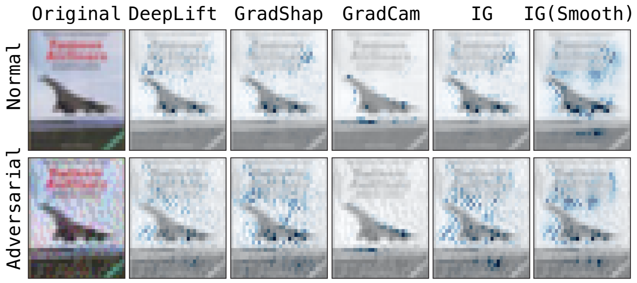

Figure 3: Generation of Attacks and Explanation.

Image Noise Reduction and Shannon Entropy

Fig 1 and Fig 3 show that the generated explanation map for normal, adversarial examples and their corresponding denoised versions are significantly different regardless of the explanation technique we used. This motivates us to transfer the problem of adversarial detection into finding a suitable entropy classifier. In the adversarial attacks settings, given the scenarios that adversarial examples are mixed with normal examples, can we automatically detect the adversarial samples by calculating the change of Shannon entropy between the original and denoised versions when the ground truth of the number and type of attacks are absent? We advocate a two-step approach where image denoising and Shannon entropy are decoupled. First, a non-Local means of denoising (Buades, Coll, and Morel 2011) is employed to obtain the denoised images. Second, we calculate the Shannon entropy for both the original image and its denoised version.

Non-local means denoising is based on a simple principle: replacing the color of a pixel with an average of the colors of similar pixels. It writes

$$NLu(p) = \frac{1}{C(p)} \int f(d(B(p), B(q))u(q)dq,$$

where $d(B(p), B(q))$ is an Euclidean distance between image patches centered respectively at p and q, f is a decreasing function and $C(p)$ is the normalizing factor.

Shannon entropy is named after Boltzmann's H-theorem, Shannon defined the entropy H of a discrete random variable X, which takes values in the alphabet $\mathcal{X}$ and is distributed according to $p : \mathcal{X} \to [0, 1]$ such that $p(x) := P[X = x]$:

$$\mathrm{H}(X) = E[\mathrm{I}(X)] = E[-\log p(X)]$$

We choose the scikit-image (Van der Walt et al. 2014) for calculating the Shannon entropy for the images.

Experiment

Dataset

We evaluated the performance of our detection framework on MNIST (LeCun et al. 1998) and CIFAR-10 (Krizhevsky, Hinton et al. 2009). On MNIST datasets, we trained a CNN-based target model with 60000 examples in the training set and 10000 examples in the validation set. For CIFAR-10, we trained the CNN-based target model with 50000 examples in the training set and 10000 examples in the validation set.

Implementation Details

As described in section , we generate adversarial examples from the testing dataset using five state-of-the-art attacks: BIM (Kurakin et al. 2018), MIM (Dong et al. 2018), and CW (Carlini and Wagner 2017) tailored to L_0, L_2, and L_∞ norm. Moreover, we generate explanation maps, as described in section , using Deeplift (Shrikumar, Greenside, and Kundaje 2017), SHAP (Lundberg and Lee 2017), Grad-CAM (Selvaraju et al. 2016), IG (Sundararajan, Taly, and Yan 2017), and IG with SmoothGrad (Smilkov et al. 2017). PCA and NMF are used to extract the low-dimensional features from the explanation maps. The last step is to cluster using K-means and DBSCAN. If the samples to be tested are not in the same cluster as the majority of normal examples from the training dataset, the samples will be classified as adversarial examples.

Model Evaluation

In this section, we thoroughly evaluate the effectiveness of EXPLAINER in different scenarios and extensively compare its performance with baselines.

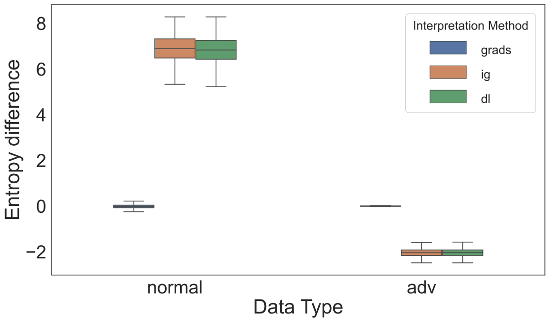

Figure 4: Shannon Entropy comparison between the normal group and adversarial group with different types of explanation method

Explanation Methods We evaluate the Shannon entropy values from different explanation techniques in Fig 4. We compared three explanation techniques here: gradient, integrated gradients, and Deeplift. Both integrated gradients and Deeplift show significant entropy differences among the normal and adversarial groups. Here the results show that the recent explanation techniques could generate explanation maps that capture the noise information noise and could be used for adversarial detection.

Different Adversarial Attacks . We evaluate the distribution of extracted features from different attack techniques in Fig 5. We generate five state-of-art attacks and apply the integrated gradient to generate explanation maps. Fig 5 provide the entropy values. As we could see, all of the attacks could have a significant entropy difference compared to the normal group.

Comparison of Detection Rate .

The detection rate of EXPLAINER is provided in the Table 1.Table 1 compares the performance of EXPLAINER with Magnet (Meng and Chen 2017) and FS (Xu, Evans, and Qi 2017) on MNIST with five state-of-art attack methods. As shown in Table 1, we get around 99% detection rates for MNIST. Compared to other detection methods, our proposed framework is efficient and accurate. Compare to MNIST dataset, CIFAR-10 dataset has more noisy normal images so the noise reduction method should be upgraded. Training a variational autoencoder will be a better choice and this is our future work.

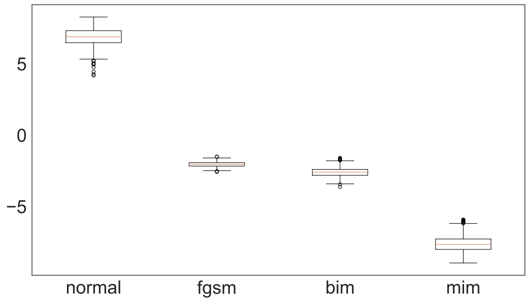

Figure 5: Shannon Entropy comparison between the normal group and adversarial group with different attacks using integrated gradient method for explaning

Table 1: Detection rate of EXPLAINER on whitebox attacks and comparison with state-of-art detection method.

Dataset	Attack	Explainer	MagNet	FS
MNIST	CW_0	99%	86%	91%
	CW_2	100%	86%	100%
	CW_∞	99%	96%	100%
	BIM	99%	100%	98%
	MIM	100%	100%	98%

Conclusion

In this paper, we proposed EXPLAINER, a framework to detect adversarial examples using explanation techniques with noise reduction. The motivation for combining explanation techniques in adversarial attack detection is that distinguishability exists between normal and abnormal explanations for any target class. Experiments showed that our approach is effective against white-box attacks in different datasets under different explanation techniques. We acknowledge the possibility of more sophisticated white-box attacks in the future and hope our work will inspire further research in this direction. We believe our proposed defense can be used in conjunction with state-of-the-art detection methods to boost the detection of adversarial attacks.

References

Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Müller, K.-R. 2009. How to explain individual classification decisions. *arXiv preprint arXiv:0912.1128*.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665*.

Buades, A.; Coll, B.; and Morel, J.-M. 2011. Non-local means denoising. *Image Processing On Line*, 1: 208–212.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. IEEE.

Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; and Blaschke, T. 2018. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6): 1241–1250.

Deng, Y.; Zheng, X.; Zhang, T.; Chen, C.; Lou, G.; and Kim, M. 2020. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE international conference on pervasive computing and communications (PerCom)*, 1–10. IEEE.

Dombrowski, A.-K.; Alber, M.; Anders, C.; Ackermann, M.; Müller, K.-R.; and Kessel, P. 2019. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Du, C.; Zewei, H.; Anshun, S.; Jiangxin, Y.; Yanlong, C.; Yanpeng, C.; Siliang, T.; and Ying Yang, M. 2019. Orientation-aware deep neural network for real image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1625–1634.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; and Reblitz-Richardson, O. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kurakin, A.; Goodfellow, I.; Bengio, S.; Dong, Y.; Liao, F.; Liang, M.; Pang, T.; Zhu, J.; Hu, X.; Xie, C.; et al. 2018. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, 195–231. Springer.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liang, B.; Li, H.; Su, M.; Li, X.; Shi, W.; and Wang, X. 2018. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Transactions on Dependable and Secure Computing*, 18(1): 72–85.

Lipton, Z. C. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Maas, A. L.; Qi, P.; Xie, Z.; Hannun, A. Y.; Lengerich, C. T.; Jurafsky, D.; and Ng, A. Y. 2017. Building DNN acoustic models for large vocabulary speech recognition. *Computer Speech & Language*, 41: 195–213.

Meng, D.; and Chen, H. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 135–147.

Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2016. Grad-CAM: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Sewak, M.; Sahay, S. K.; and Rathore, H. 2018. Comparison of deep learning and the classical machine learning algorithm for the malware detection. In *2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)*, 293–296. IEEE.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153. PMLR.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Song, F.; Diao, Y.; Read, J.; Stiegler, A.; and Bifet, A. 2018. EXAD: A system for explainable anomaly detection on big data traces. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 1435–1440. IEEE.

Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5): 828–841.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Talukder, A.; Barham, C.; Li, X.; and Hu, H. 2021. Interpretation of deep learning in genomics and epigenomics. *Briefings in Bioinformatics*, 22(3): bbaa177.

Van der Walt, S.; Schönberger, J. L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J. D.; Yager, N.; Gouillart, E.; and Yu, T. 2014. scikit-image: image processing in Python. *PeerJ*, 2: e453.

Vargas, D. V.; and Su, J. 2019. Understanding the one-pixel attack: Propagation maps and locality analysis. *arXiv preprint arXiv:1902.02947*.

Xu, W.; Evans, D.; and Qi, Y. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

Zhou, T.; Agrawal, S.; and Manocha, P. 2022. Optimizing One-pixel Black-box Adversarial Attacks. *arXiv preprint arXiv:2205.02116*.