Are Adversarial Examples Suitable To Be Test Suites for Testing Deep Neural Networks

***Abstract*—Most existing works reveal that the deep learning system is extremely susceptible to adversarial examples (AEs), which continually reverberate around the community of DL testing. Consequently, adversarial attacks are exploited to test the robustness of DL model, especially optimized gradient-based technologies in white-box testing. Although AEs have achieved competitive performance in fault revealing-ability and coverage improvement-ability in DL testing, there is little research analyzing the phenomenon theoretically. In this work, we give a formal analysis between gradient-based attack and loss minima of the loss function to prove that powerful adversaries will share similar feature representations with a high probability. Our extensive evaluation and theoretical analysis revealed that (1) the optimized gradient-based technologies can only cover several limited decision logic which is apparently contradictory to the diversity of test suites, (2) the reasons why adversarial examples can increase test coverage, and (3) the weaknesses of AEs by comparing with search-based and fuzz-based test suites generation technologies. Finally, our results prove that AEs can efficiently discover the vulnerability of DL model but are not suitable to explore more inner logic as test suites.**

***Index Terms*—Deep Learning System, Adversarial Attack, Test Suites**

I. Introduction

Advances in deep learning (DL) have achieved remarkable performance in the last decades. Unfortunately, the opacity of DL decisions renders it difficult to understand the trustworthiness of the model in response to open-world scenarios. Moreover, with the DL-based system entering the safety-critical domains [1], [21], the attention to the vulnerability of DL model is much more noticeable than before. Hence, the software engineering community proposed that DL model should have a more rigorous test process and more predictable test outcomes, in order to prove the robustness and security of the model. Following the definition of "ISO/IEC TR 29119-11 Software Testing - Guidelines on the testing of AI-based systems", DL testing should basically satisfy the following requirements: (1) high defects revealing-ability, (2) test suites should be more diverse (i.e., triggering different logic), and (3) test adequacy assurance (i.e., coverage criteria).

However, completely different from the traditional software test, the decision logic of DL model follows a data-driven programming paradigm which is determined by training data rather than code logic. Naturally, the problem of adequately testing DL model has been raised. Prior works claimed DL model is susceptible to adversarial attacks which proposed adversarial attack-based test suites generation technologies to test DL model. The adversarial examples (AEs) can improperly influence the pre-trained model which leads to the wrong classification. Empirical studies [3] suggest that preventing adversaries from computing effective AEs is unattainable until now, which gives adequate confidence to DL testers that AEs can achieve a high fault-revealing rate whatever the robustness degree of given DL models. But it is worth noting that the adversary searches for an optimal adversarial perturbation which misclassifies the DL model from correct label to targeted label. Following this line, the optimized adversary will perturb the original example along with several specific feature representations to minimize the loss between the correct class and the targeted class. Generally, although adversarial examples can efficiently detect the vulnerability of DL model, they are not suitable to be test suites from the perspective of test adequacy. It remains an open question that what is the best test suites generation technology.

In this paper, we firstly investigate the several possible explanations for AEs: that DNNs are too linear [4], that they were trained with an insufficient number of training examples [5], that images contain robust and non-robust features [6], that a new conceptual framework called dimpled manifold model [7] to demonstrate how the decision boundary between classes evolves during training. Based on the aforementioned explanations, we provide a theoretical understanding of adversarial attacks from the perspective of connections between gradients' geometrical properties and local minima of the loss function, which demonstrates that optimized AEs tend to fall into several limited local minima. To systemically understand our claim, we then combine feature visualization technologies [8] and optimized gradient-based attack algorithms to reveal AEs share similar feature representations. Furthermore, we formally analyze why AEs can increase the common coverage criteria (e.g., Neuron Coverage (NC), k-multisection Neuron Coverage (KMNC), Top-k Neuron Patterns (TKNP)). Thirdly, to explore better test suites generation technologies, we compare the search-based, fuzz-based, and adversarial-based test suites generation approaches. Our evaluation results suggest that multi-objective search techniques can produce more diverse test suites and cover more decision logic.

These findings prove that the AEs are suitable to be test sites that cannot produce diverse output behavior. This paper makes three keys contributions:

• We present theoretical analysis that the optimized AEs have similar feature representations, which is the cornerstone to prove that the existing AE algorithms cannot produce more diversity.

• We formally analyze the reasons of AEs can increase common coverage criteria, including random initialization, as

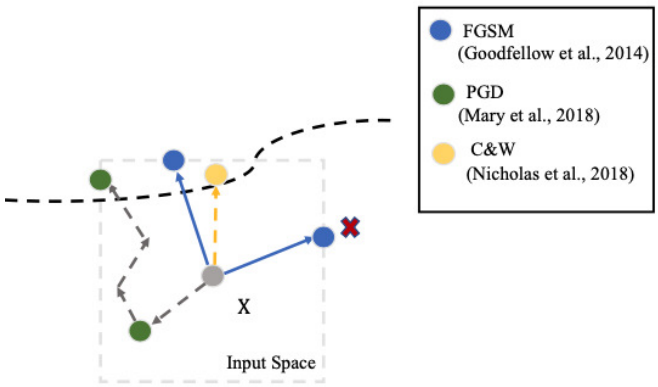

Fig. 1. A conceptual figure of adversarial attack

well as fixed optimized iterations, fixed perturbed step size, and normalization of the gradient-step.

• We give empirical experiments and formal analysis to demonstrate the following findings: 1) the inherent property of AEs is contradict to DL testing in terms of triggering more logics; 2) AEs can indeed increase the state-of-the-art coverage; 3) search-based technologies have promising performance compared with AEs.

II. BACKGROUND AND RELATED WORK

A. Notation

In this section, we first define several notions and abbreviations as follows. For a $\mathcal{K}$-class ($\mathcal{K} \geq 2$) classification task, given a training data $\mathcal{S} \in \{x_i, y_i\}_{(i=1,2,\ldots,n)}$ and test data $\mathcal{T}$ which are assumed to be independent and identically distributed(IID). In the training phase, let $\mathcal{F}(\Theta, x, y)$ represent the trained model with parameter Θ predicts the class of inputs example x. Normally, the model parameters Θ update by back-propagation to minimize the training loss $\mathbf{L}(\mathcal{F}_\Theta(x), y)$, such as the most commonly used cross entropy $-\log\left(\frac{e^{f_\theta(x)_y}}{\sum_j e^{f_\theta(x)_j}}\right)$. When facing a white-box attack, the parameters of trained model $\mathcal{F}(\Theta, x, y)$ are exposed to adversaries, such as weight $\mathbf{W}$ and gradient $\mathbf{G}$ of the trained model. Hence, the adversary can utilize the gradient of the targeted model to maximize the classification loss $\mathbf{L}(\mathcal{F}_\Theta(x+\delta), y)$ by adding well-crafted perturbation δ, whilst $||\delta||_p \leq \epsilon$ is meant that adversarial examples(AEs) must ensure perceptual similarity under ϵ - constraint measured by $\mathcal{L}_p$ norm. As a result, the targeted neural network will misclassify the adversarial input $x+\delta$ as y_{err}, that is $\mathcal{F}_\Theta(x+\delta) = y_{err} \neq \mathcal{F}_\Theta(x)$.

B. Adversarial Attack Algorithms

Many methods for generating such adversarial examples (i.e. optimizing a perturbation ϵ) have been proposed. We now summarize three state-of-the-art adversarial example generation methods (e.g. FGSM [4], PGD [9], C&W [2]).

Fast Gradient Sign Method(FGSM) Goodfellow et al. proposed FGSM to find adversarial perturbation δ by computing a single step along the input x gradient direction. Formally, the adversarial perturbation δ is generated via:

$$\delta = \epsilon \cdot sign(\nabla_x \mathbf{L}(\mathcal{F}_\Theta(x), y)) \tag{1}$$

in which ∇_x is the partial derivative of x, $sign(\cdot)$ is a sign function, and ϵ is sufficiently small perturbation parameter such that $||\delta|| \leq \epsilon$. The original formulation is only considering L_∞ norm, while there also exists L_2 norm(i.e. $||\delta|| \leq \epsilon$) version of FGSM as following:

$$\delta = \epsilon \cdot \frac{\nabla_x \mathbf{L}(\mathcal{F}_\Theta(x), y)}{\nabla_x ||\mathbf{L}(\mathcal{F}_\Theta(x), y)||_2} \tag{2}$$

Projected Gradient Descent(PGD) It is worth noting that FGSM has a significant disadvantage that its target is creating adversarial perturbations quickly by a single step, rather than optimal perturbation. Many FGSM-variants existing, among them Projected Gradient Descent(PGD) is the iterative optimization method and consider the most powerful first-order adversarial attack. Given a constraint set $S_\epsilon = \{z : d(x,z) < \epsilon\}$, we optimize by following steps:

$$x'_0 = x_0, \tag{3}$$

$$x_{i+1} = Proj_{(S_\epsilon)}\{x_i - \alpha(sign(\nabla_{x_i} \mathbf{L}(\mathcal{F}_\Theta(x_i), y_t)))\} \tag{4}$$

Here, x_0 is random initialization for original x by adding random noise in the range of $(-\epsilon, \epsilon)$, x_{i+1} denotes the generated adversarial examples at the x_i iteration. $Proj_{(S_\epsilon)}$ represents the projection function by clipping $d(x,z)$ into $[x-\epsilon, x+\epsilon]$, α denotes step size. After i iterations of perturbation, $Proj$ function projects the AE back on the ϵ-ball.

Carlini and Wagner attack(C&W) Unlike the fixed step such as FGSM and PGD, Nicholas et. al proposed gradient-based optimization attack C&W to seek the minimal perturbation to fool the classifier. Until now, C&W is extensively regarded as one of the strongest adversarial attacks. More formally, we consider the following definition:

where α is a scaling constant to steer trade-off between the objective function and l_p-norm, $\mathbf{L}$ is a suitable objective function which has seven alternative versions in Carlini and Wagner [2]. The following objective function has optimum performance.

$$\mathbf{L} = max(\max_{i \neq t}(Z(x+\delta)) - Z(x+\delta)_t, -\kappa) \tag{5}$$

where $Z(*)$ is the softmax function of neural network, κ is a tuning variable controlling misclassification confidence. More specifically, for the reason of eliminating the possibility of getting stuck in extreme regions, C&W attack introduces a new variable w and optimizes over the following setting.

$$\delta_i = 1/2(tanh(w_i) + 1) - x_i \tag{6}$$

C. DL Testing and its limitation

The striking rise of demand to incorporate deep learning (DL) components into security-critical domain [22], [33] requires rigorous testing before deployment. Both the security and software engineering community have developed several most-existing technologies for testing DL-based systems. Testing is checking whether the actual outputs match the expected ones and ensure that the system is defect free from the perspective of software engineering, which comprises following key test procedures (i) requirement analysis, (ii) test suites generation, (iii) test oracle generation, and (iv) test evaluation. More recently, many efforts have been put on generating test suites [10]–[14], [16], [17], which can mainly be divided as two research direction. One is that create adversarial examples by adding unobserved perturbation to a fixed input. More specifically, there are several optimized gradient-based adversarial attack techniques have successfully proved that neural network is extremely susceptible to adversarial examples, such as PGD, C&W. Another line is coverage-based test suites generation approaches which intend to explore more internal decision logic of the neural network.

Inspired by the first research direction, there has been a growing body of research carrying on gradient-based white-box techniques for testing the robustness of DL model [24]. DeepGauge [17] proposed a multi-granular coverage criterion based on NC which measures the proportion of neurons activated in a neural network, including k-MNC, SNAC, NBC, Top-k NC and Top-k NP. By utilizing standard adversarial attacks to generate test suites, it finds that AEs can significantly increase both the neuron-level and layer-level coverage. Furthermore, Kim [14] proposed surprise adequacy (SA) for testing DL system for guiding the selection of individual test suite. The target of SA is measuring the surprise (diversity and novelty) of the newly generated test suite compared with the training data. The experiment result revealed that the test suites with higher SA values are harder to correctly classify. Recently on-going work [18] reveals the relationship between neuron coverage and test deep learning, and claims that the NC is not really closely related to system robustness, but without any doubt that the adversarial examples can definitely increase the neuron coverage. Hence, the demand for capturing the intrinsic properties exhibited by neural networks gives rise a natural research opportunity. DeepImportance [19] excogitated an Importance-Driven test criterion that reveals the causal relationship between internal neurons and the behavior of neural networks by relevance propagation. The influential neurons have been quantitatively divided into an automatically-determined finite set of clusters to cover their behavior to a sufficient level of granularity. This effort utilizes the adversarial examples as test suites and calculates the combinations of influential internal neuron clusters coverage. Most recently, NPC [20] claimed that coverage criterion should be mapped into internal decision logic, hence this work extracted Critical Decision Paths(CDPs) to represent the control flow of the DNN. Empirical study shows that error rate has a positive correlation with NPC coverage by feeding AEs to DL model.

As aforementioned efforts, the combination between adversarial examples and coverage criteria shows its usability to test the robustness of DL system. However, it remains opaque why these different coverage criteria could achieve a similar increasing trend when feeding AEs as test suites.

D. The Explanation to Adversarial Examples

The adversarial attack is one of the concerning hindrances to limit the deployment in safety-critical applications. Hence, another line of research is exploring the cause of adversarial examples and visualizing them [**?**], [**?**], [**?**], [**?**], [8].

Adversarial Example Explanations: There has not been achieved consensus on the reason for existing AEs, but the common explanations are from the following perspectives: the curse of dimensionality [**?**], [4], the properties of data [**?**], [**?**], [6] and model itself [**?**], [**?**]. More specifically, Goodfellow et al. [4] argue that the main reason for AEs comes from the linear part of the model, in which the perturbation can significantly magnify as the dimension of weight increases. Following this, Khoury et.al [**?**] provide theoretical and empirical evidence that AEs arise when data lies on a lower dimensional manifold in a high dimensional space. Other studies show that AEs could arise due to properties of the model itself, that the non-complexity of models [**?**], [**?**], that the selection of activation function [**?**], [**?**], that the relationship between robustness and network width [**?**], [**?**]. Apart from the model, some papers explore to explanations in the data, such as insufficient training data, and the extremely exceptional samples in the data set. Further studies point out that the existence of robust features (useful and semantic-related) and non-robust features (highly useful yet perturbation-sensitive) in any figures, while non-robust features directly cause the presence of AEs. Furthermore, AEs leave the manifold of training data has been proposed by previous works, recent follow-up work provides the Dimpled Manifold Model framework by Adi Shamir [7] to describe the existence and properties of AEs from a new geometric perspective and also provides a more complete picture of this line of research. In this work, we take the two complementary explanations from as basic assumptions.

Visualization Technologies: In our works, the visualization technologies including t-Distributed Stochastic Neighbor Embedding (t-SNE) and Shapley values, are utilized to further support our theoretical analysis. T-SNE [**?**], [8] is a technology integrating dimension reduction and visualization. The Kullback–Leibler divergence (KL divergence) is regarded as a loss function between similarity distribution over instance pairs in the high-dimensional space and a probability distribution over the points in the low-dimensional map. It then minimizes the loss function to optimize the visualization result. Shapley values [**?**] is a game theoretic approach to provide human-interpretable visual explanations towards the decision of the model.

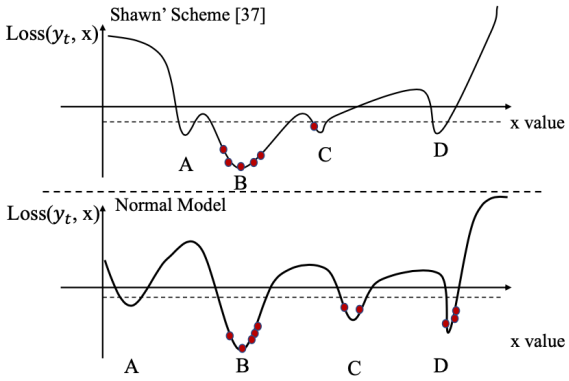

Fig. 2. An example of conceptual figure

III. Research Questions

RQ1. What is the difference between the adversarial attack and DL test? Previous works assume that the proposed coverage criteria have a positive correlation with model quality. When AEs can efficiently enhance the coverage criteria, this leads to the inescapable conclusion that AEs can be regarded as test suites. Hence, AEs are empirically utilized to test the DL model in existing schemes to certify the availability of proposed coverage criteria. Following this logic, adversarial attacks are not naturally designed to test models, there is the existing inherent difference compared to DL test.

RQ2. The relationship between adversarial examples and coverage criteria Without any doubt that adversarial examples can significantly increase the coverage criteria such as NC or NC-variants, but there is no clear explanation for the empirical experimental result. Hence, in this paper, we try to explain the phenomenon theoretically from the perspective of an optimized-based adversarial attack algorithm.

RQ3. What technology is better to generate test suites for DL testing? Most existing works proposed several test suites generations (e.g. fuzz-based [13], [25], search-based [26], [27] and adversarial attack-based), there are a little works give a plausible comparison results under the requirements from testing.

IV. Study Methods

This section describes datasets, DNN models and optimized-based adversarial attack algorithms used for our empirical experiments. And then we also give formal formula derivation to prove Theorem 1. Finally, we design the experiments for our research questions.

A. Basic Assumption

Assumption 1: The Robust Features Model

The phenomenon of adversarial examples is a natural consequence of the presence of highly predictive but non-robust features in standard datasets. Specifically, the model inevitably exploits non-robust features and robust features to predict, while the number of highly predictive but non-robust features and robust features is limited. For details, please refer to [6].

Assumption 2: The Dimpled Manifold Model The Assumption comes from the Dimpled Manifold Model [7] of Adversarial Examples in Machine Learning that introduce a new conceptual framework for how the decision boundary between classes evolves during training called the Dimpled Manifold Model.

Both assumption 1 and assumption 2 are not contradictory but complementary, because assumption 1 focuses on what DL has learned from training which is a more human-centric approach, while assumption 2 concentrates on utilizing geometric intuition to show how the decision boundary changed between classes evolves during the training process. Clearly, We can thus associate the off-manifold dimensions with the non-robust features. On the other hand, the on-manifold dimensions correspond more closely to the human-noticeable robust features.

B. Formal Theoretically analysis

In this part, we theoretically prove that the optimized gradient-based adversarial examples are not inappropriate to be test suites in respect of test coverage.

Specifically, our analysis has two steps, one is that establish a theorem on the basis of defending against adversarial attacks by the honeypot approach [23]. Given its mathematical proof and preliminary results showing that an optimized attacker will fall into an artificial large local minimum of the loss function with high probability. In other words, the honeypot focuses on exploring the relationship between the optimization attack algorithms and trapdoor, while our work focuses on exploring the relationship in AEs generated by attack optimization algorithms. Hence, our theorem rising: given a normal model, the optimized attacker will probabilistic search the perturbation at multiple local minima of the loss function and the conceptual figure is shown in Fig.2.

Definition 1. *Given a trained model $\mathcal{F}_\Theta$, $\exists||\Delta||_p \leq \epsilon$ added to the model input such that any arbitrary input $x \in \mathcal{T}$ where $\mathcal{F}_\Theta(x) = y_t$, we have $Pr(\mathcal{F}_\Theta(x+\Delta) \neq \mathcal{F}_\Theta(x) = y_t) \geq 1-u$ & $\min \mathbf{L}(\mathcal{F}_\Theta(x), y_t)$ which means the smallest loss at x is to label y_t. Here, $u \in [0,1]$ is a small positive constant. An optimized attack strategy $\mathcal{A}(*)$ attempts to find the minimal perturbation δ such that $\mathcal{F}_\Theta(x+\delta) \neq \mathcal{F}_\Theta(x) = y_t$, which called $\mathcal{A}(*)$ is $(v, \mathcal{F}_\Theta, y_t)$-effective on x if $Pr(\mathcal{F}_\Theta(\mathcal{A}(*)) \neq \mathcal{F}_\Theta(x) = y_t) \geq 1-v$. Here, v is a small positive constant.*

The following theorem shows that a trained model $\mathcal{F}_\Theta$ enables an adversary to launch a successful adversarial input attack. The detailed proof is demonstrated as follows.

Theorem 1. *Let $\mathcal{F}_\Theta$ be a trained model, $g(x)$ be the model's feature representation of input x, if the feature representations of adversarial input $g(A(x)) = g(x+\delta)$ and $g(x+\Delta)$ are similar, such as the cosine similarity $cos(g(A(x)), g(x+\Delta)) \geq \zeta$ and $\zeta \to 1$, we have the attack $A(x)$ is $(u, \mathcal{F}_\Theta, y_t)$-effective.*

Proof: The theorem assumes that $\mathcal{F}_\Theta(x)$ is composition of non-liner feature representation $g(x)$ and a linear function:

$\mathcal{F}_\Theta(x) = g(x) \circ \mathcal{L}$. Thence, after adding a perturbation Δ to model input, we have:

$$\min \mathbf{L}(\mathcal{F}_\Theta(x+\Delta), y) \;\&\; Pr(\mathcal{F}_\Theta(x+\Delta) \neq \mathcal{F}_\Theta(x) = y_t) \geq 1-u \tag{7}$$

$\mathbf{L}(\mathcal{F}_\Theta(x+\Delta), y)$ can be approximated with a linear function surrounding $x_0 = x + \Delta$ by computing its first-order Taylor expansion:

$$\mathbf{L}(\mathcal{F}_\Theta(x+\Delta)) \approx \mathbf{L}(\mathcal{F}_\Theta(x)) + \nabla_{\mathbf{x}} \mathbf{L}(\mathcal{F}_\Theta(x))(x - x_0) \tag{8}$$

Here $\nabla_{\mathbf{x}} \mathbf{L}(\mathcal{F}_\Theta(x))$ represents the partial gradient from x toward $x+\Delta$, which implies that the most effective direction of locally changing the prediction result and minimizing the loss by perturbing input away from x. Then,

$$\nabla_{\mathbf{x}} \mathbf{L}(\mathcal{F}_\Theta(x)) = -\nabla_{\mathbf{x}} \ln \mathcal{F}_\Theta(x) = \eta \tag{9}$$

where η represents the gradient value to label y_t for given x. Since $\nabla_{\mathbf{x}} \ln \mathcal{F}_\Theta(\mathbf{x}) = \nabla_{\mathbf{x}} \ln(g(x) \circ \mathcal{L}) = c \nabla_{\mathbf{x}} \ln g(x) \circ \mathcal{L}$, we can focus on partial derivative of $g(x)$ while discarding $\mathcal{L}$ and constant c. We have:

$$\frac{\partial[\mathbf{L}(\mathcal{F}_\Theta(x+\Delta)) - \mathbf{L}(\mathcal{F}_\Theta(x))]}{\partial x} = \frac{\partial[\ln g(x) - \ln g(x+\Delta)]}{\partial x} \tag{10}$$

$$Pr\left[\frac{\partial[\ln g(x) - \ln g(x+\Delta)]}{\partial x} \geq \eta\right] \geq 1 - u \tag{11}$$

Thence, under the condition of $cos(g(A(x)), g(x+\Delta)) \geq \zeta$ and $\zeta \to 1$, we assume that $g(A(x) = g(x+\Delta) + \gamma$ where $|\gamma| \ll |g(x+\Delta)|$. Next, we follow the proof [23] which implies that gradient-based adversary can approach the minima of loss $\mathbf{L}(\mathcal{F}_\Theta(x+\Delta))$ when $g(x+\epsilon)$ has similar feature representation with $g(x+\Delta)$. Specifically, because of $|\gamma| \ll |g(x+\Delta)| \;\&\; |\gamma|$ is independent on x, we have

$$\frac{\partial(g(x+\Delta)+\gamma)}{\partial x} = \frac{\partial g(x+\Delta)}{\partial x} \tag{12}$$

$$\frac{1}{g(x+\Delta)+\gamma} \approx \frac{1}{g(x+\Delta)} \tag{13}$$

Finally, by substituting the known conditions into the $Pr\left[\frac{\partial[\ln g(x) - \ln g(x+\epsilon)]}{\partial x} \geq \eta\right]$, we have

$$\begin{aligned}
&Pr\left[\frac{\partial[\ln g(x) - \ln g(x+\epsilon)]}{\partial x} \geq \eta\right] \\
=&Pr\left[\frac{1}{g(x)}\frac{\partial g(x)}{\partial x} - \frac{1}{g(x+\epsilon)}\frac{\partial g(x+\epsilon)}{\partial x} \geq \eta\right] \\
=&Pr\left[\frac{1}{g(x)}\frac{\partial g(x)}{\partial x} - \frac{1}{g(x+\Delta)+\gamma}\frac{\partial(g(x+\Delta)+\gamma)}{\partial x} \geq \eta\right] \\
\approx&Pr\left[\frac{1}{g(x)}\frac{\partial g(x)}{\partial x} - \frac{1}{g(x+\Delta)}\frac{\partial(g(x+\Delta))}{\partial x} \geq \eta\right] \\
=&Pr\left[\frac{\partial[\ln g(x) - \ln g(x+\Delta)]}{\partial x} \geq \eta\right] \\
\geq&1 - \mu.
\end{aligned} \tag{14}$$

When adversarial input $g(a+\epsilon)$ has similar feature representations with the local minimum of loss function, the attack strategy is $(u, \mathcal{F}_\Theta, y_t)$-effective. Hence, we claimed that existing powerful optimized attack searches for small perturbations to minimal loss along the gradient, which means the adversarial inputs generated by the same label have a greater chance to share similar feature representations. The verification of our theoretical analysis will show in the following empirical experiments of RQ1 and RQ2.

V. Empirical Result

A. Experimental Design

Adversarial attacks aim at creating perturbed inputs to achieve two primary objectives-maximizing loss while keeping l_p-norm distance from the original inputs. To answer RQ1, we first generate a large set of adversarial examples under l_2 version of adversarial attack technologies for highlighting its visually noticeable difference. Then, we perform the visualization technologies (e.g., t-SNE and Shapley values) to AEs, which can illustrate the change of feature representations between normal examples and AEs. This process aims to verify the conclusion that the powerful adversarial attacks will share similar adversarial feature representations with a high probability. Ideally, we would expect to see the feature representations of adversarial perturbations will converge to several specific patterns. To answer RQ2, we calculate the coverage metric value (i.e., NC, KMNC, TKNP, NPC) change by adding AEs generated by FGSM, PGD and C&W. Second, we compare the growth of coverage under different adversarial attacks and try to explain the phenomenon theoretically. To answer RQ3, we conduct the following experiments. Firstly, we utilize different test generation techniques such as Fuzz-based, search-based (coverage criteria-based), and PDG (gradient-based) to generate test suites, against the same model. Secondly, we calculate the defects-revealing capability of different test generation techniques and compare the quality of test suites by comparing their (similarity or logic coverage).

B. Experiments Setup

We follow the configuration of Reference [17] and [18]. More details about the architecture of all DNNs under test set and attack configurations are exhibited in Table.I.

TABLE I
List of abbreviated names

Variable	Values
Adversarial attacks	FGSM, PGD, C&W
DL models	Lenet1, Lenet4, ResNet56, DenseNet121
Datasets	MNIST, CIFAR-10
ϵ Limit (PGD)	1/255, 2/255, 3/255, 4/255, 5/255
Confidence (C&W)	0 ,20

Datasets and DNNs Architectures

MINIST [28] is a well-known handwritten digit dataset that seeks to recognize 10 classes of 28*28*1 gray-scale pixel images. For this dataset, both well-recognized neural network architecture Lenet1 and Lenet4 [29] were utilized

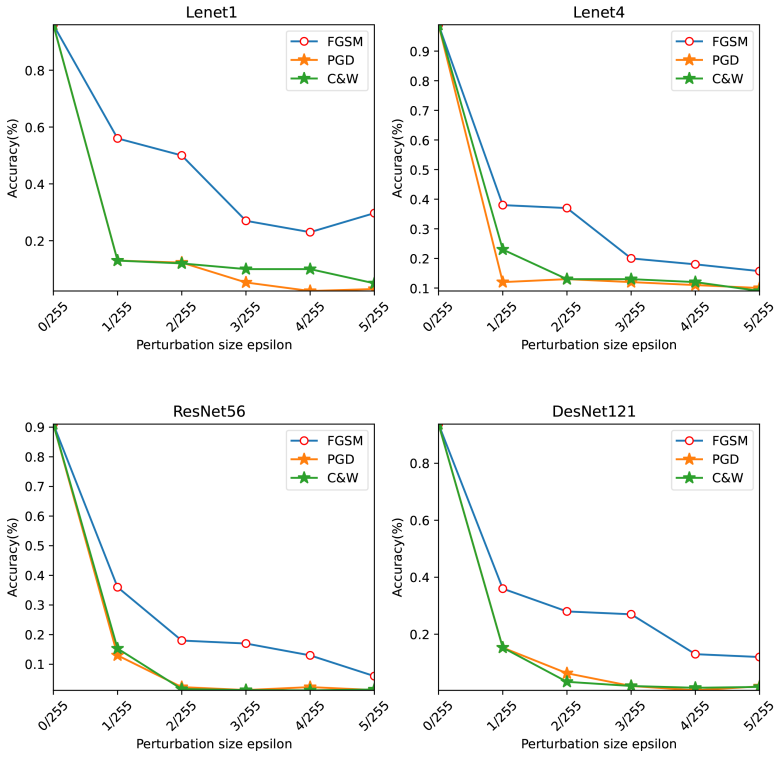

Fig. 3. The classification accuracy of four classifiers on AEs crafted by FGSM, PGD and C&W with increasing perturbation size. Strong attacks including PGD and C& W can succeed most of the time (more than 95%). All AEs were generated in a white-box setting.

in our experiments. CIFAR [30] is a 32*32*3 RGB image dataset for 10 mutually excursive classes including airplanes, cars and et.al. We select two CNNs models(ResNet56 [15], DenseNet121 [31]) which achieve competitive performance under this dataset.

Experiment Configuration and Results

When it comes to attack configurations, FGSM, PGD and C&W attack technologies were utilized to generate the adversarial examples.

C. RQ1: Relationship between AEs & DL testing

Many efforts [17], [20] utilize the AEs to test the robustness of DL model, but without declaring the usage scope. It definitely can generate test suites to fool the model, but the difference is that adversarial attack is ad-hoc and aim to generate similar AEs. Following the theoretical analysis of Theorem 1, we proved that if the feature representations between adversarial input $x + \epsilon$ and well-crafted perturbation $x+\Delta$ which results to minimal loss are similar, the adversarial attack technologies are efficient. In other words, existing powerful optimized attack searches for small perturbations to minimal loss along the gradient, which means the adversarial inputs generated by the same label have a large possibility to share similar feature representations. Hence, the external expression of the similar feature representation will exhibit similar neuron activation patterns. While DL testing aims not only to generate more incorrect inputs but also systematically explore more decision logic of the model [20]. From this perspective, the adversarial examples can only cover limited several decision logics in the theoretical proof.

We conduct the empirical experiments on the MNIST and CIFAR datasets and generate the adversarial examples by l_2 version of PDG and C&W attack technologies in Fig. 4 and Fig. 5. The columns in both two figures from left to right are the clean image, the adversarial image, and the adversarial perturbation (maximally amplifying its entries to the full range of [0, 1] to make it visually clearer). From the simulated results, we found that the powerful attacks (e.g. PDG or C&W) are more potential to craft the semantically-interpretable perturbation under l_2 norm, which also can be reflective of the Assumption. 1 and Theorem. 1. The latent feature representations which the human visual system would understand are regarded as robust features. Under the constraint of the minor magnitude of C&W perturbation distance, this attack procedure will change the pixels that are most influential for the model's predictions, which leads to feature representations of adversarial perturbation aligning better with human perception. Hence we can extract the human-aligned information by amplifying adversarial perturbation. More specifically, in Fig. 4, the PGD attack attempts to close the "7" figure's circle to create a "9" figure, and blacken the loops in the "4" to get a "9" on MNIST. Additionally, this phenomenon isn't just unique to MNIST as PGD also can produce the same kind of visible perturbation on a CIFAR trained model. On the left of Fig. 4, PGD attempts to add two tires to the boat which makes it look more like a truck. In

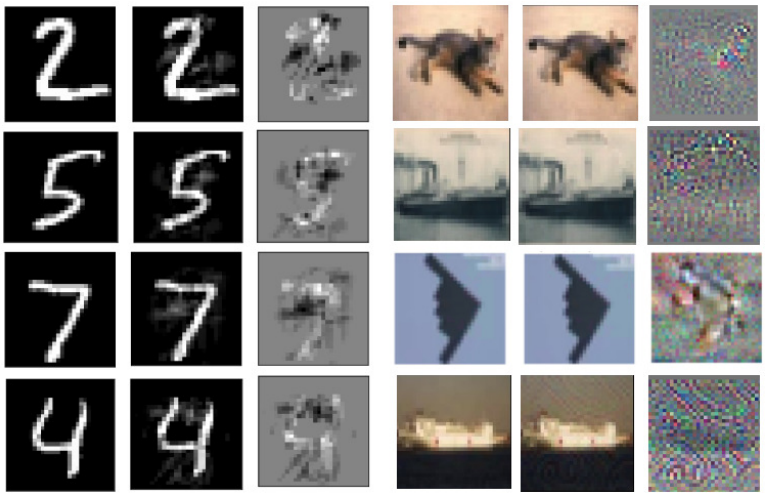

Fig. 4. PGD attack to MNIST and CIFAR

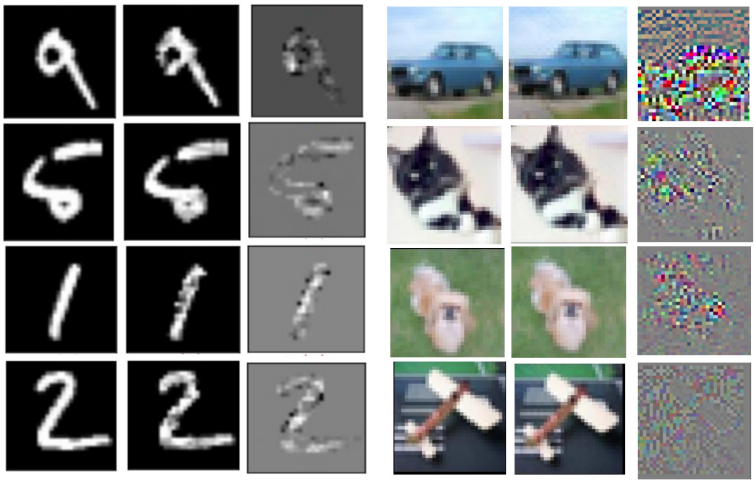

Fig. 5. C&W attack to MNIST and CIFAR

Fig. 5, the C&W attack also has such similar phenomenon, that enlarges the ear of cat to construct the wings of airplane, that the tail of dog is emphasized to turn the dog into a horse. Our results evidently demonstrate that the powerful attacks tend to produce plausible AEs following that gradient, which looks more semantically meaningful rather than randomly-looking to humans. Hence, we believe that AEs will activate inherent neuron patterns for every class.

To better illustrate the difference between adversarial and normal feature representations, we visualize the 2D embeddings of the deep features for adversarially perturbed airplanes using t-SNE in Fig. 6 on ResNet56. More specifically, low dimensional representations of PGD and C&W attack are more clustered rather than scattered in each class's representation space (e.g. FGSM). In addition, we also visualize the projected representations of adversarial truck, ship, horse and airplane under PDG and C&W attack. Clearly, the projected adversarial representations of each class tend to flock together, which also proves the correctness of Theorem 1. Furthermore, we utilize the visualization technology (e.g. Shapley values) to present the strength for feature contribution to the final output in Fig. 8. The "blue" and "red" colors indicate the negative and positive contributions to outputs, respectively. Clearly, we observe that PGD and C&W AEs are basically aligned with semantic understanding from human vision. For instance, the core positive contribution of mistaking "3" for "6" is fill-in the bottom ring of the "3" figure. Another finding is that the main positive contributing features that lead to model misclassification are often clustered together.

In conclusion, by visualizing adversarial perturbation, normal/adversarial feature representations, and important features to prediction, we verified theorem 1 and claimed that the inherent proprieties of optimized adversarial attack algorithms determined that AEs can generate similar incorrect inputs or neuron activation patterns but not more diverse, which is contradictory with testing requirements in terms of model coverage.

In conclusion, we maximally amplify the adversarial per-

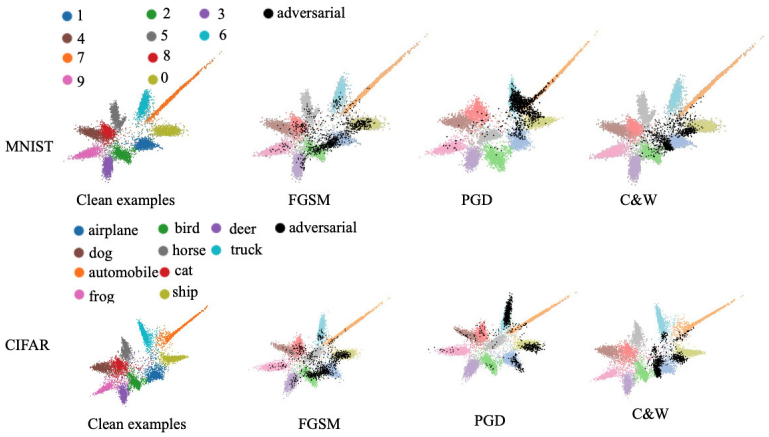

Fig. 6. Visualizations of MNIST and CIFAR-10 test data penultimate layer representations. Each row is a dataset, while each column the distribution under an attack.(a) shows only the projected representations of clean data. (b), (c), and (d) show faded non-adversarial points along with black points corresponding to FGSM, PGD, and CW adversarially perturbed airplanes, respectively.

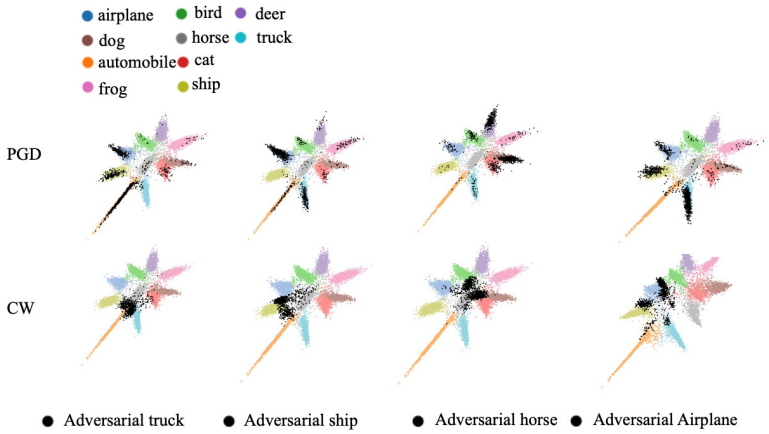

Fig. 7. Visualizations of CIFAR-10 test data penultimate layer representations under PGD and C&W attack. We select truck, ship, horse and airplane as examples.

turbation and visualize feature representation to demonstrate the validity of theorem 1. We also assert that the inherent proprieties of optimized adversarial attack algorithms determined that AEs can generate similar incorrect inputs or neuron activation patterns but not more diverse, which is contradictory to testing requirements in terms of model coverage.

D. RQ2: Relationship between AEs & Coverage Criteria

Existing coverage criteria cannot formally explain the inherent connection with the model, but what we can ensure is that AEs can explicitly improve the aforementioned coverage criteria. Nevertheless, existing works almost attempt to demonstrate the applicability of the proposed coverage criteria by comparing the coverage of the original examples

and adversarial examples. Table. II shows the comparison of average percent increases in coverage criteria over the baseline of randomly clean test suites. We select four off-the-shelf coverage criteria as the baseline of the evaluation: Neuron Coverage (NC), k-Multisection Neuron Coverage (KMNC), Top-k Neuron Patterns (TKNP), and Neuron Path Coverage (NPC). Note that we set the threshold of activation value equal to 0.2 because highly activated DNNs are difficult to increase coverage under the lower threshold, while only small portions of neurons have been activated under too higher threshold. For KMNC and TKNP, we follow the experimental configuration in DeepGauge [17], in which KMNC divides the range $[low_n, high_n]$ into k=1000 equal sections and TKNP sets top-2 neurons on each layer. For NPC, we set the simplest version of experimental configuration as [20] and set α, β and k are 0.7, 0.6, and 1 respectively. In this work, we do not discuss the inherent correlation between model quality and coverage criteria. Specifically, we find that all AEs can efficiently improve the existing coverage criteria (e.g. NC, KMNC, TKNP and NPC), particularly the top-K neuron patterns (TKNP) show a high distinguishing capability to find the defects in DL model as shown in Table. II. From Fig. 9, NC, KMNC, and NPC have a similar increasing pattern in that the coverage increases rapidly at the beginning and then keeps a steady but slow-growing performance. Noting that TKNP basically shows linear growth as increasing with AEs which also reflected in

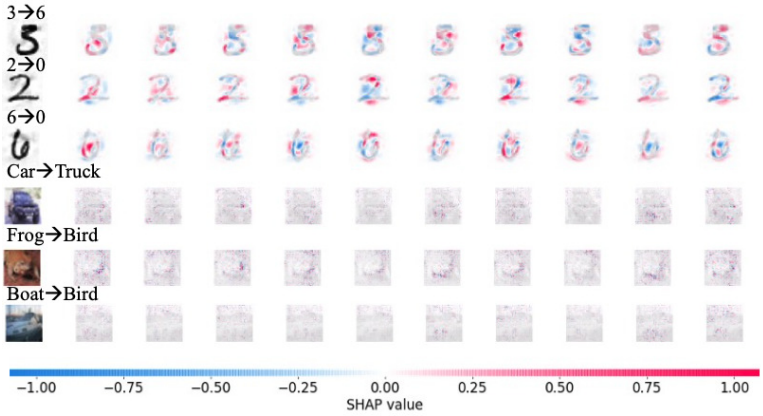

Fig. 8. SHAP values of adversarial features to model output.

Model	FGSM				PDG				C&W			
	NC	KMNC	TKNP	NPC	NC	KMNC	TKNP	NPC	NC	KMNC	TKNP	NPC
LeNet-1	22.59	14.88	38.90	15.67	20.78	6.31	29.50	10.89	18.70	10.85	37.9	11.23
LeNet-4	18.26	15.73	50.69	13.43	19.98	14.10	48.59	7.76	18.09	4.82	48.78	9.89
ResNet56	12.86	14.59	60.78	10.76	13.80	10.10	56.89	8.87	10.90	15.59	47.89	10.23
DenseNet121	10.08	10.46	58.78	10.50	9.08	7.67	57.78	9.87	8.82	10.87	40.89	13.38
Average	15.95	13.92	52.29	12.59	15.91	9.55	48.19	9.35	14.13	10.53	44.32	11.18

TABLE II
THE AVERAGE PERCENT INCREASE OF EXISTING COVERAGE CRITERION (NC, KMNC, TKNP, NPC) ON AES OVER THAT ON ORIGINAL IMAGES

Answer to RQ1: Our theoretically analysis and empirical experiment demonstrate that optimized gradient-based attack methods only cover limited several neuron activate patterns. Hence, the common thing between DL testing and AEs is generating incorrect inputs to test DL model. Differently, AEs tend to be similar incorrect inputs, which cannot trigger different logic of the model, while DL test should ensure diversity and adequacy.

[32]. However, there is little work to analyze why the AEs can improve coverage. Yan [32] conclude that *optimization algorithms have the implicit capabilities of exploring different activation patterns by following the direction of gradients*, but without giving more detailed explanations.

As such, we plan to analyze this experimental result from Theorem 1. Firstly, we can infer that the optimized adversarial attack technologies try to minimize the loss function, but in a normal model, there are existing many local minima in the loss function. Under limited of (1) fixed optimization iterations, (2) fixed step size and (3) random initialization, the optimized adversarial attack technologies potentially find the perturbation ϵ which falls to different points around the local minima. Intuitively, almost every distribution range of projected representations of adversarial examples exceeds that of clean examples from Fig. 6 and Fig. 7. Moreover, we can see that all mentioned coverage criteria share a similar increasing pattern in Fig. 9. Hence, we believe that optimized adversarial attacks can actually activate more neuron patterns along the direction of gradient. Of particular note is that the adversarial attack (FGSM) with perturbation size ($L_\infty - norm$), can efficiently improve the coverage criteria than optimized-based adversarial attacks(e.g. PGD, C&W). We believe that it mainly attributes to L_∞ normalization. The equation (1) and (2) present the difference between L_∞ normalization and L_2 normalization. Clearly, we add or subtract the ϵ-size perturbation to each pixel which is no longer the optimal direction to misclassify the model, while the L_2 version remains the direction of gradient unchanged. Hence, we can understand that FGSM adds randomly-looking perturbation to the original image and randomly activates more neuron patterns (e.g. NC, KMNC, TKNP and NPC).

Answer to RQ2: AEs can efficiently improve the coverage criteria such as NC, KMNC, TKNP and NPC, because the the optimization algorithm can explore the different activation patterns along the direction of gradient for the following reason: (1) fixed optimization iterations, (2) fixed step size, (3) random initialization and (4) normalization norm.

E. RQ3: Exploring suitable test suites generation approaches

Several test suites generation approaches have been proposed such as Fuzz-based, and search-based generation technologies. To answer RQ3, we investigate the experiments on MNIST and CIFAR datasets to generate the test suites by DeepHunter (coverage criteria based Fuzz) [12], PDG (gradient-based), and search based [26] against DL models, hereafter these test suite generation technologies will be abbreviated as D_H, D_P and D_S respectively. DeepHunter performs metamorphic (e.g. semantic-preserving) fuzz to produce the test suites

TABLE III
COMPARISON OF ATTACK IMAGES GENERATED BY DEEPHUNTER D_H, PDG D_P AND SEARCH-BASED TECHNOLOGY D_S IN TERM OF ATTACK SUCCESS RATE AND NPC

Model	Attack Success Rate(%)			NPC (%)		
	D_H	D_S	D_P	D_H	D_S	D_P
LeNet-1	90.2	87.6	100.0	80.4	88.0	40.8
LeNet-4	86.4	90.7	100.0	75.9	87.6	38.4
ResNet56	88.6	80.5	90.9	55.6	60.9	37.3
DenseNet121	80.7	85.5	86.8	75.3	77.5	43.3

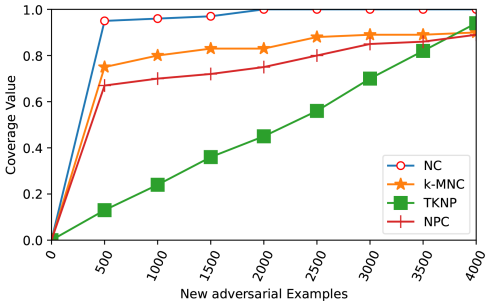

Fig. 9. The coverage increasing by adding AEs.

under the feedback from multiple pluggable coverage criteria. Search-based technologies can combine with various existing meta-heuristics to seek out the approximate optimal solution, while the solution optimality is judged by the fitness function.

We compare them from the following aspects: We first compare the attack success rate of these methods by calculating the percentage of D_H, D_P and D_S that can lead to misprediction. Secondly, we compare the quality of test suites by calculating their capability of exploring more diverse logics. Noting that NPC was proposed to measure the adequacy of test suites in exploring the decision logic. Hence, We use NPC as an indicator to assess the quality of test suites. From the experimental result in Table. III, we found that the AEs generated by PGD basically achieve almost 100% attack success rate on all models and datasets, while D_G and D_S achieve relatively lower attack success rate compared with D_P. On the other hand, we found that D_S achieved the optimal results on the NPC, which indicated that search-based test suite generation technology D_S can explore more decision logics from the perspective of NPC. These findings preliminarily demonstrate that optimized adversarial attacks have remarkable advantages in revealing the defects of models, but search-based test suites generation technology is a promising direction to reduce the test space and explore more diverse logic of models.

Answer to RQ3: Search-based test suites generation technologies achieve better performance compared with AEs and Fuzz technologies in terms of test sufficiency. We believed that this research direction can closely connect with existing coverage criteria and other objectives to guide test suites generation.

F. Discussion

These findings naturally prompted two further research directions. (1) Design a novel optimized objective for existing adversarial attack algorithms to generate more diverse test suites, rather than aiming at maximizing loss while keeping l_p-norm distance from the original inputs. Regularization technique that can be seamlessly integrated into existing adversarial attack methods to promote neural activation diversity. (2) Explore various test suites generation technologies in the community of software engineering, especially search-based technology is the key direction of future research to test DL model. Although AEs are closely corrective with existing coverage criteria in given theoretical and empirical analysis, there is no formal analysis that these criteria are correlated with model robustness. In future work, we also look forward to proposing a more plausible coverage criterion to establish a close connection with model quality.

VI. CONCLUSION

To the best of our knowledge, our work is the first to formally claim the contradiction between the inherent properties of existing AE algorithms and the requirements of DL test. In this paper, we provide theoretical and empirical evidence that the optimized adversarial examples (AEs) tend to activate several limited neuron patterns compared with normal test suites under the formula derivation and visualization technologies. Using this, we further give a formal analysis that AEs can dramatically enhance existing coverage criteria. Finally, given preliminary comparison results in different test suites generation technologies show that search-based test suites technologies are a promising direction to test DL model.

REFERENCES

[1] X. Ma, H. Xu, H. Gao, M. Bian, and W. Hussain, "Real-time virtual machine scheduling in industry iot network: A reinforcement learning method," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, 2022, pp. 2129–2139.

[2] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[3] K. Gupta and T. Ajanthan, "Improved gradient-based adversarial attacks for quantized networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6810–6818.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] F. Tramer, "Detecting adversarial examples is (nearly) as hard as classifying them," in *International Conference on Machine Learning*. PMLR, 2022, pp. 692–712.

[6] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019, pp. 12-23.

[7] A. Shamir, O. Melamed, and O. BenShmuel, "The dimpled manifold model of adversarial examples in machine learning," *arXiv preprint arXiv:2106.10151*, 2021.

[8] D. Steinberg and P. Munro, "Visualizing representations of adversarially perturbed inputs," *arXiv preprint arXiv:2105.14116*, 2021.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[10] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[11] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems," in *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2018, pp. 132–142.

[12] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See, "Deephunter: a coverage-guided fuzz testing framework for deep neural networks," in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 146–157.

[13] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun, "Dlfuzz: Differential fuzzing testing of deep learning systems," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 739–743.

[14] J. Kim, R. Feldt, and S. Yoo, "Guiding deep learning system testing using surprise adequacy," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019, pp. 1039–1049.

[15] Y. Idelbayev, "Proper ResNet implementation for cifar10/cifar100 in pytorch," https://github.com/akamaster/pytorch_resnet_cifar10, 2021.

[16] K. Jammalamadaka and N. Parveen, "Testing coverage criteria for optimized deep belief network with search and rescue," *Journal of big data*, vol. 8, no. 1, 2021, pp. 1–20.

[17] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu *et al.*, "Deepgauge: Multi-granularity testing criteria for deep learning systems," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018, pp. 120–131.

[18] F. Harel-Canada, L. Wang, M. A. Gulzar, Q. Gu, and M. Kim, "Is neuron coverage a meaningful measure for testing deep neural networks?" in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 851–862.

[19] S. Gerasimou, H. F. Eniser, A. Sen, and A. Cakan, "Importance-driven deep learning system testing," in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE, 2020, pp. 702–713.

[20] X. Xie, T. Li, J. Wang, L. Ma, Q. Guo, F. Juefei-Xu, and Y. Liu, "NPC: Neuron path coverage via characterizing decision logic of deep neural networks," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 3, 2022, pp. 1–27.

[21] H. Gao, B. Qiu, R. J. D. Barroso, W. Hussain, Y. Xu, and X. Wang, "TSMAE: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder," *IEEE Transactions on Network Science and Engineering*, 2022.

[22] H. Gao, W. Huang, T. Liu, Y. Yin, and Y. Li, "PPO2: location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems," *IEEE transactions on intelligent transportation systems*, 2022, pp. 56–73.

[23] S. Shan, E. Wenger, B. Wang, B. Li, H. Zheng, and B. Y. Zhao, "Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 67–83.

[24] H. Gao, B. Dai, H. Miao, X. Yang, R. J. D. Barroso, and H. Walayat, "A novel GAPG approach to automatic property generation for formal verification: The gan perspective," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 1, 2023, pp. 1–22.

[25] H. Dai, C.-a. Sun, and H. Liu, "Deepcontroller: Feedback-directed fuzzing for deep learning systems," *strategies*, vol. 5, p. 6, 2020, pp. 22–33.

[26] H. B. Braiek and F. Khomh, "Deepevolution: A search-based testing approach for deep neural networks," in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2019, pp. 454–458.

[27] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand, and T. Stifter, "Testing autonomous cars for feature interaction failures using many-objective search," in *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2018, pp. 143–154.

[28] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, 2012, pp. 141–142.

[29] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger *et al.*, "Comparison of learning algorithms for handwritten digit recognition," in *International conference on artificial neural networks*, vol. 60, no. 1. Perth, Australia, 1995, pp. 53–60.

[30] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, no. 7, 2010, pp. 1–9.

[31] N. Dif and Z. Elberrichi, "A new deep learning model selection method for colorectal cancer classification," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 11, no. 3, 2020, pp. 72–88.

[32] S. Yan, G. Tao, X. Liu, J. Zhai, S. Ma, L. Xu, and X. Zhang, "Correlations between deep neural network model coverage criteria and model quality," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 775–787.

[33] H. Gao, J. Xiao, Y. Yin, T. Liu, and J. Shi, "A mutually supervised graph attention network for few-shot segmentation: The perspective of fully utilizing limited samples," *IEEE Transactions on Neural Networks and Learning Systems*, 2022, pp. 2–10.