An Empirical Study on Correlation between Coverage and Robustness for Deep Neural Networks

Yizhen Dong*, Peixin Zhang†, Jingyi Wang‡, Shuang Liu*, Jun Sun§, Jianye Hao*‖,
Xinyu Wang†, Li Wang*, Jinsong Dong‡, and Ting Dai¶
*College of Intelligence and Computing, Tianjin University
†College of Computer Science and Technology, Zhejiang University
‡National University of Singapore
§Singapore Management University
¶Huawei International Pte.Ltd.
‖Noah's Ark Lab, Huawei

***Abstract*—Deep neural networks (DNN) are increasingly applied in safety-critical systems, e.g., for face recognition, autonomous car control and malware detection. It is also shown that DNNs are subject to attacks such as adversarial perturbation and thus must be properly tested. Many coverage criteria for DNN since have been proposed, inspired by the success of code coverage criteria for software programs. The expectation is that if a DNN is well tested (and retrained) according to such coverage criteria, it is more likely to be robust. In this work, we conduct an empirical study to evaluate the relationship between coverage, robustness and attack/defense metrics for DNN. Our study is the largest to date and systematically done based on 100 DNN models and 25 metrics. One of our findings is that there is limited correlation between coverage and robustness, i.e., improving coverage does not help improve the robustness. Our dataset and implementation have been made available to serve as a benchmark for future studies on testing DNN.**

I. Introduction

Recent years have seen rapid development on deep learning techniques as well as applications in a variety of domains like computer vision [1], [2] and natural language processing [3]. There is a growing trend to apply deep learning for solving safety-critical tasks, such as face recognition [4], self-driving cars [5] and malware detection [6]. Unfortunately, deep neural networks (DNN) are shown to be vulnerable to attacks and lack of robustness. For instance, they are easily subject to adversarial perturbation [7], [8], i.e., a DNN makes a wrong decision given a carefully crafted small perturbation on the original input. This suggests that DNN, just like software systems, must be properly analyzed and tested before they are applied in safety-critical systems.

The software engineering community welcomed the challenge and opportunity. Multiple software testing approaches, i.e., differential testing [9], mutation testing [10], [11] and concolic testing [12], have been adapted into the context of testing DNN. Inspired by the noticeable success of code coverage criteria in testing traditional software systems, multiple coverage criteria[1], e.g., neuron coverage [9] and its extensions DeepGauge [13], MC/DC [14], and Surprise Adequacy [15], have been proposed. Coverage criteria quantitatively measures how well a DNN is tested and offers guidelines on how to create new test cases. The underlying assumption is that a DNN which is better tested, i.e., with higher coverage, is more likely to be robust.

This assumption however is often not examined or only evaluated with limited DNN models and structures, making it unclear whether the results generalize. Furthermore, how a test suite improves the quality of a DNN is different from that of a software system. A software system is improved by fixing bugs revealed by a test suite. A DNN is typically improved by retraining with the test suite. While existing studies show that retraining often improves a DNN's accuracy to some extent [9], [12], it is not clear whether there is correlation between the coverage of the test suite and the improvement, i.e., does a set of inputs with higher coverage imply better improvement (on DNN robustness)?

Inspired by the work in [16], we conduct an empirical study to evaluate whether coverage is correlated with robustness of DNN and additional metrics which are associated with the quality of DNN [17]. In particular, we would like to answer the following research questions.

- Are there correlations between testing coverage criteria and the robustness of DNN?
- Are there correlations among different coverage criteria themselves?
- Are there correlations between the improvement of coverage criteria and the improvement in terms of robustness after the DNN is retrained?
- Are there metrics that are strongly correlated to the robustness of DNN or the robustness improvement after retraining?

Based on the answers to the above questions, we aim to provide practical guidelines for developing testing methods which contribute towards improving the robustness of DNN.

Conducting such an empirical study is highly non-trivial. First, we need a large set of real-world DNN for the study. However, training realistic DNN often takes significant amount

Yizhen Dong and Peixin Zhang are co-first author

Jingyi Wang and Xinyu Wang are corresponding authors

[1] Metric and criterion are used interchangeably.

of time and resource. For instance, it takes 15 GPU hours to train a ResNet-101 model. Our study trained 100 state-of-the-art DNN models[2] with a variety of architectures with two popular datasets, i.e., MNIST [18] and CIFAR10 [19]. Obtaining these models took a total of 150 GPU hours.

Second, we need to obtain adversarial samples by attacking the trained original models. We adopt 3 state-of-the-art attack methods, i.e., FGSM [8], JSMA [20] and C&W [7], to attack the original models, in order to obtain different adversarial sample sets. Some of the adversarial attack methods, e.g., JSMA and C&W, are known to be time-consuming. It takes us a total of $1,810$ GPU hours to obtain adversarial samples for all the original models with the 3 attack methods.

Last but not least, we need a systematic and automatic way of evaluating the coverage, robustness, and other associated metrics, which is not always straightforward. For instance, there are multiple definitions of robustness in the literature [21], [22], some of which are complicated and expensive to compute (e.g., it took 12 GPU hours to compute a CLEVER score [22] for GoogLeNet-22.). In this work, we develop a self-contained toolkit called *DRTest* (*D*eep *R*obustness *T*esting), which calculates a comprehensive set of metrics on DNN, including 1) 8 testing coverage criteria proposed for DNN, 2) 2 robustness metrics for DNN, and 3) a set of 15 attack and defense metrics for DNN. A total of $4,150$ GPU hours are spent on computing these metrics.

Our empirical study is conducted as follows. For each dataset, we first train 25 diverse seed models (with state-of-the-art architectures), attack each seed model with different attacking methods to generate adversarial samples (with varying attack parameters), augment the training dataset with the generated adversarial samples, and retrain the model. We apply *DRTest* to calculate a range of metrics for every model. Afterwards, we apply a standard correlation analysis algorithm, the Kendall's rank correlation coefficient [23], to analyze the correlations between the metrics.

In summary, we make the following contributions.

- We conducted an empirical study to systematically investigate the correlation between coverage, robustness and related metrics for DNN. Based on the study results, we discuss potential research directions on DNN testing.
- We implemented a self-contained and extensible toolkit which calculates a large set of metrics, which can be used to quantitatively measure different aspects of DNN.
- We publish online our models, adversarial samples as well as *DRTest*, which can be used as a benchmark for future proposals on methods for DNN testing.

We organize the remainder of the paper as follows. Section II introduces the background knowledge of this work. Section III presents our research methodology. Section IV shows details on our implementations. Section V reports our findings on the research questions. We present related works in Section VI and conclude in Section VII.

II. Preliminaries

In this section, we briefly review preliminaries related to this work, which include Deep Neural Networks (DNN), adversarial attacks on DNN, testing methods for DNN, and robustness of DNN.

A. Deep Neural Networks

In this work, we focus on DNN classifiers $\mathcal{D}(X) : X \rightarrow Y$, where X is a set of inputs and Y is a finite set of labels. Given an input $x \in X$, a DNN classifier transforms information layer by layer and outputs a label $y \in Y$ for the input x. We try to cover a wide range of DNN architectures in our work including LeNet [24], VGG [2], GoogLeNet [25] and ResNet [1].

B. Adversarial Attack

Since Szegedy *et al.* discovered that DNNs are intrinsically vulnerable to adversarial samples (i.e., sample inputs which are generated with the intention to trick a DNN into wrong decisions) [26], many attacking approaches have been developed to craft adversarial samples. We adopt 3 popular attacking algorithms i.e., FGSM [8], JSMA [20] and C&W [7] in our work to generate adversarial examples for further study.

C. Testing Deep Neural Networks

A variety of traditional software testing methods like differential testing [27], [28], concolic testing [29] have been adapted to the context of testing DNN [9], [12] to find adversarial samples (in hope of revealing bugs in DNN). In the following, we review some recently proposed coverage criteria for DNN. Neuron coverage [9] is the first coverage criteria proposed for testing DNN, which quantifies the percentage of activated neurons by at least one test case in the test suite. Later, Ma *et al.* proposed DeepGauge [13], which extends neuron coverage with coverage criteria which are defined based on the activation values from both *neuron-level* and *layer-level*. Based on the idea that a good test suite should be 'surprising' compared to the training set, Kim *et al.* [15] defined two measures on how surprising a testing input is to the training set, readers are referred to [15] for details.

D. Robustness of Deep Neural Networks

Given the existence of adversarial samples, adversarial robustness becomes an important desired property of a DNN which measures its resilience against adversarial perturbations. Following the definitions proposed by Katz et al. [30], adversarial robustness can be categorized into local adversarial robustness and global adversarial robustness.

Definition II.1. *(Local Adversarial Robustness) Given a sample input x, a DNN $\mathcal{D}$ and a perturbation threshold ϵ, $\mathcal{D}$ is $\epsilon-$local robust iff for any sample input x' such that $||x - x'||_p \leq \delta$, we have $\mathcal{D}(x) = \mathcal{D}(x')$, where $|| \cdot ||_p$ is the p-norm to measure the distance between two sample inputs.*

Definition II.2. *(Global Adversarial Robustness) For any sample inputs x and x', a DNN $\mathcal{D}$ and two thresholds δ, ϵ,*

[2] 25 seed models trained with original dataset and 75 models retrained using original dataset augmented with adversarial samples.

$\mathcal{D}$ is $(\delta, \epsilon)-$robust iff for any $||x - x'||_p \leq \delta$, we have $|\mathcal{D}(x) - \mathcal{D}(x')| \leq \epsilon$.

Local robustness measures the robustness on a specific input, global robustness measures the robustness on all inputs.

Verifying whether a DNN satisfies local or global robustness is an active research area [30], [31], [32] and existing methods do not scale to state-of-the-art DNNs (especially for global robustness). Thus, multiple metrics have been proposed in order to empirically evaluate the adversarial robustness of a DNN [22], [21], [33], [34]. We introduce two widely used adversarial robustness metrics adopted in this work as follow.

Global Lipschitz Constant Lipschitz Constant [21] measures the sensitivity of a model to adversarial samples and the Lipschitz constant is only related to the parameters of f. In our context, the function is in the form of a DNN. Its Lipschitz constant can be calculated recursively layer-by-layer from the output layer all the way to the input layer, taking consideration of short-cuts in ResNet and inception module in GoogLeNet. For the calculation details of Lipschitz Constant for the fully connected layer or the convolution and aggregation layers, please refer to [21] and [35] respectively.

CLEVER Score Another robustness metric we adopt is the CLEVER score (Cross-Lipschitz Extreme Value for nEtwork Robustness) [22], which is a recently proposed attack-independent robustness score for large scale networks. Readers are referred to [22] for details.

III. Methodology

The overall workflow of our experiment is shown in Figure 1. We follow a common DNN testing process (e.g., by [9], [13]), as shown at the top of the figure, whilst extracting a variety of metrics (as shown in the middle of the figure) which are used for correlation analysis (as shown at the bottom). We start with training a model from a training set using state-of-the-art training methods. Afterwards, various adversarial attacks [8], [20], [7] are applied to generate new test cases. The last step is to augment the training set with the new test cases and obtain a retrained model.

We collect four different groups of metrics to characterize different components of the process, i.e., (1) a set of testing coverage metrics of models, (2) a set of attack metrics of different kinds of adversarial attacks on the original models, (3) a set of robustness metrics of both the original models and the retrained models, and (4) a set of defense metrics which measure the differences between the retrained model and the original model. We repeat the above process for the 25 seed models, obtain in total 100 models, calculate the corresponding metrics and then conduct correlation analysis on all these metrics. In the following, we illustrate the challenges and our design choices of each part in detail.

Adversarial Attacks We adopt three state-of-the-art DNN attack methods i.e., FGSM [8], CW [7] and JSMA [20] to generate adversarial samples, these generated adversarial samples are combined with the original datasets as new (training and testing) datasets for model retraining

TABLE I: Summary of metrics

Metric Type	Metric Name	Description
Testing	NC	Neuron Coverage [9]
	KNC	K-multisection Neuron Coverage [13]
	SNAC	Strong Neuron Activation Coverage [13]
	NBC	Neuron Boundary Coverage [13]
	TKNC	Top-k Dominant Neuron Coverage [13]
	TKNP	Top-k Dominant Neuron Patterns Coverage [13]
	LSA/DSA	Surprise adequacy to training set [15]
Robustness	Lipschitz constant	The global Lipschitz constant [21]
	CL1/CL2/CLi	Clever score with $L_1/L_2/L_\infty$ norm [22]
Attack	MR	Misclassification Ratio [17]
	ACAC	Average Confidence of Adversarial Class [17]
	ACTC	Average Confidence of True Class [17]
	ALD_p	Average L_p Distortion [17]
	ASS	Average Structural Similarity [36]
	PSD	Perturbation Sensitivity Distance [37]
	NTE	Noise Tolerance Estimation [37]
	RGB	Robustness to Gaussian Blur [17]
	RIC	Robustness to Image Compressionr [17]
	CC	Computation Cost [17]
Defense	CAV	Classification Accuracy Variance [17]
	CRR/CSR	Classification Rectify/Sacrifice Ratio [17]
	CCV	Classification Confidence Variance [17]
	COS	Classification Output Stability [17]

Model Retraining For each original model, we obtain three sets of adversarial samples using attack methods. We combine original training set with one set of the adversarial samples for retraining one model. As a result, we obtain 3 retrained models for each original model, one for each attacking method. We follow the standard partition of $6:1$ for training and testing on the MNIST dataset and $5:1$ for the CIFAR10 dataset.

Metric Calculation As our objective is to investigate the correlations between coverage, robustness and other metrics associated with DNN, we conduct a thorough survey on existing metrics and collected 25 metrics in total which are summarized in Table I. Note that the attack metrics measure to what extent the attacks are successful, imperceptible, whereas the defense metrics measure mainly on how the retrained models preserve the accuracy of the original model. For brevity, we refer the readers to the original papers for details. We calculate values of all metrics based on their original definitions and use default parameters according to their original papers.

Correlation Analysis We conduct correlation analysis, a statistical technique that shows whether and how strongly pairs of variables are correlated, on the metrics. We are particularly interested to observe which metrics are correlated to the robustness of a DNN model. In this work, we adopt a commonly used correlation coefficients, Kendall's τ rank correlation coefficient [23], which is a rank based correlation that measures monotonic relationship between two variables, to measure the correlations between different metrics. Note that compared to alternative methods like Pearson product-moment correlation coefficient [38], Kendall's τ rank correlation coefficient does not require that the dataset follows a normal distribution or the correlation is linear. We calculate the correlations of different metrics for the two dataset separately, in order to avoid the potential impact due to the training data.

IV. Implementation and Configurations

Our system is implemented based on the TensorFlow framework [39] and the architecture is shown in Figure 2. There are 4 layers, i.e., the data layer, the algorithm layer, the

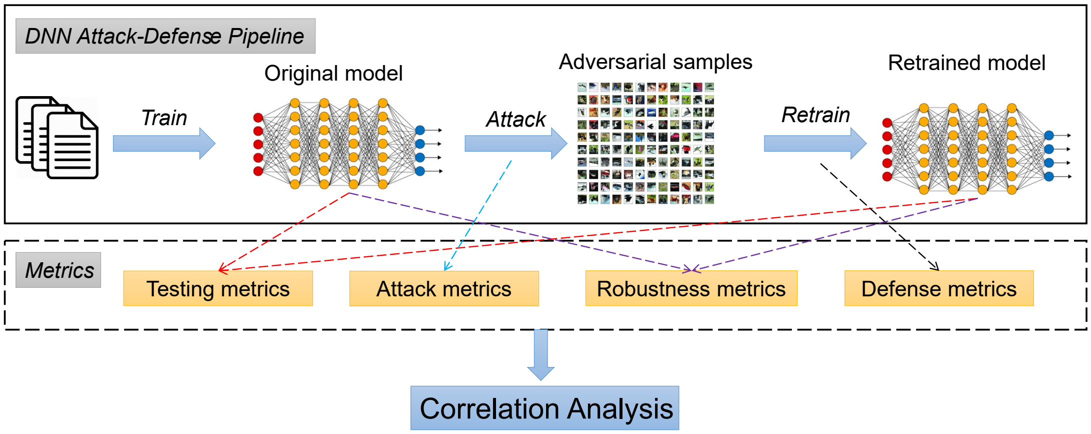


Fig. 1: Overview of experiment design

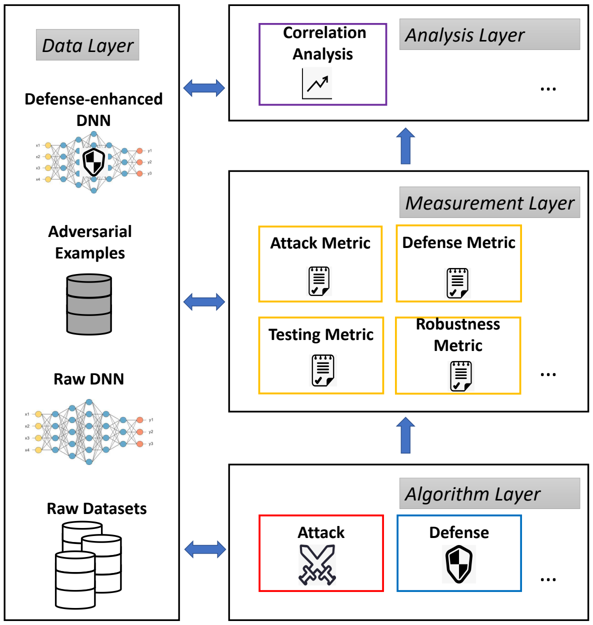


Fig. 2: The architecture of our framework

measurement layer and the analysis layer. Our implementation is designed to be extensible, i.e., each layer can be extended with new models and algorithms with little impact on the other layers. Our implementation, including all the data and algorithms, is open source on GitHub[3].

The data layer maintains all data used in our study which interacts with all other layers. We cover a wide range of different deep learning model structures, including 3 LeNet family models (LeNet-1,4,5 [24]), 4 VGG family models (VGG-11,13,16,19 [2]), 4 ResNet family models (ResNet-18,34,50,101 [1]) and 3 GoogLeNet family models (GoogLeNet-12,16,22 [25]). We adopt two popular publicly-available datasets, i.e., MNIST [18] and CIFAR10 [19] to train DNN models in our work, due to the limitation of space, we will not introduce them here.

The algorithm layer contains a set of algorithms for attacking DNN as well as algorithms for defending DNN through retraining. For each trained model, we use three attack methods (e.g., FGSM, CW and JSMA) to generate adversarial samples. The principle of choosing parameters for each attack is to balance the imperceptibility and success rate of generating adversarial samples. For MNIST, we adopt the same parameters from cleverhans [40] for all three attacks. For CIFAR10, we slightly changed the parameters of FGSM and CW in order to obtain better imperceptibility.

To further avoid bias introduced by hyper-parameters, we run each attack method on the original dataset for 3 times with different hyper-parameter configurations. Then we combine the successful adversarial samples generated from 3 runs of attacks as the adversarial sample set for model retraining. Reader can find the details of the hyper-parameter configurations for each attack method on our GitHub repository.

During training and retraining, we adopt a learning rate of 0.001, a batch size of 128 for all models in the two datasets. For MNIST, a test accuracy above 98% is accepted in both training and retraining. For CIFAR10, a test accuracy above 80% is accepted during training process and a test accuracy above 85% is required for retraining which is widely accepted.

The measurement layer contains all implementation as shown in Table I. We calculate four robustness values, i.e., Global Lipschitz Constant (Lipz) and the CLEVER score (CL1, CL and CLi) for each model. Note that LeNet is not feasible for CIFAR10. In our experiment, since calculate CLEVER score is extremely time-consuming for GoogLeNet, we reduce the number of images to 50 and sampling parameter $N_b = 50$, as it is reported that 50 or 100 samples are usually sufficient to obtain a reasonably accurate robustness estimation [22]. We calculate the coverage criteria of different DNN models with the same test suite (i.e., the original test suite of MINIST or CIFAR10) and obtain $14*4$ and $11*4$ values of each coverage criteria on MNIST and CIFAR10, respectively.

Defense Metrics are calculated for all the defense enhanced models, i.e., models after adversarial training, according to their original definitions [17]. For each dataset, We obtain

[3] https://github.com/ICECCS2020/DRtest

TABLE II: Time for different steps in the experiment

dataset	model family	generate AE	train & retrain	metric calc
MNIST	LeNet	<0.5	<0.5	<0.5
	VGG	160	6	420
	ResNet	240	12	1200
	GoogLeNet	120	25	300
CIFAR10	VGG	540	12	550
	ResNet	450	45	1350
	GoogLeNet	300	50	330

$14{*}3$ and $11 * 3$ values for each defense metric on MINIST and CIFAR10, respectively. Attack Metrics are calculated for the generated adversarial examples of each attack method, all parameters of attack metrics are set based on their original definitions [17]. We obtain $14{*}3$ and $11 * 3$ values for each attack metric on MINIST and CIFAR10, respectively.

We additionally calculate a set of Δ-metrics, which are denoted as Metric-diff. For instance, Lipz-diff is the Lipschitz Constant of the retrained model minus that of the original model. We obtain $14{*}3$ and $11 * 3$ Δ-robustness for each robustness metric on MINIST and CIFAR10. Similarly, we calculate Δ-coverage metrics by subtracting the coverage achieved by the augmented test set (i.e., the original test set plus the adversarial samples) from that of the original test set. We obtain $14{*}3$ and $11 * 3$ Δ-coverage values for each coverage metric on MINIST and CIFAR10.

The analysis layer implements the correlation analysis algorithm [23]. We first plot the data to observe the trend and then decide on the correlation analysis method to use. By observing the data plot, we found that the data does not show a linear trend. Therefore, we choose the Kendall's τ rank correlation coefficient [23], which does not assume that the data follows a normal distribution or the variables have a linear correlation.

All experiments were conducted using four GPU servers, readers can find configurations of our servers on our GitHub repository. In total, the experiment took more than $6,100$ GPU hours to finish. Table II shows the time spent on different steps of our experiments, the unit is 1 GPU hour.

V. Findings

A. Research Questions

RQ1: Are there any correlations between existing test coverage criteria and the robustness of the DNN models?

To answer the question, we conduct correlation analysis on the coverage metrics and the robustness metrics of all models on the original test set. The results are shown in Figure 3. The number and the color represent the strength of the correlation. The correlation value is a number between -1 and 1. Positive number (and blue color) indicates positively correlated and negative number (and red color) indicates negative correlated. The larger the absolute number is, the stronger the correlation is. The darker the color is, the stronger the correlation is. We measure the p-value of the sample data set we have and regard p-value greater than 0.05 as insignificant. An "X" mark means that we cannot make a decision because p-value is larger than 0.05 (i.e., insignificant) and a question mark "?" means that there are no valid results since the standard variation of the data is 0. According to the definition of correlation in Guildford scale [41], an absolute value of less than 0.4 means that the (positive or negative) correlation is low; an absolute value of 0.4 - 0.7 means that the correlation is moderate; and otherwise the correlation is high or very high (i.e., 0.7-0.9 or above 0.9, respectively).

We have the following observations based on Fig. 3. First, there is no significant or negative correlation between coverage and robustness metrics. In particularly, neural coverage is negatively correlated (i.e., with a value between -0.16 and -0.29) with the CLEVER score and is not significantly correlated with Lipschitz constant for both MNIST and CIFAR10. Moreover, KNC, TKNC and LSA also show negative correlations with CLEVER score on CIFAR10. It suggests that a DNN is less robust if the test set has a larger neuron coverage (although the strength of the correlation is weak), which is unexpected. Second, there is no significant correlation between any of the other coverage and any of the robustness metrics on the MNIST dataset. For the CIFAR10 dataset, positive correlation is only observed between SNAC and the CLEVER score with low strength. This result suggests that a DNN model which achieves high coverage is not necessary robust and vice versa.

We further investigate the correlation among all test coverage criteria themselves. It can be observed from Fig. 3 that NC, KNC, TKNC, LSA and DSA are positively correlated with each other. NBC and SNAC are correlated with each other with medium or high strength, whereas they have no (or weak negative) correlation with the other metrics. The results are consistent with observations reported in [13] and [15] which propose these coverage. This suggests that despite that different coverage criteria are defined differently, they are in general correlated (except for the boundary coverage).

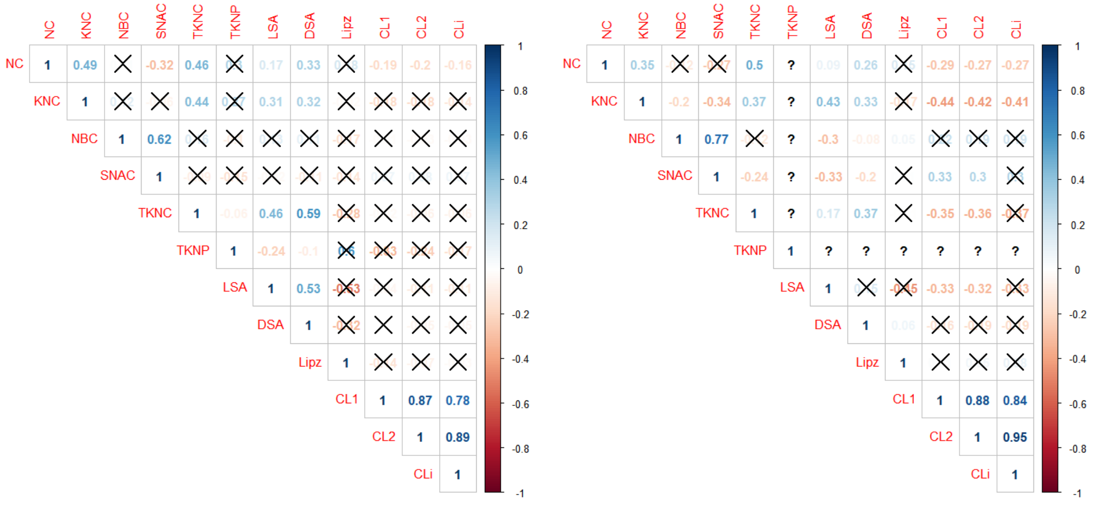
We have the following answer to RQ1.

Different coverage criteria are correlated with each other. There is limited correlation between the coverage criteria and the robustness metrics.

RQ2: Does retraining with new test cases which improves coverage criteria improve the robustness of a DNN model?

To answer this question, we conduct correlation analysis on the difference on coverage criteria and the difference on robustness metrics before and after retraining. The results are shown in Fig. 4. We observe that there is no correlation between the difference on any coverage criteria and the difference on any robustness metrics, except that there is *negative* correlation between TKNC-diff and the CLEVER scores for all the CIFAR10 models. *This result casts a shadow over existing testing approaches, as the existing testing approaches are designed to generate test cases for high coverage, with the hope that such test cases can be used to improve the adversarial robustness of the DNN models.*

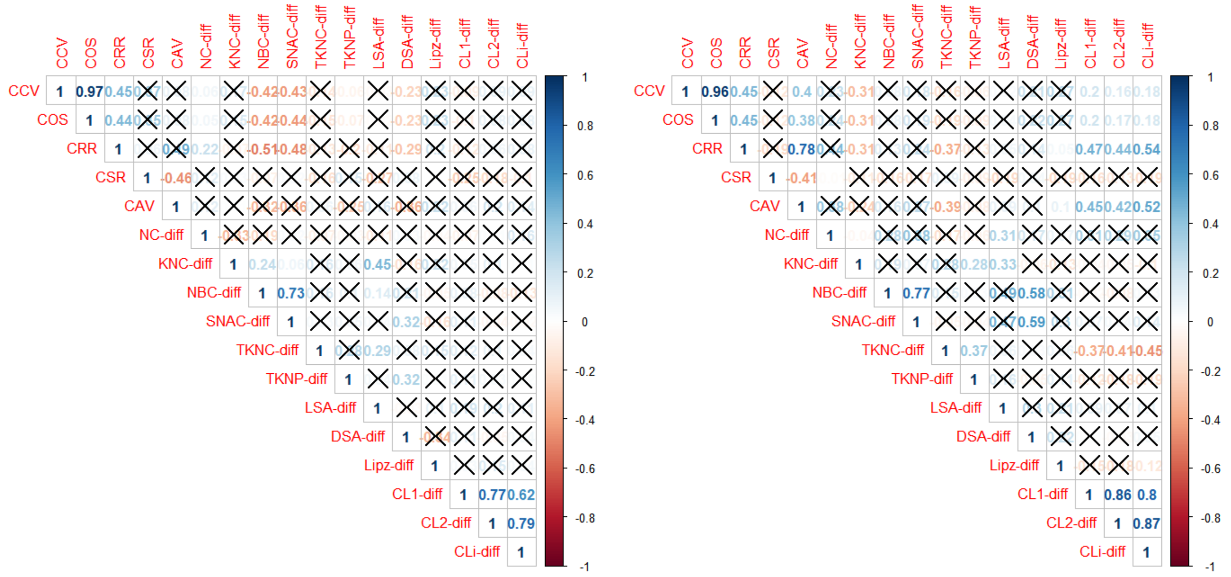
We thus have the following answer to RQ2.



(a) MNIST

(b) CIFAR10

Fig. 3: Test coverage vs. robustness metrics



(a) MNIST

(b) CIFAR10

Fig. 4: Defense Metrics vs. Coverage Criteria Differences vs. Robustness Differences

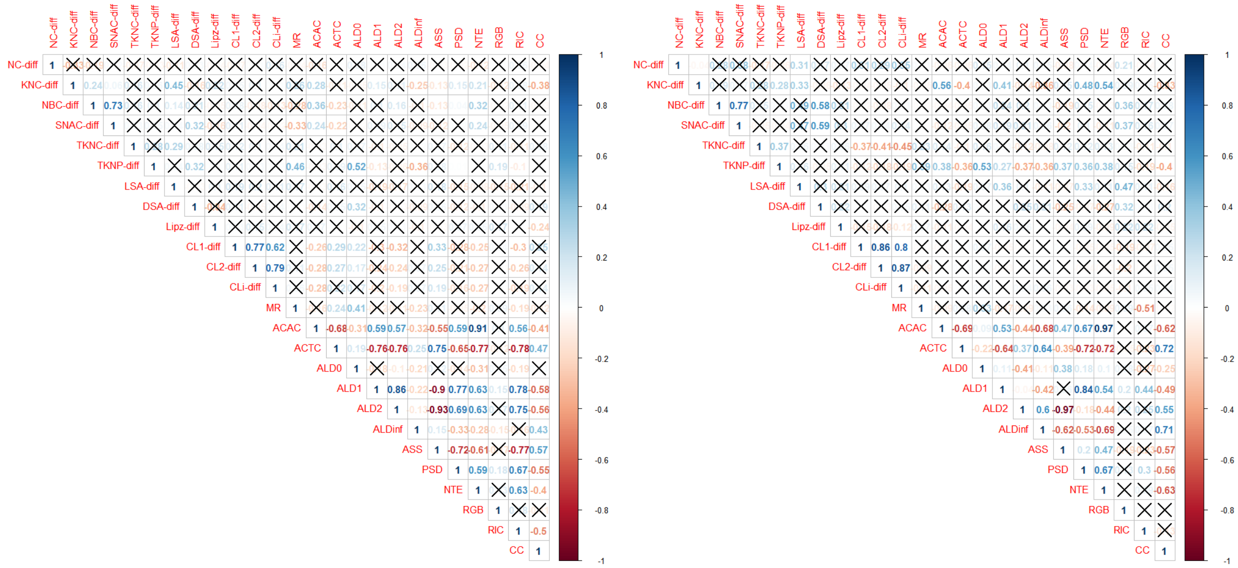
Retraining with new test cases which improve the coverage criteria does not necessarily improve the model robustness.

RQ3: Are there metrics that are strongly correlated to the improvement of model robustness?

The above results show that existing test coverage criteria have limited correlations with the robustness of DNN models and testing methods based on improving the coverage do not improve the robustness of DNN models. The question is then: are there metrics which are correlated to the improvement of the model robustness? To answer the question, we systematically conduct correlation analysis between all metrics (or the metrics's difference before and after retraining) and the improvement of the model robustness.

The correlations between the defense metrics and the improvement of robustness are shown in Fig. 4. There are positive correlations between the difference of the CLEVER scores and all the defense metrics on CIFAR10. In particular, the correlation is of medium level for CRR and CAV. CRR and CAV measure how much the defense-enhanced model preserves the functionality of the original model [17]. Intuitively, this indicates that *a defense method leads to more robustness improvement if the original model is better preserved by the defense-enhanced model.* Furthermore, given the huge cost on computing robustness metrics, such positive correlations potentially provide a lightweight way of estimating on the effectiveness of a model enhancement method.

We additionally analyze the correlation between the attack metrics and the improvement of coverage criteria. We have



(a) MNIST

(b) CIFAR10

Fig. 5: Coverage Criterion Difference vs. Robustness Difference vs. Attack Metric

the following observations from the results shown in Fig. 5. There are correlations between the differences of TKNP and KNC and the attack metrics. Furthermore, NTE is positively correlated with KNC-diff, NBC-diff and SNAC-diff. RGB is positively correlated with NC-diff, NBC-diff and SNAC-diff. Intuitively, NTE and RGB measure the robustness of adversarial samples, which implies that more robust adversarial samples contribute more to the improvement of coverage metrics. Lastly, there is no correlation between the robustness differences and the attack metrics for the CIFAR10 dataset. For the MINIST dataset, we observe negative correlations between the CLEVER score differences with ACAC, ALD2, RIC and NTE, and positive correlations with ASS and ACTC. These observations indicate that *more confident, perceptible and robust adversarial samples contribute more to improving the coverage criteria.*

We have the answer to RQ3.

Some defense metrics are positively correlated to the improvement of model robustness.

RQ4: Are the correlation results consistent across different datasets, model families and correlation analysis methods? This question examines whether the correlation results are universal or rather may vary cross different datasets, model families or correlation analysis methods. To answer this question, we systematically conduct the different correlation analysis using data obtained from different datasets and model families. For the sake of space, we omit the details and refer the readers to the online repository for details.

While the correlation between testing coverage and robustness on MNIST and CIFAR10 are mostly consistent, we do observe some difference of the results across the two datasets. For instance, the attack metrics (except ALDinf) show correlation with CL-diff on MNIST but not on CIFAR10. The defense metrics show strong correlation with robustness and robustness-diff on CIFRA10 but not the case on MNIST.

There are also inconsistent correlation results across different model families. The correlation results on the MNIST, LeNet and VGG families are consistent, which is expected since they have similar model structures. However, it is surprise that models in the GoogLeNet family often show opposite correlation results to those of the MNIST, LeNet and VGG families, especially for correlation between the attack metrics and the improvement of the model robustness. This can be explained as GoogLeNet has a rather different architecture from MNIST, LeNet and VGG (GoogLeNet tends to have more neurons in a layer instead of having more layers).

The above-mentioned inconsistency suggests that the correlation may depend on the dataset and, more noticeably, the model architecture, which further complicates the picture.

Lastly, we apply different correlation analysis algorithms (including Pearson product moment correlation [38] and Spearman's rank-order correlation [42]) to observe whether the results are consistent. Overall, although the results are not identical, the differences are not significant and the results (e.g., whether it is positively or negatively correlated or whether it is strongly or weakly correlated) remain consistent. We choose to present the results of Kendall correlation coefficients in this work as it requires the least assumption on the underlying data. The results of other correlation analysis algorithms are presented in our online repository.

We have our answer to RQ4.

The correlation results are consistent across different correlation analysis algorithms but may vary across different datasets or model families.

B. Explanation

In the following, we aim to interpret and 'explain' the results. These explanations must, however, be taken a grain of salt as they should be properly examined in the future.

First, the reason that existing coverage criteria are not correlated with robustness may simply be due to the fact these coverage criteria are too weak to differentiate robust and not-robust DNN models. It has been shown that high neuron coverage could be easily achieved with a small number of samples [14], and similar conclusions are given by Odena et al. [43] for coverages proposed in DeepGauge, such as neuron boundary coverage. This finding is confirmed by another recent research work [44], which reports that adversarial examples are pervasively distributed in the space divided by coverage criteria. The work [44] also suggests that using structural coverage to measure the neural network robustness can be questionable.

Second, our results suggest that retraining with the test case does not necessarily improve robustness. For software systems, a test case which reveals a bug naturally leads to bug fixing, which "definitely" improves the 'robustness' of the system. This is not certain for DNN models. because the retrained model could be rather different from the original model, i.e., it is like a new model, due to how such models are trained (i.e., through optimization techniques which embody a lot of non-determinism and carry little theoretical guarantee).

Third, we consider it to be intuitive that defense metrics are correlated with robustness as these defense metrics are indeed less formal ways of measuring robustness (i.e., in term of how well a DNN model defends adversarial attacks).

As for the answer to RQ4, we take the consistency between different correlation analysis algorithms positively as it shows that our results are not the result of certain 'biased' correlation analysis algorithm. The second part of the answer may suggest that a testing method may have to be tailored according to different DNN architectures.

C. Discussion

The results discussed so far are mostly negative, i.e., only several defense metrics are correlated with the improvement of model robustness and existing testing methods designed based on coverage have limited effectiveness on improving the robustness of the DNN models. The results question the usefulness of coverage criteria proposed for DNN models. Indeed, a well tested (and improved by retraining) DNN through existing testing methods might produce a new model which has higher empirical accuracy on the testing set. However, the new model is not necessarily more robust than the original model against adversarial perturbations. In fact, a recent finding shows that *DNN model robustness maybe at odds with accuracy since robust classifiers are learning fundamentally different feature representations than standard classifiers [45].* For DNN models to be deployed in safety-critical applications, we believe that robustness is an as (if not more) important property as accuracy. The real question thus remains: *how should we test DNN models and make use of the testing results so that the robustness of the DNN models is improved? Or are there ways to improve the robustness of the DNN models in general?*.

To this question, we do not have a clear answer and thus it remains an open question to us. It is possible that there could be other coverage criteria which are correlated with the model robustness or the associated testing method can help improve the model robustness. It is however important that no matter what coverage is proposed, it must be thoroughly analyzed to show its effect on model robustness.

Our view is that finding adversarial samples should not be the end of DNN testing. Rather, testing DNN models should be designed in consideration of the model enhancement methods, i.e., a testing method should produce test cases which are useful according to the model enhancement methods. For instance, given the positive correlation between robustness and the defense metrics, we might want to generate test cases which could improve defense metrics such as CAV and CCV.

D. Threats to validity

First, there may be threats to validity due to the selected datasets and model structures. In this work, we regard each DNN model as a program of the same functionality and calculate different metrics on these models. We assume the metrics are valid across different model structures and conduct correlation analysis on the obtained metrics. However, some metrics are not applicable to certain model structures (e.g., MC/DC is not applicable to ResNet and GoogLeNet). Besides, the results may be biased to these specific datasets and model structures even though we are adopting the most popular datasets and state-of-the-art models.

Second, there may be threats to validity due to the limited size of data. While we are working on more datasets, model structures, etc., we could not significantly increase the scale due to the huge cost (more than $6,100$ GPU hours) of the empirical study. For more statistical significant results, more data points are helpful (or even necessary). We call upon the open source community to jointly upscale our study. To make sure that our correlation analysis results are valid, we only report the results beyond a certain significant level by measuring its p-value [46] in this work.

Third, it is shown in [47] that the adversarial samples in floating point numbers generated by FGSM, CW and JSMA may become benign after transforming back into integer images, which is called the Discretization Problem. Thus this problem may affect the results reported in the paper.

Forth, the evaluation of DNN model robustness in general is still an open and challenging research problem [48]. Although we are adopting the most popular robustness metrics, there might still be threat to validity to what extent these metrics can actually reflect the robustness of the models.

VI. Related Work

In this section, we review related works, with a focus on recent progress on 1) testing approaches which propose different testing criteria for DNN models, 2) different robustness metrics to evaluate the quality of the DNN models, and 3) state-of-the-art adversarial attacks and defense methods.

Testing of deep learning models Several recent papers proposed different coverage criteria for evaluating the effectiveness of a test set, along with different methods to generate test cases to improve the coverage criteria. For instance, DeepXplore [9] proposed the first testing criterion for DNN models, i.e., Neuron Coverage (NC), which calculates the percentage of activated neurons (w.r.t. an activation function) among all neurons. Later, DeepGauge [13] extended the idea and proposed a serial of more fine-grained multi-granularity testing criteria from both neuron level and layer level. Inspired by the MC/DC test criteria from traditional software testing, Sun et al. proposed four test criteria based on syntactic connections between neurons in adjacent layers and a concolic testing strategy to systematically improve MC/DC coverage of DNN models [12]. More recently, two surprise adequacy criteria [15] are proposed to measure the level of 'surprise' of a new test case to the training set. Our work implemented and reviewed most of the above-mentioned coverage criteria for a comprehensive evaluation. Note that some are omitted as they are extremely costly to compute.

Robustness of deep learning models In the machine learning and the formal verification community, multiple metrics are used to measure the robustness of DNN models. The Lipschitz constant was proved to be useful as a metric for Feed-forward Neural Networks by Xu, H. [21]. Segedy et al. [26] leveraged the product of Lipschitz constants for each layer as a measure of the DNN robustness and proposed Parseval Networks [35] to achieve improved robustness by maintaining a small Lipschitz constant at every hidden layer. Adversarial manipulation, which looks at the required distortion of adversarial samples is another direction. Matthias et al. intended to gave a formal guarantee on the robustness of a classifier by obtaining a robustness lower bound using a local Lipschitz continuous condition [32]. Recently, Weng et al. [22] extended their work and proposed a robustness metric called CLEVER score which is calculated using extreme value theory. Our work adopted one latest criteria from each direction.

Attack and Defense for deep learning models There is a large body work on adversarial attack and defense in recent years, which we are only able to cover the most relevant ones. In particular, we adopted three state-of-the-art attacks to generate adversarial samples, i.e., a gradient-based approach (the FGSM method [8]), a saliency map-based approach (JSMA [20]), and an optimization-based approach (C&W attack [7]). On the defense side, multiple attempts are available to obtain a relatively robust model at training phase or detect adversarial samples at runtime. For instance, adversarial training tries to include adversarial samples into consideration [49]. Another relevant direction is robust training which tries to train a robust DNN model by considering all the possible perturbation at training phase [50]. Besides, mutation testing is adopted to find adversarial samples at runtime [11]. Essentially, testing is complementary to these defense works.

VII. Conclusion

In this work, we conducted a systematic and quantitative empirical study on 100 state-of-the-art DNN models to investigate the relevance and effectiveness of recently proposed testing criteria and approaches for deep neural networks. Our study is based on a self-contained toolkit which implements all the testing coverage criteria, two robustness metrics and a large set of measurable metrics during the adversarial attack and defense pipeline. Our results obtained from correlation analysis on all these metrics from different perspectives suggest that existing testing coverage criteria have limited correlation with the robustness (or the improvement of the robustness) of DNN models. Furthermore, we provide potential directions to improve DNN testing in general by correlation analysis of robustness metrics and other kinds of metrics.

While our results are mostly negative, we believe it is important that future proposed testing criteria and methods undergo similar evaluation so as to provide evidence of their relevance. Our models, adversarial samples, and programs for calculating the metrics are publicly available and can be used as a benchmark for evaluating future research in this direction.

VIII. Acknowledgment

This work is funded by projects U1836214, 61802275 from National Science Fundation of China, National Key R&D Program of China under Grant No. 2019YFB1600700 and is supported by industrial grant MG19C08 in collaboration with Huawei.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] K. Simonyan and A. Zisserman, "Very deep coanvolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations*, 2015.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: Deep learning in android malware detection," in *Conference of the ACM Special Interest Group on Data Communication*, 2014, pp. 371–372.

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

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations*, 2015.

[9] 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*. ACM, 2017, pp. 1–18.

[10] L. Ma, F. Zhang, J. Sun, M. Xue, B. Li, F. Juefei-Xu, C. Xie, L. Li, Y. Liu, J. Zhao *et al.*, "Deepmutation: Mutation testing of deep learning systems," in *29th International Symposium on Software Reliability Engineering*, 2018, pp. 100–111.

[11] J. Wang, G. Dong, J. Sun, X. Wang, and P. Zhang, "Adversarial sample detection for deep neural network through model mutation testing," in *Proceedings of the 41th International Conference on Software Engineering*, 2019.

[12] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, and D. Kroening, "Concolic testing for deep neural networks," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018, pp. 109–119.

[13] 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*. ACM, 2018, pp. 120–131.

[14] Y. Sun, X. Huang, and D. Kroening, "Testing deep neural networks," *arXiv preprint arXiv:1803.04792*, 2018.

[15] J. Kim, R. Feldt, and S. Yoo, "Guiding deep learning system testing using surprise adequacy," in *Proceedings of the 41th International Conference on Software Engineering*, 2019.

[16] L. Inozemtseva and R. Holmes, "Coverage is not strongly correlated with test suite effectiveness," in *36th International Conference on Software Engineering*, 2014, pp. 435–445.

[17] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, "Deepsec: A uniform platform for security analysis of deep learning model," in *IEEE Symposium on Security and Privacy*, 2019.

[18] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[19] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[20] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *European Symposium on Security and Privacy*, 2016, pp. 372–387.

[21] H. Xu and S. Mannor, "Robustness and generalization," *Machine learning*, vol. 86, no. 3, pp. 391–423, 2012.

[22] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," in *6th International Conference on Learning Representations*, 2018.

[23] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[26] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations*, 2014.

[27] C. Brubaker, S. Jana, B. Ray, S. Khurshid, and V. Shmatikov, "Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations," in *IEEE Symposium on Security and Privacy*, 2014, pp. 114–129.

[28] W. M. McKeeman, "Differential testing for software," *Digital Technical Journal*, vol. 10, no. 1, pp. 100–107, 1998.

[29] K. Hayhurst, D. Veerhusen, J. Chilenski, and L. Rierson, "A practical tutorial on modified condition/decision coverage," NASA, Tech. Rep., 2001.

[30] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 97–117.

[31] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai 2: Safety and robustness certification of neural networks with abstract interpretation," in *IEEE Symposium on Security and Privacy*, 2018.

[32] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2266–2276.

[33] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Advances in Neural Information Processing Systems*, 2018, pp. 3835–3844.

[34] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," *arXiv preprint arXiv:1906.04893*, 2019.

[35] M. Ciss, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 854–863.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[37] B. Luo, Y. Liu, L. Wei, and Q. Xu, "Towards imperceptible and robust adversarial example attacks against neural networks," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 1652–1659.

[38] K. PEARSON, "Notes on the history of correlation," *Biometrika*, vol. 13, no. 1, pp. 25–45, 1920.

[39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[40] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, and A. Roy, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2016.

[41] J. B. Stroud, "Fundamental statistics in psychology and education." *Journal of Educational Psychology*, vol. 42, p. 318, 05 1951.

[42] C. Spearman, ""general intelligence," objectively determined and measured," *The American Journal of Psychology*, vol. 15, no. 2, pp. 201–292, 1904.

[43] A. Odena and I. Goodfellow, "Tensorfuzz: Debugging neural networks with coverage-guided fuzzing," *arXiv preprint arXiv:1807.10875*, 2018.

[44] Z. Li, X. Ma, C. Xu, and C. Cao, "Structural coverage criteria for neural networks could be misleading," in *Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results*. IEEE Press, 2019, pp. 89–92.

[45] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *arXiv preprint arXiv:1805.12152*, 2018.

[46] R. L. Wasserstein, N. A. Lazar *et al.*, "The asa's statement on p-values: Context, process, and purpose," *The American Statistician*, vol. 70, no. 2, pp. 129–133, 2016.

[47] Y. Duan, Z. Zhao, L. Bu, and F. Song, "Things you may not know about adversarial example: A black-box adversarial image attack," *CoRR*, vol. abs/1905.07672, 2019.

[48] J. Zhang and X. Jiang, "Adversarial examples: Opportunities and challenges," *arXiv preprint arXiv:1809.04790*, 2018.

[49] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *5th International Conference on Learning Representations*, 2017.

[50] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations*, 2018.