# POSTER: Fidelity: A Property of Deep Neural Networks to Measure the Trustworthiness of Prediction Results

Ziqi Yang
National University of Singapore
yangziqi@comp.nus.edu.sg

## ABSTRACT

With the increasing performance of deep learning on many security-critical tasks, such as face recognition and malware detection, the security issues of machine learning (ML) have become increasingly prominent. Recent studies have shown that deep learning is vulnerable to *adversarial examples*: carefully crafted inputs with negligible perturbations on legitimate samples could mislead a deep neural network (DNN) to produce adversary-selected outputs while humans can still correctly classify them. Therefore, we need an additional measurement on the trustworthiness of the results of a machine learning model, especially in adversarial settings. In this paper, we analyse the root cause of adversarial examples, and propose a new property, namely *fidelity*, of machine learning models to describe the gap between what a model learns and the ground truth learned by humans. One of its benefits is detecting adversarial attacks. We formally define fidelity, and propose a novel approach to quantify it. We evaluate the quantification of fidelity of DNNs in adversarial settings on two neural networks. Our preliminary results show that involving the fidelity enables a DNN system to detect adversarial examples with true positive rate 97.7%, and false positive rate 1.67% on a studied DNN model.

## CCS CONCEPTS

• **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computing methodologies** → **Neural networks**.

## KEYWORDS

adversarial example; neural network; transferability

**ACM Reference Format:**
Ziqi Yang. 2019. POSTER: Fidelity: A Property of Deep Neural Networks to Measure the Trustworthiness of Prediction Results. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3321705.3331005

## 1 INTRODUCTION

Deep neural networks (DNNs) have been applied and achieved excellent performance on many tasks. When DNNs are deployed in security-critical settings, such as face recognition and malware detection [1, 5, 11], designers of these systems make implicit security assumptions about DNNs. For instance, the result, including confidence values with predications, in a classification task is assumed to be trusted. However, this assumption is broken by recent work in security and machine learning communities. It has been demonstrated that adversaries can force machine learning models, including DNNs, to output adversary-selected targets using carefully crafted inputs, namely *adversarial examples* [4, 8, 10].

Motivated by adversarial attacks to DNNs, we analyse the root cause of adversarial examples. In a classification task, DNNs can output confidence values for predictions [3], trying to make the predictions analysable. However, how much confidence does a DNN model have in such confidence values remains a question. In other words, how much fidelity to the *ground truth* or human knowledge do these results have, which reflects the trustworthiness of these results. More precisely, ML models learn from training data. Adversarial examples are demonstrated not naturally drawn from the training data distribution [4], but we believe they are drawn from the underlying population distribution, because they are so close to legitimate samples that humans can correctly classify them. Therefore, the training data does not perfectly reflect the underlying population, and thus a *gap* exits between the training data (precisely, what a model learns) and the underlying population.

In this work, we propose a new property, namely *fidelity* of DNNs to name the gap. We formally define fidelity, and propose an approach to measure it. We evaluate fidelity in DNN models, but we believe that it also exists in other ML models as long as they are vulnerable to adversarial examples. Note that the significance of fidelity is not to increase the robustness of DNNs to adversarial examples [2, 9], but to provide an additional value to measure the trustworthiness of its results, which can be used to detect adversarial attacks to DNN-based systems.

## 2 FIDELITY DEFINITION

Given a model $H$, the output class of an input sample $x$ is $y = H(x)$. It follows the model distribution $P_{model}(y|x)$ learned by the model. The fidelity $F(x, y; H)$ of the model $H$ with respect to an input sample $x$ is defined as follows:

$$F(x, y; H) = 1 - |P_{model}(y|x) - P_{pop}(y|x)| \tag{1}$$

where $P_{pop}(y|x)$ is the probability distribution of $y$ given $x$ in the population. It is easy to get $F(x, y; H) \in [0, 1]$. A higher fidelity indicates the output $y$ with respect to the input $x$ matches the population distribution more closely, and a lower fidelity is contrary. Note that fidelity is significantly different from confidence values (probabilities) of predictions. The confidence values are computed following the model distribution $P_{model}(y|x)$. while fidelity is a property to reflect to how much degree the predictions match the population distribution $P_{pop}(y|x)$.

*AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand*
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6752-3/19/07.
https://doi.org/10.1145/3321705.3331005

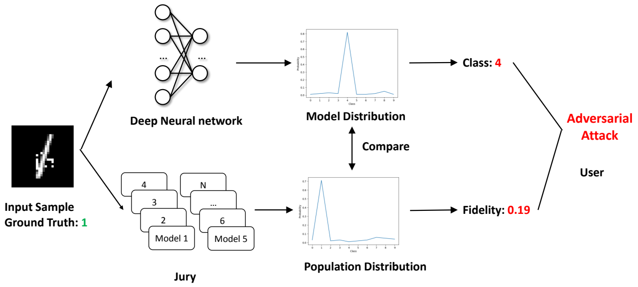

**Figure 1: Overview of our work: providing an additional value to measure the fidelity of DNNs. Fidelity is able to detect adversarial attacks.**

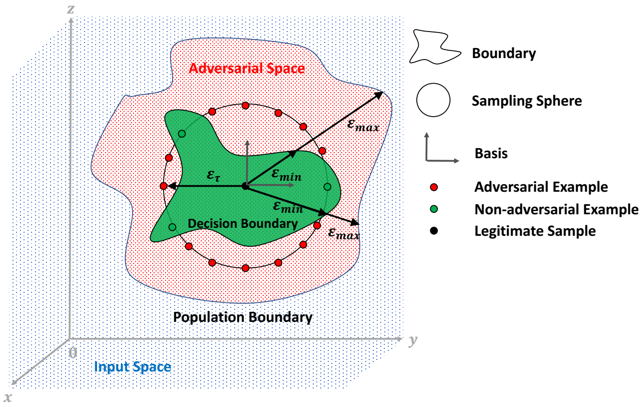

**Figure 2: Illustration of the adversarial space.**

## 3 FIDELITY MEASUREMENT

The goal of this work is not to increase the accuracy or robustness of DNNs to adversarial examples, but rather to provide an additional value to reflect the fidelity. Figure 1 shows an overview of our work. For an input sample, the original DNN model still outputs its predictions on it. We in parallel measure the fidelity of this result.

To quantitatively estimate fidelity, we propose a method that uses a set of traditional ML models, namely *jury*, to estimate the population distribution, and then compute the fidelity based on it. The intuition is that the combined action of a set of diverse and accurate models can help retain the local constancy/smoothness of the local region around an legitimate input sample, such that the adversarial example which is in such local region can be detected.

Selecting the optimal jury from universal models is expensive and usually intractable. Therefore, we constrain the selection in a predefined model pool. That is, we randomly train a set of traditional ML models on the same training set, so as to select diverse and qualified models (jury members). We propose a method of balancing the diversity and qualification of jury members to select the optimal jury. Specifically, we measure the diversity between models in adversarial settings because adversarial examples are typical instances breaking the fidelity of DNN models. We theoretically compute the dimensionality of the adversarial space (as shown in Figure 2), and then uniformly sample perturbations in such space to measure its boundaries. We propose a new metric *symmetric transfer rate* to measure the transferability of adversarial examples sampled from the adversarial space. The transferability between a pair of models is able to reflect the divergence between them in adversarial settings. We qualify jury members according to their accuracy in normal settings (on benign samples) because a random classifier could be very diverse from a well-working model, but it is not a working model. Therefore, we need to qualify a jury member first before involving it.

**Table 1: Accuracy of DNN and Pool Models**

| | | MNIST | | | CIFAR10 | | |
|---|---|---|---|---|---|---|---|
| | **Model** | **Test** | **FGSM** | **JSMA** | **Test** | **FGSM** | **JSMA** |
| **DNN** | MNIST | 99.33 | 9.97 | 6.2 | - | - | - |
| | CIFAR10 | - | - | - | 82.35 | 14.08 | 9.54 |
| **Pool** | KNN | 96.65 | 80.47 | 85.18 | 38.59 | 36.05 | 38.45 |
| | LR | 91.98 | 28.11 | 68.61 | 38.83 | 22.67 | 28.26 |
| | L-SVM | 94.74 | 36.44 | 69.46 | 38.54 | 34.53 | 37.67 |
| | R-SVM | 98.36 | 15.15 | 77.04 | 18.28 | 14.75 | 16.25 |
| | DT | 87.8 | 15.57 | 39.97 | 26.75 | 21.01 | 25.63 |
| | RF | 97.0 | 20.19 | 81.96 | 47.10 | 40.26 | 46.34 |
| | AdaB | 72.99 | 13.52 | 53.92 | 31.08 | 26.49 | 30.27 |
| | GNB | 55.58 | 10.4 | 38.19 | 29.76 | 29.8 | 29.53 |
| | QDA | 14.46 | 11.35 | 14.00 | 36.24 | 10.01 | 9.27 |
| | **Average** | 78.84 | 25.69 | 58.70 | 33.90 | 26.17 | 29.07 |

## 4 PRELIMINARY RESULTS

We evaluate the effectiveness of the fidelity estimation method by detecting adversarial samples on two canonical ML datasets: the MNIST [7] and CIFAR10 [6] datasets. We train two deep convolutional neural networks on them. We use both FGSM [4] and JSMA [8] approaches to generate adversarial samples for DNN models. We randomly train 9 traditional ML models as the predefined model pool, from which we select the jury. We evaluate the model pool on the original test set, the FGSM set and the JSMA set. The accuracy of each model on each test set is presented in Table 1. We can see that though the 9 models are affected by the adversarial examples, they are generally more robust against them than the DNN model.

### 4.1 Jury Can Estimate Fidelity

We test whether a jury can estimate fidelity by measuring the ability of the jury to detect adversarial examples crafted for the DNN model. If a jury performs well in detecting adversarial attacks, we believe that such jury estimates the population distribution well, and thus estimates the fidelity well.

The model pool consists of 9 models, so there are totally $\binom{9}{1} + \binom{9}{2} + \ldots + \binom{9}{9} = 511$ distinct combinations of jury. We enumerate the 511 candidate juries to evaluate their performance. The probability density function of false positive rate (FPR) and true positive rate (TPR) for the MNIST (CIFAR10) dataset are plotted in Figure 3. On the MNIST (CIFAR10) original test set, the mean FPR of adversarial attack detection is 9.20% (48.2%), the minimum FPR is 1.28% (25.7%), and the maximum FPRs are both 100% (with extremely few juries). On the MNIST (CIFAR10) JSMA set, the mean TPR is 97.26% (84.23%), the minimum TPR is 82.0% (74.10%), and the maximum TPRs are both 100%. On the MNIST (CIFAR10) FGSM set, the mean TPR

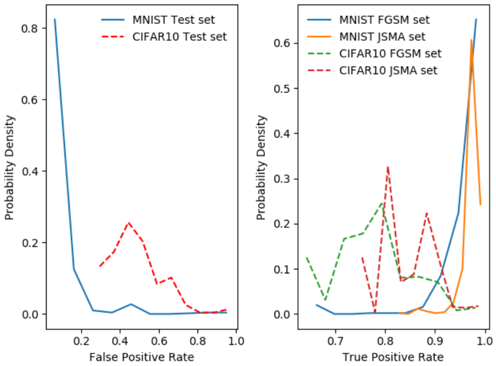

**Figure 3: Performance of juries in detecting adversarial attacks for the MNIST and CIFAR10 models.**

**Table 2: Ground Truth Rank of Juries**

| | MNIST | | CIFAR10 | |
|---|---|---|---|---|
| Rank | Score | Jury | Score | Jury |
| 1 | 0.86 | R-SVM | -1.87 | RF |
| 2 | 0.83 | R-SVM, QDA | -2.32 | KNN, RF |
| 3 | 0.81 | KNN, R-SVM | -2.32 | L-SVM, RF |
| 4 | 0.79 | KNN, R-SVM, QDA | -2.43 | KNN, L-SVM, DT, RF, GNB, QDA |
| 5 | 0.78 | R-SVM, RF | -2.45 | KNN, L-SVM, R-SVM, DT, RF, GNB, QDA |
| 6 | 0.76 | R-SVM, RF, QDA | -2.46 | KNN, LR, L-SVM, DT, RF, GNB, QDA |
| 7 | 0.75 | R-SVM, GNB | -2.47 | KNN, L-SVM, DT, RF, AdaB, GNB, QDA |
| 8 | 0.75 | KNN, R-SVM, RF | -2.48 | R-SVM, RF |
| 9 | 0.74 | KNN, R-SVM, RF, QDA | -2.48 | L-SVM, DT, RF, GNB, QDA |
| 10 | 0.73 | KNN, R-SVM, GNB | -2.48 | KNN, DT, RF, GNB, QDA |
| ... | ... | ... | ... | ... |
| 507 | -6.28 | AdaB, GNB | -8.86 | LR, R-SVM |
| 508 | -7.65 | QDA | -8.97 | LR, R-SVM, AdaB |
| 509 | -7.95 | AdaB, GNB, QDA | -8.99 | LR, AdaB |
| 510 | -9.0 | AdaB | -9.0 | AdaB |
| 511 | -9.0 | AdaB, QDA | -9.0 | R-SVM, AdaB |

is 96.26% (77.27%), the minimum TPR is 64.53% (62.38%), and the maximum TPRs are both 100%.

The results on the MNIST dataset demonstrate that the jury is able to detect adversarial examples with a respectable TPR and FPR. The TPR on the CIFAR10 dataset is comparable to that on the MNIST dataset, but the average FPR is not that good. This result comes as no surprise because the accuracy of the jury members on the CIFAR10 test set is too low to be qualified. The average accuracy (33.90%) of the pool models is even lower than the half of the accuracy (82.35%) of the DNN model. Around half of the test samples are misclassified by the juries. The FPR is expected to decrease as the accuracy of the jury members increases.

We can conclude that the jury with sufficiently qualified jury members is able to estimate the underlying population distribution well and thus can estimate the fidelity.

## 4.2 Jury Selection

The goal of selecting a good jury is to maximize TPR and minimize FPR. We rank the performance of all the 511 distinct juries in Table 2. This ranking is based on the FGSM and the JSMA approach. Considering the two approaches of crafting adversarial examples are typically used in the community, we assume such ranking as the ground truth of the jury performance to evaluate the jury selection approach in this paper. The evaluation strategy is to compare the selection result with the ground truth.

**Jury selection: balancing diversity and qualification**. We measure the diversity of the model pool and the DNN model according to the transferability of adversarial examples between them. After sampling 100 adversarial examples from the adversarial space around each sample, we compute the symmetric transfer rate of them between each pair of models, including the DNN model. Then we are able to construct the matrix of symmetric transfer rate. Such matrix is used as the measurement of the diversity between models.

We enumerate the model pool and the DNN model to find the solution, which gives the jury of "KNN" and "R-SVM" ("KNN" and "RF") for the MNIST (CIFAR10) dataset. In Table 2, the jury given by our approach ranks 3rd (2nd) in all 511 juries, which demonstrates that our approach is effective to select the best possible jury. The selected jury detects adversarial attacks with TPR 97.7% (81.9%), 96.9% (66.4%) and FPR 1.67% (30.6%) on the JSMA set, FGSM set and benign test set of the MNIST (CIFAR10) dataset.

## 5 CONCLUSION

In this paper, we propose fidelity of ML models to name the gap between what a model learns (model distribution) and what humans learn (underlying population distribution). Fidelity is able to reflect the trustworthiness of the outputs of machine learning models. We propose an approach of using a set of traditional ML models as a jury to measure the fidelity. The preliminary results of our evaluation show that a jury is effective to estimate the fidelity, with respectable high TPR and low FPR in detecting adversarial examples. We also propose a jury selection approach to optimize the fidelity measurement and evaluate the robustness of the selection approach. As on-going work, the fidelity measurement is evaluated in other task domains such as learning-based malware detection.

**Acknowledgment.** This research is supported by the National Research Foundation, Prime Minister's Office, Singapore (Grant No. NRF2016NCR-NCR002-012).

## REFERENCES

[1] George E Dahl, Jack W Stokes, Li Deng, and Dong Yu. 2013. Large-scale malware classification using random projections and neural networks. In *2013 IEEE International Conference on ICASSP*. IEEE, 3422–3426.
[2] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2015. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590* (2015).
[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.
[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[5] Eric Knorr. 2015. How paypal beats the bad guys with machine learning.
[6] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. (2009).
[7] Yann LeCun, Corinna Cortes, and Christopher JC Burges. 1998. The MNIST database of handwritten digits.
[8] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 372–387.
[9] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 582–597.
[10] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[11] Zhenlong Yuan, Yongqiang Lu, Zhaoguo Wang, and Yibo Xue. 2014. Droid-sec: deep learning in android malware detection. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 371–372.