Styx: A Data-Oriented Mutation Framework to Improve the Robustness of DNN

Meixi Liu[1,2], Weijiang Hong[1,2], Weiyu Pan[1], Chendong Feng[1], Zhenbang Chen[1], Ji Wang[1,2]
[1]College of Computer, National University of Defense Technology, Changsha, China
[2]State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, China
{liumeixi,hongweijiang17,panweiyu,fengchendong13,zbchen,wj}@nudt.edu.cn

ABSTRACT

The robustness of deep neural network (DNN) is critical and challenging to ensure. In this paper, we propose a *general* data-oriented mutation framework, called Styx, to improve the robustness of DNN. Styx generates new training data by slightly mutating the training data. In this way, Styx ensures the DNN's accuracy on the test dataset while improving the adaptability to small perturbations, *i.e.*, improving the robustness. We have instantiated Styx for image classification and proposed pixel-level mutation rules that are applicable to any image classification DNNs. We have applied Styx on several commonly used benchmarks and compared Styx with the representative adversarial training methods. The preliminary experimental results indicate the effectiveness of Styx.

CCS CONCEPTS

• **Software and its engineering** → *Software notations and tools*;

KEYWORDS

DNN, Robustness, Mutation, Adversarial examples

ACM Reference Format:
Meixi Liu, Weijiang Hong, Weiyu Pan, Chendong Feng, Zhenbang Chen, and Ji Wang. 2020. Styx: A Data-Oriented Mutation Framework to Improve the Robustness of DNN. In *35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20), September 21–25, 2020, Virtual Event, Australia.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3324884.3418903

1 INTRODUCTION

The success of Deep learning (DL) techniques can't cover up the fact that it is still challenging to ensure the safety and security of DNN-based applications, especially in safety-critical areas, such as autonomous driving [5] and flight control systems [4]. One representative threat is the existence of adversarial examples [15], which are produced by adding imperceptible perturbation to the original example but cause the DNN to produce wrong outputs.

Adversarial training [2] is an effective method for improving DNN's robustness *w.r.t.* adversarial examples. The basic idea of *adversarial training* is to retrain the DNN with the adversarial examples to improve the DNN's robustness. However, the improved robustness sacrifices the DNN's test accuracy. For example, when we use BIM [6] to train a CNN model for *CIFAR-10*, the test accuracy drops from 75.62% (using *traditional training*) to 53.84%.

According to DNN's back-propagation training mechanism [12], we observe that there may be a balance between robustness and test accuracy. If we only slightly mutate the training dataset, the model trained on the mutated dataset will have a similar test accuracy with the one trained by the original dataset. On the other hand, the model trained by the mutated training dataset will be more robust to the adversarial examples generated by small perturbations. Based on this observation, we propose a general mutation framework, called Styx. It generates the new training dataset by slightly mutating the training dataset to improve the robustness of DNN while maintaining the test accuracy[1]. In this paper, we instantiate Styx in the area of image classification and propose several *pixel-level* mutation rules. The results of the preliminary experiments on the representative benchmarks indicate the effectiveness of Styx.

2 BASIC FRAMEWORK

Figure 1 shows the basic procedure of Styx, which has a two-stage procedure. The first stage is to use Styx to generate a new training dataset which is produced by slightly mutating the original data. The second stage contains the training and evaluation. We train different DNN models by the original training dataset and the new training dataset. After that, we use different adversarial attacking methods to evaluate the robustness of the model as follows: for the set of correctly classified samples in the test dataset (denoted by $dataset_c$), we apply an adversarial attack to each sample in $dataset_c$; if the new sample is misclassified, it is an adversarial example. We record the number of samples that can be successfully attacked (represented by $\#attacked$) and define the robustness of the model:

$$Robustness = 1 - \frac{\#attacked}{\#dataset_c} \quad (1)$$

We instantiate Styx to the applications employing image classification DNNs and provides the following four mutators:

- `Zero Mutation`: To eliminate the influence of these pixels to the prediction, we reset the value of the pixel to be zero.
- `Average Mutation`: Replacing the value of the pixel with the average pixel value around it.
- `Random Mutation`: Using a random value to replace the pixel's value.
- `Gaussian Noise Mutation`: Mutating the value of a pixel by adding Gaussian noise to the original value.

ASE '20, September 21–25, 2020, Virtual Event, Australia

ACM ISBN 978-1-4503-6768-4/20/09…$15.00
https://doi.org/10.1145/3324884.3418903

[1]This is the reason why we call the framework Styx, which is a river offering invulnerability powers. Here we strengthen the training data by mutation.

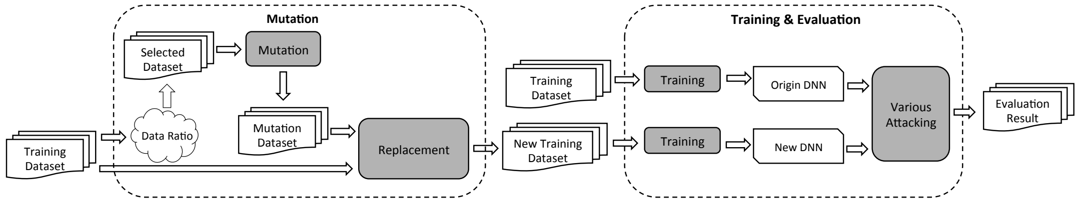


Figure 1: The basic procedure of Styx.

3 PRELIMINARY EVALUATION

Experimental Setup. Our evaluation uses three benchmarks: *MNIST*, *Fashion-MNIST* and *CIFAR-10*. We use the standard model structures (*i.e.*, the multilayer perceptron "MLP" and the convolutional neural network "CNN") provided in Keras for the benchmarks. During evaluation, we use BIM [6] and DeepFool [8] as the attacking methods and calculate the robustness by the Formula 1. IBM's adversarial-robustness-toolbox[2] is the implementation of these attacking methods. The experiments were carried out on a server with 8 cores and 32G memory. The GPU is RTX 2080 and the OS is Ubuntu Linux 16.04.

Experimental Results. Figure 2 shows the test accuracy result of different training methods. The test accuracy under *adversarial training* decreases compared with the other two training methods. Styx has a similar test accuracy with that of the *traditional training*.

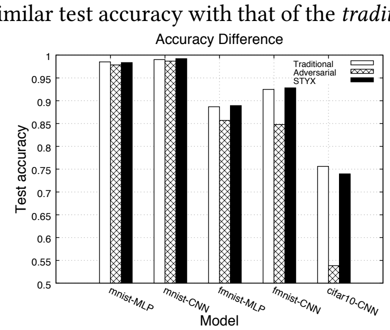
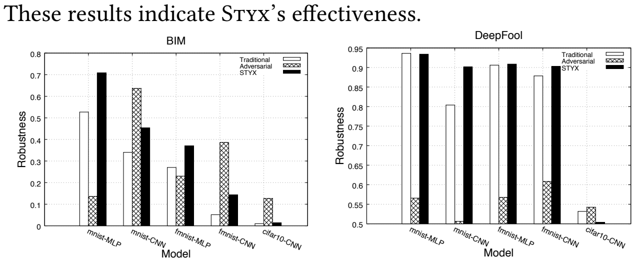


Figure 2: The Accuracy Evaluation.

Figure 3 shows the average robustness of these models. For 10 comparisons (*i.e.*, 2 attacks × 5 models), Styx improves the robustness by 9.8% (BIM) and 1.9% (DeepFool) on average, respectively. These results indicate Styx's effectiveness.



(a) Robustness under BIM (b) Robustness under DeepFool

Figure 3: The Robustness Results.

4 RELATED WORK AND OUR PLAN

Existing methods for defending against adversarial attacks and improving the robustness of DNN can be divided into three categories: adversarial retraining [2, 8, 15], network modification [9, 10], and pre-detection [3, 13]. These methods are challenged by the problems, including specific attacking defense, scalability, feasibility, *etc.* Styx is close to *adversarial training*. Styx uses a mutated training dataset for network training and prevents the over-fitting problem of the specific attacking method.

Measuring the robustness of DNN is also an active topic. In [8], the authors quantify the robustness of DNN by measuring the minimal perturbation that results in adversarial examples. In [1], the authors propose two different metrics: adversarial frequency and adversarial severity. Furthermore, many coverage criteria designed for DNN have been proposed, such as neuron coverage [11], k-multisection neuron coverage [7], the coverage criteria inspired by MC/DC [14], to name a few. Different from them, we measure the DNN's robustness from the perspective of attacking methods, and the measurement is more intuitive and realistic.

The next step lies in several aspects: 1) investigate more general mutation rules; 2) recommend the mutation strategy that results in the best robustness result; 3) apply Styx to more representative benchmarks with respect to more attacking methods.

ACKNOWLEDGEMENTS This research was supported by National Key R&D Program of China (No. 2017YFB1001802) and NSFC Program (No. 61632015, 61690203, and 61532007).

REFERENCES

[1] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya V. Nori, and Antonio Criminisi. [n.d.]. Measuring Neural Net Robustness with Constraints. In *NeurIPS 2016, pp.2613–2621, 2016*.
[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *CoRR* abs/1412.6572 (2014).
[3] Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G. Dimakis. 2017. The Robust Manifold Defense: Adversarial Training using Generative Models. *CoRR* abs/1712.09196 (2017).
[4] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *CAV*.
[5] Jiman Kim and Chanjong Park. 2017. End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars. In *CVPR 2017*. 1194–1202.
[6] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *CoRR* abs/1607.02533 (2016).
[7] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. 2018. DeepGauge: multi-granularity testing criteria for deep learning systems. In *ASE 2018*.
[8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *CVPR 2016*.
[9] Aran Nayebi and Surya Ganguli. 2017. Biologically inspired protection of deep networks from adversarial attacks. *CoRR* abs/1703.09202 (2017).
[10] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *S&P 2016*.
[11] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. In *SOSP 2017*.
[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Leaning internal representations by back-propagating errors. *Nature* 323, 6088 (1986), 318–362.
[13] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. 2017. AE-GAN: adversarial eliminating with GAN. *CoRR* abs/1707.05474 (2017).
[14] Youcheng Sun, Xiaowei Huang, and Daniel Kroening. 2018. Testing Deep Neural Networks. *CoRR* abs/1803.04792 (2018).
[15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR* abs/1312.6199 (2013).

[2] https://github.com/IBM/adversarial-robustness-toolbox