Fired neuron rate based decision tree for detection of adversarial examples in DNNs

Wang, Si; Liu, Wenye; Chang, Chip-Hong

2020

Wang, S., Liu, W., & Chang, C.-H. (2020). Fired neuron rate based decision tree for detection of adversarial examples in DNNs. Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS). doi:10.1109/ISCAS45731.2020.9180476

https://hdl.handle.net/10356/144346

https://doi.org/10.1109/ISCAS45731.2020.9180476

Fired Neuron Rate Based Decision Tree for Detection of Adversarial Examples in DNNs

Si Wang, Wenye Liu and Chip-Hong Chang
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: si003@e.ntu.edu.sg, wliu015@e.ntu.edu.sg, echchang@ntu.edu.sg

***Abstract*—Deep neural network (DNN) is a prevalent machine learning solution to computer vision problems. The most criticized vulnerability of deep learning is its susceptibility towards adversarial images crafted by maliciously adding infinitesimal distortions to the benign inputs. Such negatives can fool a classifier. Existing countermeasures against these adversarial attacks are mainly developed based on software model of DNNs by using modified training during learning or modified input during testing, modifying networks or changing loss/activation functions, or relying on add-on models for classifying unseen examples. These approaches do not consider the optimization for hardware implementation of the learning models. In this paper, a new thresholding method is proposed based on comparators integrated into the most discriminative layers of the DNN determined by their layer-wise fired neuron rates between adversarial and normal inputs. Effectiveness of the method is validated on the ImageNet dataset with 8-bit truncated models for the state-of-the-art DNN architectures. A high detection rate of up to 98% with only 4.5% of false positive rate is achieved. The results show a significant improvement on both detection rate and false positive rate compared with previous countermeasures against the most practical non-invasive universal perturbation attack on deep learning based AI chip.**

I. Introduction

Two artificial intelligence breakthroughs of epoch-making significance in the early 2010s are Microsoft's speech recognition system [1] and AlexNet system [2]. Their highly end-to-end problem-solving and exemption of extra feature extraction process have inspired blossoming deployment of deep learning in many applications [3], in particular, image classification [4], speech recognition and language processing [5]. With the evolutionary development of models and liable accessibility of the essential hardware resources for training, deep learning is also permeated into security and safety areas. Examples are its adoption in video surveillance, malware detection and self-drive cars [6].

Despite the astonished performance of deep neuron networks (DNNs), recent studies [7]–[11] have shown that they are prone to adversarial attacks by adding carefully crafted small perturbations to benign samples. The resultant distorted inputs are commonly known as adversarial examples. Apart from the most malicious image generation methods that produce image-specific distortions, Moosavi-Dezfooli et al. [12] have proposed distinct universal perturbations that can change classification results on any image. Such harmful and stealthy disturbance originally targeting software models of DNN can be successfully extended to their real physical hardware implementations, which poses severe threat to the deployment of DNNs in safety and security critical applications.

To tackle this problem, a series of defenses that focus on either robustness improvement or adversarial example detection have been proposed. The former attempts to restore the correct prediction of a perturbed input without affecting the classification on normal samples. Common strategies are adversarial training, gradient masking and input transformation. Adversarial training mixes the clean dataset with off-the-shelf adversarial examples in the training phase to increase the model tolerance towards adversarial noise patterns. However, it is found to be vulnerable to new attacks [8]. Gradient masking obfuscates the gradient information exploited by most adversarial attacks for adversarial example generation, but it fails to successfully defend against black-box attack due to transferability of adversarial examples [13], [14]. Input transformation reconstructs images to mitigate the effect of aberration. Recent detection approaches [15]–[17] include sample statistics, detector training and prediction inconsistency. Sample statistics detects the difference between adversarial and normal samples by statistically analyzing the data. Detector training modifies the network architecture to incorporate a module dedicated for identifying malicious inputs. Prediction inconsistency assumes that the perturbations are model-specific and measures the degree of inconsistency in predictions from different classifiers.

Most protection methods are either evaluated using small-scale datasets that may not be scalable or designed and investigated from software perspective without considering the attack and defense practicality when they are mounted on a hardware-oriented (with input truncation and fixed-point arithmetic) DNN model. In this paper, we propose a detection technique directed by the difference in layer-wise fired neuron rate between the harmful and benign inputs. The detector construction strategy belongs to the category of sample statistics but the detector extracted for integration into the hardware DNN is simpler than the additional module used by detector training and prediction inconsistency. Our solution is inspired by the observation that there exists a disparity in the number of fired neurons between adversarial and benign samples and the gap between their statistical distributions varies across layers. This implies the presence of bias in the distribution of neuron activities in some layers, which can be exploited for adversarial distinguishability. Dominant layers with high discriminability against adversarial examples are used to successively filter out the malicious images. A decision tree is used to determine the most discriminative layer (MDL) of the trained DNN and its threshold at each level. This fired neuron rate (FNR) based adversaries detection method is implemented on four different state-of-the-art 8-bit truncated Convolutional Neural Network (CNN) models targeting ImageNet classification.

The rest of this paper is organized as follows. Section II briefly introduces the four target CNN architectures and the universal perturbation method for generating adversarial samples. Section III elaborates the construction of the decision tree detector based on most discriminative FNR layers. Section IV presents and discusses the simulation results. Section V concludes the paper.

II. Preliminaries

A. Deep Neural Network Architectures

The four deep learning models used in our experiments are listed below. Their basic properties are outlined in Table I.

AlexNet: AlexNet [2] is of momentous significance in the field of computer vision. It has a relatively simple structure compared with other CNN architectures. AlexNet consists of

TABLE I: Four CNN models used for the evaluation

Models	Year	# layers	# parameters	Top-1/5 accuracies
AlexNet [2]	2012	8	60 million	56.9% / 80.1% [18]
VGG 16 [19]	2014	16	138 million	71.5% / 89.8% [18]
VGG 19 [19]	2014	19	144 million	71.1% / 89.8% [18]
MobileNet [20]	2017	28	4.3 million	70.9% / 89.9% [18]

5 convolutional layers followed by 3 fully connected layers. Except the first two layers, which use 11×11 and 5×5 kernels, the other layers use a 3×3 kernel.

VGGNet: VGGNet [19] is a family of CNN architectures including VGG 16 and VGG 19. They inherit similar model structure as AlexNet [2], but with increased depth. Moreover, VGGNets use 3×3 filter for all the convolutions.

MobileNet: MobileNet [20] is a family of efficient DNN models dedicated to resource-constrained applications. They factorize a standard convolution into a depth-wise convolution followed by a point-wise convolution to reduce the number of parameters and computational complexity. The former uses input-channel-length number of single-channel filters to convolve with respective input channel. The intermediate feature map is convolved with filters of size 1×1×(depth of intermediate feature map). MobileNet-v1 is used in our experiment.

B. Universal Perturbation

Universal perturbation [12] is a distortion pattern that causes misclassification without considering the variation in images. It is an extended version of DeepFool [11] for changing the prediction with minimum perturbation. The image-agnostic adversarial noise pattern can be determined by [12]:

$$v := \left\{ v | Err(X_v) := \frac{1}{m} \sum_{i=1}^{m} 1_{\widehat{k}(x_i+v) \neq \widehat{k}(x_i)} \geq 1 - \delta \right\}, \quad s.t. \, \|v\|_p \leq \xi. \tag{1}$$

where v denotes a distortion vector that produces wrong classification for most samples in X. X is a set of clean images $\{x_1, ..., x_m\}$ collected from a distribution. $\widehat{k}$ represents a classifier that outputs an approximate label $\widehat{k}(x)$ for each input x and X_v denotes a set of tampered inputs $\{x_1 + v, ..., x_m + v\}$. The accuracy of the perturbed data points is quantified by δ and the extent of distortion v is limited by ξ. The algorithm computes the minimum distortion required to send the current input to the decision boundary and adds the calculated perturbation to the instance of universal distortion. The updated universal aberration is then augmented to the incoming clean input to prepare for the next iteration.

III. Proposed Fired Neuron Rate Directed Detector

A. Threat Model

The attackers are assumed to be equipped with information about the target integrated DNN classifier. However, they can neither access nor physically tamper its inner circuit. The attackers may produce adversarial images using the white-box technique to confuse the classifier. Attacks that inject infinitesimal perturbations, such as Fast Gradient Sign Method (FGSM) [8], C&W attack [9] and DeepFool [11], while effective on software models, fail terribly on hardware-oriented DNNs. This is because the fractional malicious distortions are largely eliminated upon the floating-point to fixed-point conversion into the 8-bit color space and quantization carried out internally for efficient hardware implementation. Furthermore, the search space of an ImageNet image by Jacobian-based saliency map approach (JSMA) [10] is enormous. By iteratively perturbing one or two pixels a time, this method is time consuming and the high intensity perturbed pixels are easily detectable. Therefore, this work focuses mainly on universal perturbations that can be more practically and efficiently carried by a non-invasive attack on a hardware-oriented DNN model.

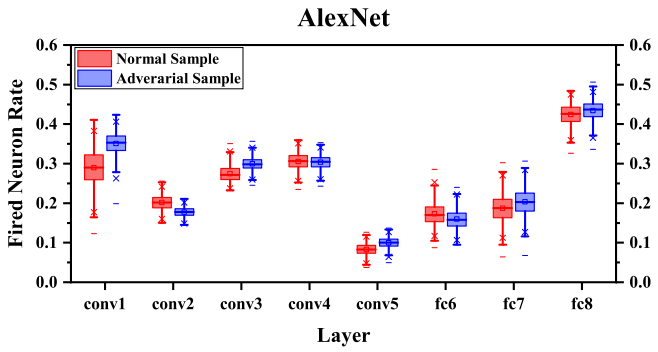

Fig. 1: FNR distribution in different layers of AlexNet.

B. Most Discriminative Layer (MDL)

The fired neuron rate (FNR) is defined as the number of positive activations in the output feature map over the total number of activations. The hypothesis of the proposed method is that tampered and clean images have different FNRs, and such disparity is reinforced in some dominant layers. To evaluate the hypothesis, 1000 correctly classified clean images and 1000 successfully misclassified adversarial images crafted using universal perturbation algorithm are tested for each model. The FNR of each layer is calculated for each image. The distribution of FNR for normal and adversarial images in various layers of different models are shown in Figs. 1 to 4. For MobileNet, some layers that have nearly the same distribution for both benign and malicious samples are removed from Fig. 4 to refrain them from cluttering the useful information. It is evident that there exists a large deviation among different layers in terms of discriminability between the clean and tampered samples based on the FNR for each model. Besides, the layers that possess high differentiability are all convolutional layers, for example, $conv1$, $conv2$ and $conv5$ layers in AlexNet, $conv2\_2$, $conv4\_3$ and $conv5\_1$ layers in VGG 16, $conv4\_1$, $conv4\_4$ and $conv5\_4$ layers in VGG 19, and $conv2\_2/dw$, $conv3\_1/sep$ and $conv6/dw$ layers in MobileNet. This finding can be attributed to the feature extraction role of convolutional layers, which tends to be heavily weighted in the process of crafting powerful adversarial perturbations. It also implies that some layers are more responsive to the addition of distortions than others. The layer that has the least overlap between the FNR distributions of the normal and adversarial images is designated as the most discriminative layer (MDL).

C. Proposed Decision Tree Detector Framework

Our proposed adversarial example detection method divides a set of clean and adversarial images into two bins by thresholding the FNR distributions of the MDL progressively until there is no improvement in the correct classification rate of both types of images. Once the threshold in each node of the binary decision tree is established, the thresholding operation of the entire decision tree can be integrated into the DNN to determine if an input image is benign or adversarial. The detection framework is illustrated in Fig. 5, where $v_{i,j}$ denotes the j-th node at the i-th level of the decision tree. $i = 1, 2, \cdots, i_{max}$, $j = 1, 2, \cdots, j_{max}$ and $j_{max} \leq 2^{i-1}$. $\tau_{i,j}$ is the threshold for separating the malicious and benign samples at node j of level i. Each node $v_{i,j}$ consists of a string variable $MDL_{i,j}$ for the name of the MDL of the target CNN, a degree of impurity $\delta_{i,j}$ and a type $t_{i,j}$. Each node $v_{i,j}$ is generated by assigning to it a set of

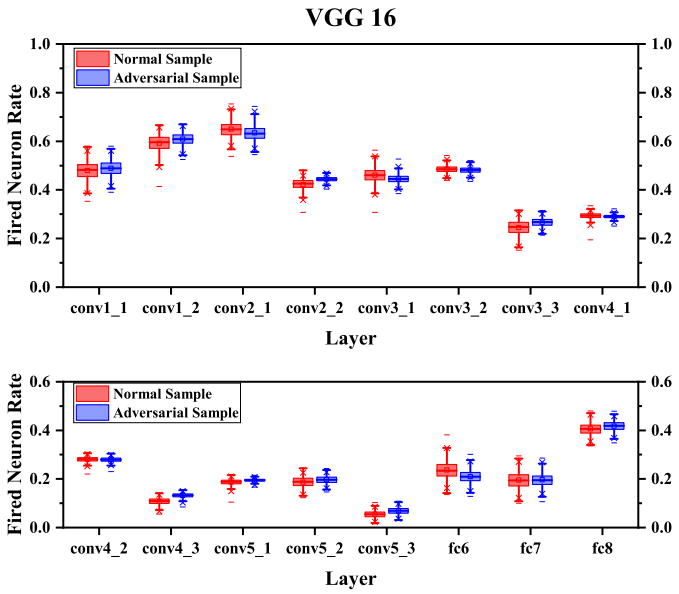

Fig. 2: FNR distribution in different layers of VGG 16.

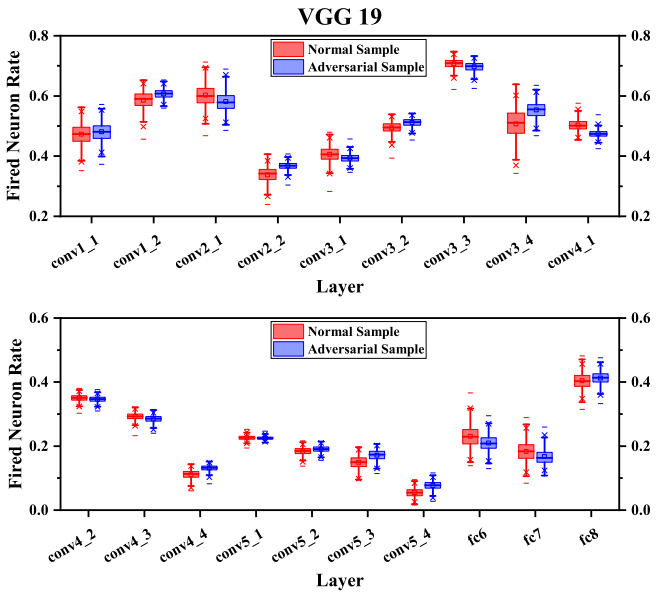

Fig. 3: FNR distribution in different layers of VGG 19.

training images $I_{i,j}$. $\delta_{i,j}$ is measured by the Gini index [21]. It is calculated as follows:

$$\delta_{i,j} = 1 - ({P_a}^2 + {P_b}^2) \tag{2}$$

where P_a and P_b are the proportions of adversarial and benign images, respectively in $I_{i,j}$. It has a minimum value of 0 if the images in $I_{i,j}$ are all adversarial or all clean, and a maximum value of 0.5 if $I_{i,j}$ has equal proportion of adversarial and benign images. $t_{i,j} = 0$ if majority of the images in $I_{i,j}$ are clean and 1 otherwise. The tree terminates at depth i when there is no improvement in the correct classification rate p_i determined by a separate set of validation images I_v. The procedure for the construction of the decision tree is given as follows:

Step 1: Set $i = 1, j = 1$ and add a root node $v_{i,j}$ to the decision tree. Assign N randomly selected malicious and N labelled benign images to $I_{i,j}$ as the training images and M randomly selected malicious and M labelled benign images to I_v. Set $p_i = 0$ and $\delta_{i,j} = 0.5$.

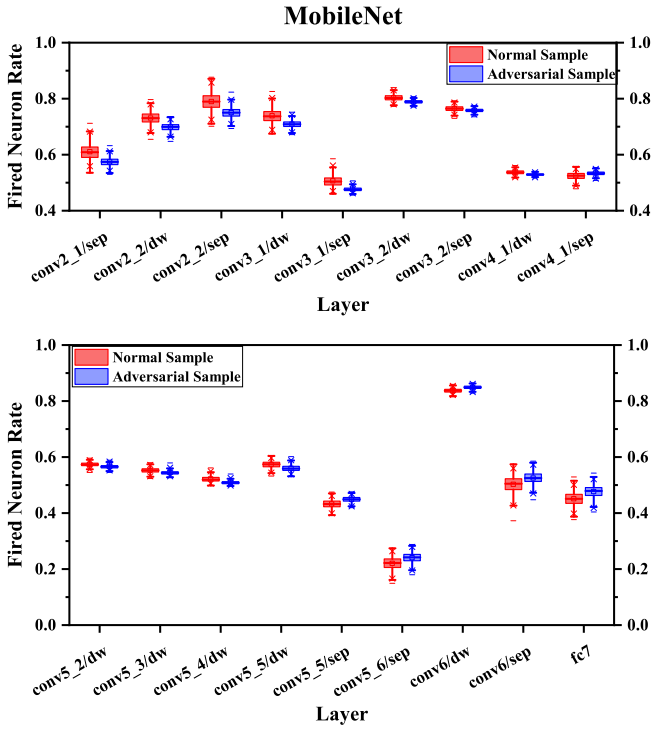

Fig. 4: FNR distribution in different layers of MobileNet.

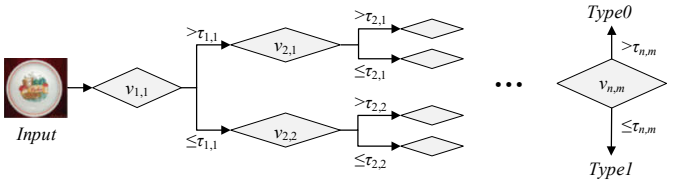

Fig. 5: Decision tree for successive thresholding of normal and adversarial FNR distributions in MDL.

Step 2: Input the images $I_{i,j}$ to the target trained DNN to obtain the FNR distribution for each type of images in each layer. Select the MDL L of DNN and record $MDL_{i,j} = L$. Determine and record the threshold $\tau_{i,j}$ that maximizes the inter-class variance of the two distributions of $MDL_{i,j}$.

Step 3: Add two children nodes, $v_{i+1,2j-1}$ and $v_{i+1,2j}$, to node $v_{i,j}$. Assign those images in node $I_{i,j}$ that have FNR in $MDL_{i,j} \leq \tau_{i,j}$ to $I_{i+1,2j-1}$ of its left child node and the remaining images to $I_{i+1,2j}$ of its right child node. Set the current predicted label t of the child node to 0 (benign) or 1 (adversarial) by majority type of its images. Calculate $\delta_{i+1,2j-1}$ and $\delta_{i+1,2j}$. Input the validation images I_v to the target DNN to obtain the FNR for each image. Calculate their correct classification rate p_i.

Step 4: If $p_{i+1} \leq p_i$, terminate. Otherwise, set $i = i + 1$ and repeat Steps (2)-(4) for any leaf node $v_{i,k}$, $k \in \{2j-1, 2j\}$ that has $\delta_{i,k} > 0$.

At the end of the process, each node of the decision tree will contain a threshold associated with a MDL of the target DNN. Each leaf node will also have a type label for integration into the CNN hardware for detecting if a presented input image is highly probable to be adversarial or benign. The advantage of using multiple instead of a single threshold is that malicious perturbation is likely to affect multiple levels of feature abstraction and a single threshold may not be able to sufficiently minimize the intra-class variance of the two distributions. For example, although $conv1$ layer in AlexNet has the most discriminative FNR distributions between the tampered

TABLE II: Parameters of the final decision trees for the four DNN models.

Models	Depth	MDL	τ
AlexNet	6	$MDL_{1,1}$: $conv1$	0.321
		$MDL_{2,1}$: $conv2$	0.193
		$MDL_{2,2}$: $conv3$	0.281
VGG 16	7	$MDL_{1,1}$: $conv4\_3$	0.123
		$MDL_{2,1}$: $conv2\_2$	0.444
		$MDL_{2,2}$: $conv4\_3$	0.128
VGG 19	8	$MDL_{1,1}$: $conv4\_1$	0.491
		$MDL_{2,1}$: $conv5\_4$	0.064
		$MDL_{2,2}$: $conv3\_2$	0.528
MobileNet	6	$MDL_{1,1}$: $conv3\_1/sep$	0.488
		$MDL_{2,1}$: $conv6/dw$	0.837
		$MDL_{2,2}$: $conv5\_4/dw$	0.508

TABLE III: Correct detection and false positive rates.

Models	Configuration	Detection rate (TPR) (%)	FPR (%)
AlexNet	single-level	87.70	28.00
	6-level	89.60	17.10
VGG 16	single-level	81.20	11.70
	7-level	91.90	6.70
VGG 19	single-level	91.60	21.80
	8-level	93.20	6.60
MobileNet	single-level	94.10	18.10
	6-level	98.10	4.50

and clean inputs according to Fig. 1, there is still a large unresolved overlap.

IV. Implementation Results and Discussions

The proposed method is examined using 8-bit truncated Caffe models for AlexNet, VGG 16, VGG 19 and MobileNet with MatLab interface to a PC equipped with an E5-1630 v4 3.70GHz CPU, 16GB system memory, and a GeForce GTX 1070Ti.

A. Experimental Setup

2000 correctly predicted benign images from ImageNet dataset and their corresponding 2000 successfully misclassified tampered images by each network architecture are used in this experiment. These 4000 images are randomly divided into two groups, each of which contains 1000 clean samples and 1000 tampered samples for each model. One group is utilized for the determination of the MDLs and τ of each decision node. The other group is used for testing. Five-fold cross-validation technique is used to prevent the decision tree algorithm from overfitting or selection bias. It partitions the training database into five groups. For each group, validation is performed after the completion of the analysis on the remaining groups in each round. Then the average performance of the five rounds is considered. The final depth upon termination, and the MDLs and threshold values in the first two levels of the first round (due to limited space) of the decision tree for each CNN architecture are summarized in Table II.

B. Results

Table III shows the results of the proposed method on detection rate of unseen adversarial inputs to each model. The single-level method refers to the use of only one MDL with one threshold at the root of the decision tree. In the testing stage, majority voting of the five rounds of prediction is conducted. TPR and FPR are acronyms for true positive rate and false positive rate, respectively. They are calculated by:

$$TPR = TP/(TP + FN) \tag{3}$$

$$FPR = FP/(FP + TN) \tag{4}$$

TABLE IV: Comparison with feature squeezing method.

Metrics	This work	Feature squeezing [15]
Model	MobileNet	MobileNet
Truncated model	Yes	No
Dataset	ImageNet	ImageNet
Attack	Universal Perturbation	DeepFool
Configuration	6-level	3-squeezer
Detection rate (%)	98.10	78.60
FPR (%)	4.50	8.33

where TP, TN, FP and FN are acronyms for true positives, true negatives, false positives and false negatives, respectively. In our context, an 'adversarial' image is considered as a positive class so that a FP implies a benign sample is wrongly predicted as an adversarial sample. Our results show that all the multi-level decision trees have higher detection rates and lower FPRs than their single-level counterparts. In addition, the detection accuracy of the multi-level decision tree also increases with the CNN model depth. With the increasing depth of the DNN, the responsiveness and overall ability of the MDLs to discriminate finer difference between tampered and untampered image feature information may also increase. However, a comparatively high FPR occurs for all the single-level detectors and the multi-level decision tree of AlexNet because of the lack of loss control in the training stage. A better parameter tuning method that imposes a loss constraint can be considered in the near future to improve this aspect.

The performance is also compared with feature squeezing method [15] in Table IV. The universal perturbation considered in our countermeasure is originated from DeepFool, which is the attack algorithm considered by feature squeezing method. Our proposed method, though assessed using 8-bit truncated version of the model, still has a much higher detection rate and a lower FPR than feature squeezing, with nearly 20% of improvement in detection rate. Furthermore, our results are based on 1000 adversarial and benign images whereas only one-tenth of these two different types of images are tested by [15]. With ten times larger sample size, our evaluation is more rigorous and hence the results are more reliable.

V. Conclusions

A FNR based defense is proposed for adversarial example detection. Greater distinguishability between adversarial and natural images are observed in their FNR distributions at some convolution layers. The optimal decision threshold at selected MDL is obtained successively in a tree algorithm by applying a five-fold cross-validation technique. 8-bit truncated version of four DNN models are used for the assessment to account for their typical fixed-point implementation in hardware. The simulation results show a significant improvement in detection accuracy of adversarial inputs compared with feature squeezing method. Our future work is to map the read and write memory access patterns into layer-wise FNR in order to integrate appropriate comparators (for thresholding) more efficiently into the identified MDLs of the DNN accelerator.

Acknowledgment

This research is supported by the National Research Foundation, Singapore, under its National Cybersecurity Research & Development Programme / Cyber-Hardware Forensic & Assurance Evaluation R&D Programme (Award: CHFA-GC1-AW01).

References

[1] L. Deng *et al.*, "Recent advances in deep learning for speech research at Microsoft," in *Proc. 2013 IEEE Int. Conf. Acoustics, Speech Signal Processing*, Vancouver, Canada, May 2013, pp. 8604–8608.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances Neural Info. Processing Syst.*, Harrahs and Harveys, Lake Tahoe, USA, Dec. 2012, pp. 1097–1105.

[3] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[4] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1561/2200000006

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[6] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[7] C. Szegedy *et al.*, "Intriguing properties of neural networks," presented at Int. Conf. Learning Representations (ICLR), Scottsdale, Arizona, USA, May 2013. [Online]. Available: https://arxiv.org/abs/1312.6199

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," presented at Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, May 2014. [Online]. Available: http://arxiv.org/abs/1412.6572

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. 2017 IEEE Symp. Security Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 39–57.

[10] N. Papernot *et al.*, "The limitations of deep learning in adversarial settings," in *Proc. 2016 IEEE Euro. Symp. Security Privacy (EuroS&P)*, Saarbrcken, Germany, Mar. 2016, pp. 372–387.

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Computer Vision Pattern Recogn.*, Las Vegas, Nevada., USA, Jun. 2016, pp. 2574–2582.

[12] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comp. Vision Pattern Recogn.*, Honolulu, Hawaii, USA, Jul. 2017, pp. 1765–1773.

[13] N. Papernot *et al.*, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. 2016 IEEE Symp. Security Privacy (SP)*, San Jose, CA, USA, May 2016, pp. 582–597.

[14] ——, "Practical black-box attacks against machine learning," in *Proc. 2017 ACM Asia Conf. Comp. Comm. Security*, Abu Dhabi, United Arab Emirates, 2017 Apr., pp. 506–519.

[15] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," presented at 2018 Net. Distributed Syst. Security Symp. (NDSS), San Diego, CA, USA, Feb. 2018. [Online]. Available: https://arxiv.org/abs/1704.01155

[16] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," presented at 5th Int. Conf. Learning Representations (ICLR), Toulon, France, Apr. 2017. [Online]. Available: https://arxiv.org/abs/1702.04267

[17] J. Wang, J. Sun, P. Zhang, and X. Wang, "Detecting adversarial samples for deep neural networks through mutation testing," vol. abs/1805.05010, 2018. [Online]. Available: http://arxiv.org/abs/1805.05010

[18] D. Su *et al.*, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *Proc. Euro. Conf. Comp. Vision (ECCV)*, Munich, Germany, Sep. 2018, pp. 631–648.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[20] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[21] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - a survey," *Trans. Sys. Man Cyber Part C*, vol. 35, no. 4, pp. 476–487, Nov. 2005. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2004.843247