Received: 13 July 2022 | Revised: 18 July 2022 | Accepted: 24 August 2022 | Published online: 5 October 2022

Journal of Computational and Cognitive Engineering
2022, Vol. 1(4) 159–164
DOI: 10.47852/bonviewJCCE2202320

BON VIEW PUBLISHING

RESEARCH ARTICLE

# Block Switching: A Stochastic Approach for Deep Learning Security

**Xiao Wang[1], Siyue Wang[2],*, Pinyu Chen[3], Xue Lin[2] and Peter Chin[1]**

[1]*Boston University, USA*

[2]*Northeastern University, USA*

[3]*IBM Research, USA*

**Abstract:** Recent study of adversarial attacks has revealed the vulnerability of modern deep learning models. That is, subtly crafted perturbations of the input can make a trained network with high accuracy and produce arbitrary incorrect predictions, while maintaining imperceptible to human vision system. In this paper, we introduce Block Switching (BS), a defense strategy against adversarial attacks based on stochasticity. BS replaces a block of model layers with multiple parallel channels, and the active channel is randomly assigned in the run time, hence unpredictable to the adversary. We show empirically that BS leads to a more dispersed input gradient distribution and superior defense effectiveness compared with other stochastic defenses such as stochastic activation pruning. Compared to other defenses, BS is also characterized by the following features: (i) BS causes less test accuracy drop; (ii) BS is attack-independent; and (iii) BS is compatible with other defenses and can be used jointly with others.

**Keywords:** deep learning security, adversarial attack, stochastic defense, DNN robustness

## 1. Introduction

Powered by rapid improvements of learning algorithms (He et al., 2016; LeCun, 2015; Krizhevsky et al., 2012; Zhao et al., 2019), computing platforms (Abadi et al., 2016; Jia et al., 2014), and hardware implementations (Han et al., 2016; Li et al., 2019), deep neural networks become the workhorse of more and more real-world applications, many of which are security critical, such as self-driving cars (Bojarski et al., 2016) and image recognition (Parkhi et al., 2015; He et al., 2016; Krizhevsky et al., 2012; Zhao et al., 2017; Wang et al., 2018c), where malfunctions of these deep learning models lead to serious loss.

However, the vulnerability of deep neural networks against adversarial attacks is discovered by Szegedy et al. (2013), who show that in the context of classification, malicious perturbations can be crafted and added to the input, leading to arbitrary erroneous predictions of the target neural network, while the perturbations can be small in size and scale or even invisible to human eyes.

This phenomenon triggered wide interests of researchers, and a large number of attacking methods have been developed. Some typical attack methods include Fast Gradient Sign Method (FGSM) by Goodfellow et al. (2015), Jacobian-based Saliency Map Attack by Papernot et al. (2016a), and CW attack by Carlini and Wagner (2017b). These attacks utilize gradients of a specific object function with respect to the input and design perturbations accordingly in order to have a desired output of the network. Among the attacks, CW attack is known to be the strongest and is often used as a benchmark for evaluating model robustness.

In the meantime, a rich body of defending methods has been developed, attempting to improve model robustness in different aspects. Popular directions include adversarial training (Madry et al., 2017), detection (Grosse et al., 2017; Metzen et al., 2017), input rectification (Das et al., 2017; Xie et al., 2017), and stochastic defense (Dhillon et al., 2018; Wang et al., 2018b; Wang et al., 2018a; Wang et al., 2019). However, although these defenses alleviate the vulnerability of deep learning to some extent, they are either shown to be invalid against countermeasures of the adversary (Carlini & Wagner, 2017a) or require additional resources or sacrifices. A significant trade-off of these methods is between defense effectiveness and test accuracy, where a stronger defense is often achieved at the cost of worse performance on clean examples (Wang et al., 2019).

Motivated by designing defense method with less harm on test accuracy, in this article we introduce Block Switching (BS) as an effective stochastic defense strategy against adversarial attacks. BS involves assembling a switching block consisting of a number of parallel channels. Since the active channel in the run time is random, it prevents the adversary from exploiting the weakness of a fixed model structure. On the other hand, with proper training, the BS model is capable of adapting the switch of active channels and maintains high accuracy on clean examples. As a result, BS achieves drastic model variation and thus has strong resistance

*Corresponding author: Siyue Wang, Northeastern University, USA. Email: wang.siy@husky.neu.edu

against adversary without noticeable drop in legitimate accuracy. The nature of BS also enables its usage jointly with other types of defenses such as adversarial training.

Our experimental results show that a BS model with five channels can reduce the fooling ratio (the percentage of generated adversarial examples that successfully fool the target model) of CW attack from 100% to 21.0% on MNIST dataset and to 22.2% on CIFAR-10 dataset respectively with very minor testing accuracy loss on legitimate inputs. As a comparison, another recent stochastic defense stochastic activation pruning (SAP) only reduces the fooling ratio to 32.1% and 93.3% given the same attack. The fooling ratio can be further decreased with more parallel channels.

The rest of this article is organized in the following way: In Section 2, we introduce related works in both attacking and defending sides. The defense strategy and analysis are given in Section 3. Experimental results are given in Section 4. And Section 5 concludes this work.

## 2. Adversarial Attack

**FGSM.** FGSM (Goodfellow et al., 2015) utilizes the gradient of the loss function to determine the direction to modify the pixels. They are designed to be fast, rather than optimal.

Specifically, adversarial examples are generated as follows:

$$x' = x - \epsilon \cdot \mathrm{sign}(\nabla(\mathrm{loss}_{F,t}(x))) \tag{1}$$

where $\epsilon$ is the magnitude of the added distortion, $t$ is the target label. Since it only performs a single step of gradient descent, it is a typical example of "one-shot" attack.

**CW.** Carlini & Wagner (CW) attack (Carlini & Wagner, 2017b) generates adversarial examples by solving the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad D(\delta) + c \cdot f(x + \delta) \\ &\text{subject to} \quad x + \delta \in [0, 1]^n \end{aligned} \tag{2}$$

where $c > 0$ controls the relative importance between the distortion term $D$ and loss term $f$. The loss term $f$ takes the following form:

$$f(x + \delta) = \max(\max\{Z(x + \delta)_i : i \neq t\} - Z(x + \delta)_t, -\kappa) \tag{3}$$

where $\kappa$ controls the confidence in attacks.

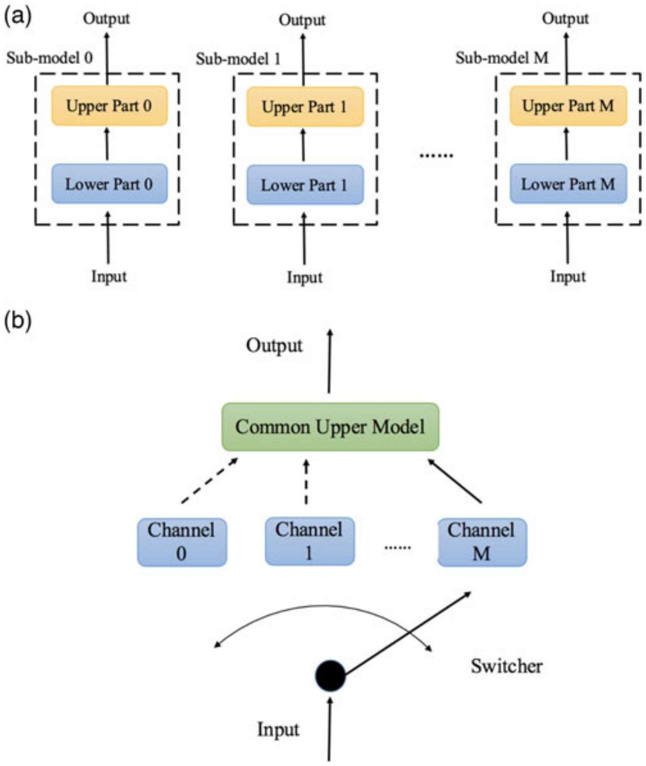
## 3. Method

### 3.1. Block switching implementation

Training a BS model involves two phases. In the first phase, a number of sub-models with the same architecture are trained individually from random weights initialization. With the training process and data being the same, these models tend to have similar characteristics in terms of classification accuracy and robustness, yet different model parameters due to random initialization and stochasticity in the training process.

After the first round of training, each sub-model is split into two parts. The lower parts are grouped together and form the parallel **channels** of the switching block, while the upper parts are discarded. The switching block is then connected to a randomly initialized common upper model as shown in Figure 1. In the run time, a random channel is selected to be **active** that processes the input while all other channel remains inactive, resulting in a stochastic model that has different behavior at different time.

**Figure 1**
**The steps of assembling a block switching. (a): Sub-models are trained individually. (b): The lower parts of sub-models are used to initialize parallel channels of block switching**



The whole BS model is then trained for the second round on the same training dataset in order to regain classification accuracy. In this phase, the common upper model is forced to adapt inputs given by different channels so that a legitimate example can be correctly classified given whichever channel is active. Usually, this phase is much faster than the first round of training since the parallel channels are already trained.

### 3.2. Defense analysis

Let $Y = \widetilde{F}(x)$ denoted the learned mapping of a stochastic model. Note that $\widetilde{F}$ is a stochastic function and now $Y$ is a random variable. The defending against adversarial attacks can be revealed in two aspects.

- **Stochasticity of Inference**: Since $Y = \widetilde{F}(x)$ is a random variable, an adversarial example that fools an instance $F^1$ of the stochastic model $\widetilde{F}$ sampled at $t_1$ may not be able to $F^2$ sampled at $t_2$.
- **Stochasticity of Gradient:** Due to the stochasticity of the network, the gradient of attacker's objective loss with respect to the input is also stochastic. That is, the gradient backpropagated to the input is just an instance sampled from the gradient distribution. And this instance may not represent the most promising gradient descent direction.

Note that these two aspects are actually correlated. From the attacker's point of view, the goal is to find $\arg\max_{x} \mathbb{E}[A(\widetilde{F}(x), T)]$ where $A(\cdot)$ outputs 1 if the attack is successful and 0 otherwise, and $T$ is the target class. Therefore, the attacker is benefited from using stochastic gradients other than gradients from a fixed model instance, in order to generate adversarial examples that are robust to model variation. In another word, this means the adversary cannot benefit from simply disabling the variation of the stochastic model and craft perturbations using a fixed model instance.

The above analysis holds for any stochastic model but the question is what makes a good randomization strategy against adversarial attacks? Intuitively, a good randomization strategy should cause the input gradients to have wider distributions. In an extreme case, if the gradient direction is uniformly distributed, performing gradient descent is no better than random walking, which means the attacker cannot take any advantage of the target model.

Knowing this, we explain why BS performs better than existing stochastic strategies such as SAP. In Figure 2, we visualize gradient distributions under CW attacks to a SAP model and a BS model, respectively. We observe that the gradient (of the attacker's object function w.r.t the input) distribution of the SAP model is unimodal and concentrated, while the gradient of BS has a multimodal distribution in a wider range. This distribution indicates that it is harder to attack BS than SAP which is verified by our experiment results in Section 4.

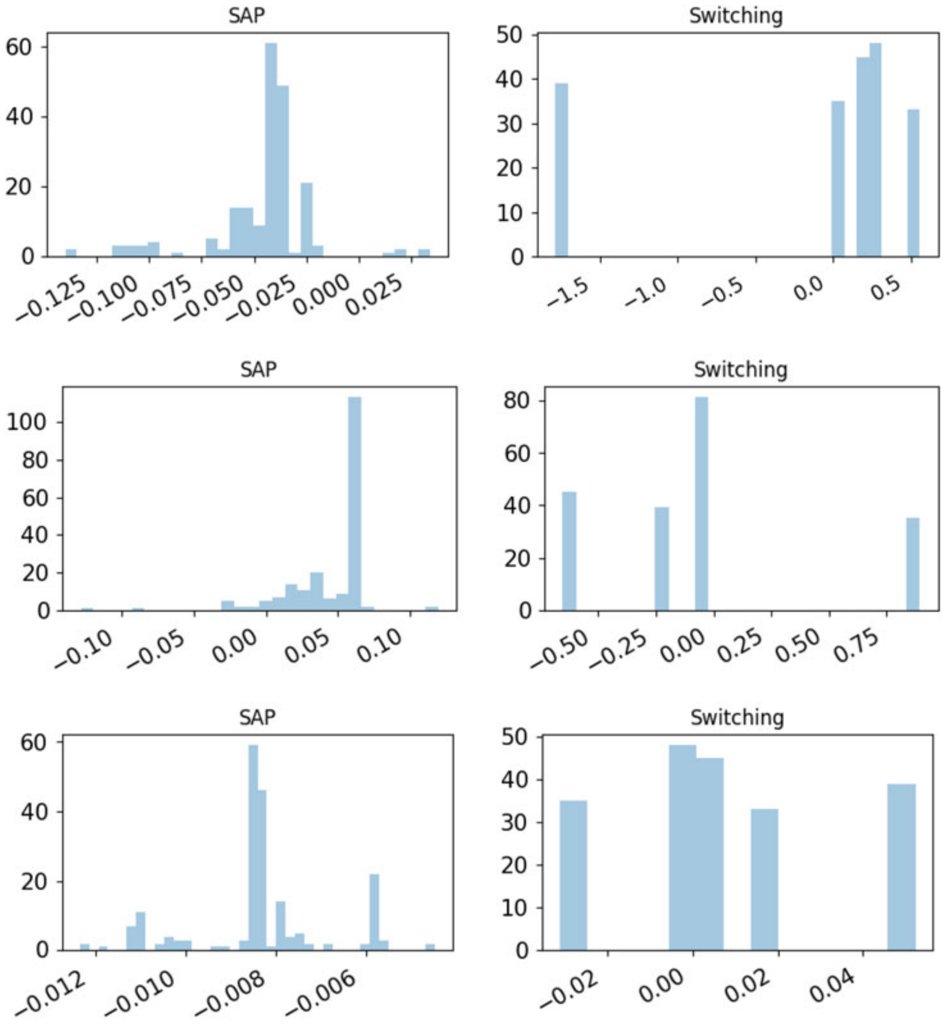
Usually, dramatic variations of the stochastic model tend to harm classification accuracy on clean inputs. That is why in SAP, smaller activation outputs have more chance to be dropped. The reason that BS is able to maintain high test accuracy even facing drastic change of the model is due to the reason that each channel connected to the common upper model is able to function independently. As long as the common upper model can learn to adapt different knowledge representations given by different channels, the stochastic model will not suffer from significant test accuracy loss.

An interesting question that readers may ask is: why stochasticity of the model does not impede the second round of training? The fact is that although the gradients with respect to the input are random variables, the gradients with respect to model parameters are not. Since gradients of the inactive channel are just zeros, only weight parameters in the activate channel will be updated in each training step. Therefore, although the weights to be updated alternates, the gradients with respect to model parameters are deterministic at any time.

## 4. Experiments

In this section, we compare the defense effectiveness of regular, SAP and BS models against FGSM (Goodfellow et al., 2015) and CW (Akhtar & Mian, 2018) attacks on MNIST (LeCun, 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009) datasets. FGSM is a typical "one-shot" method which performs only one gradient descent step and CW attack is known to be the strongest attack method so far (Akhtar & Mian, 2018).

**Figure 2**
**We use three images. (a–c): Gradient distributions of CW attack on a SAP model. (d–f): Corresponding gradient distributions on a block switching. Distributions in the same column belong to the same input dimension. Each distribution is sampled for 100 times**



Both of these two datasets contain separated training and testing sets. In our experiments, the training sets are used to train the defending models and the testing sets are used to evaluate classification performance and generate adversarial examples.

This section is organized in the following way: Details about the defending models, including the models' architectures and training methods, are given in Section 4.1. Defending records against FGSM and CW attacks are shown in Section 4.2. Study on how the number of channels in the BS influences its defending effectiveness and classification accuracy is provided in Section 4.3.

## 4.1. Model details

### *4.1.1. Regular models*

We use two standard Convolutional Neural Networks (CNNs) architectures for MNIST and CIFAR-10 datasets, respectively, as they serve as baseline models repeatedly in previous works (Papernot et al., 2016b). Both of these two CNNs have four convolutional layers, two pooling layers, and two fully connected layers, but the kernel size of convolution filters and layer width is different.

Both models are trained using stochastic gradient descent with the mini batch size of 128. Dropout (Srivastava et al., 2014) is used as regularization during training.

### *4.1.2. SAP*

SAP can be applied post hoc to a pre-trained model (Dhillon et al., 2018). Therefore, in order to make the experimental results more comparable, we use the same trained weights for SAP model as of the regular model. SAP is added between the first and second fully connected layers.

### *4.1.3. Block switching*

The switching block in this experiment consists of five channels. During the first round of training, five regular models are trained as described above. Each regular model is split into a lower part, containing all convolutional layers and the first fully connected layer, and a upper part, containing the second fully connected layer. The lower parts of regular model are kept, providing parallel channels of BS while the upper parts are discarded. A upper model, which is the same as the upper part of regular models except that its weights are randomly initialized, is added on top of all channels. The whole BS is then trained on original training set for the second time. We found that the second round of training is much faster than the first round. On MNIST dataset, BS is retrained for 1 epoch and on CIFAR-10 dataset 5 epochs.

The test classification accuracy of all models is summarized in Table 1. The direct comparisons are between the regular model and the SAP model, since they share the same weights, and the average of sub-models is used to construct BS and BS itself. We can conclude that both SAP and BS are excellent in maintaining testing accuracy.

**Table 1**
**Testing accuracy of different models on MNIST and CIFAR-10 datasets**

| Model | Test Acc. on MNIST | Test Acc. on CIFAR |
|---|---|---|
| Regular | 99.04% | 78.31 % |
| SAP | 99.02% | 78.28 % |
| Sub-models Avg. | 99.02% | 78.97% |
| Switching | 98.95% | 78.73% |

## 4.2. Defense against adversarial attacks

We use the fooling ratio, which is the percentage of adversarial examples generated by a attack method that successfully fools a neural network model to predict the target label and to evaluate the defense effectiveness of the target model. The lower the fooling ratio is, the stronger the model is in defending adversarial attacks.

We also record the average $L_2$ norm of the generated adversarial examples from legitimate input images, since it is only fair to compare two attacks at similar distortion levels. For attacks like CW attack that uses a leveraged object function between distortion and misclassification, a large distortion also indicates that it is hard for the attacking algorithm to find an adversarial example in a small region.

### *4.2.1. Experiments on MNIST dataset*

For the sake of reproducibility of our experiments, we report the hyper-parameter settings we use for FGSM and CW attacks. FGSM has one hyper-parameter, the attacking strength $\epsilon$ as shown in equation (1). When using $\epsilon = 0.1$, the $L_2$ norm of adversarial examples roughly matches CW, but the fooling ratio is way too small. Thus, we also test the case when $\epsilon = 0.25$ in order to provide a more meaningful comparison, although the $L_2$ norm is significantly larger. For CW attack, gradient descent is performed for 100 iterations with step size of 0.1. The number of binary searching iterations for $c$ in 2 is set to 10.

We use FGSM and CW attacks to generate adversarial examples targeting the regular model, the SAP model, and BS, respectively. Experimental results are shown in Table 2.

Although the SAP model demonstrates its extra robustness against both FGSM and CW than the regular model, BS is apparently superior and decreases the fooling ratio further.

### *4.2.2. Experiments on CIFAR-10 dataset*

We use $\epsilon = 0.01$ for FGSM in this experiment in order to have adversarial examples with similar distortion level compared to examples generated by CW attack. The hyper-parameter setting for CW attack is the same as above.

Experimental results on CIFAR-10 datasets are shown in Table 3. And BS significantly decreases fooling ratio of FGSM

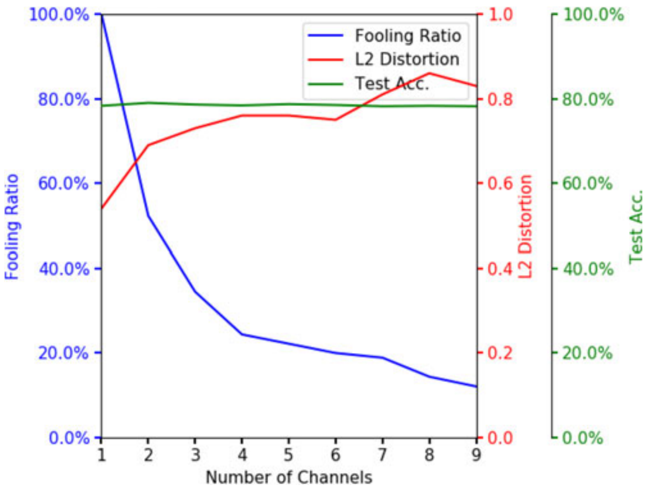
**Table 2**
**Fooling ratio (FR) and distortion of FGSM and CW attacks with different target models on MNIST dataset**

| | Regular | | SAP | | Switching | |
|---|---|---|---|---|---|---|
| Attack | FR | L2 | FR | L2 | FR | L2 |
| FGSM $\epsilon = 0.1$ | 3.9% | 2.73 | 3.7% | 2.73 | 1.6% | 2.73 |
| FGSM $\epsilon = 0.25$ | 34.0% | 6.84 | 32.8% | 6.84 | 20.3% | 6.84 |
| CW | 100.0% | 2.28 | 32.1% | 2.28 | 21.0% | 2.37 |

**Table 3**
**Fooling ratio (FR) and distortion of FGSM and C&W attacks with different target models on CIFAR-10 dataset**

| | Regular | | SAP | | Switching | |
|---|---|---|---|---|---|---|
| Attack | FR | L2 | FR | L2 | FR | L2 |
| FGSM $\epsilon = 0.01$ | 25.0% | 0.55 | 24.8% | 0.55 | 8.1% | 0.55 |
| CW | 100.0% | 0.54 | 93.3% | 0.52 | 22.2% | 0.69 |

**Figure 3**
**Quantifying the impact of channel numbers: we plot defending effectiveness in terms of fooling ratio and $L_2$ distortion, and testing classification accuracy of block switchings with one channel to nine channels**



and CW to 8.1% and 22.2% respectively while the SAP model only shows minor advantages over the regular model.

## 4.3. The effect of channel number

To provide an analysis on how the number of channels in a BS affects its defense effectiveness as well as testing accuracy, we run CW attack on BS models with different number of channels ranging from 1 (which is a regular model) to 9.

In Figure 3, we plot the fooling ratio, distortion, and test accuracy over different channel numbers: in general, the defense becomes stronger with more channels of BS and the fooling ratio is lowest, 12.1%, when using nine channels. The fooling ratio drops rapidly from one channel to four channels while the drop of fooling ratio decelerates after five channels, which indicates the effectiveness provided by switching channels starts to saturate. The increasing distortion of adversarial examples also indicates that BS with more channels is stronger when defending adversarial attacks. The trend of testing accuracy, on the other hand, is almost flat with a very slight descent from 78.31% to 78.17%. This indicates that BS is very effective in defending adversarial attacks with very minor classification accuracy loss.

## 5. Conclusions

In this paper, we investigate BS as a defense against adversarial perturbations. We provide analysis on how the switching scheme defends adversarial attacks as well as empirical results showing that a BS model can decrease the fooling ratio of CW attack from 100% to 12.1% . We also illustrate that stronger defense can be achieved by using more channels at the cost of slight classification accuracy drop.

BS is easy to implement which does not require additional training data nor information about potential adversary. Also, it has no extra computational complexity than a regular model in the inference phase since only one channel is used at a time. In practice, the parallel channels can be stored distributedly with periodical updating, which can provide extra protection to the model that prevents important model information leak.

More importantly, BS demonstrates that it is possible to enhance model variation yet maintain test accuracy at the same time. And we hope this paper can inspire more works toward this direction.

## Acknowledgments

This work is supported by the Air Force Research Laboratory FA8750-18-2-0058, National Science Foundation DMS 1737897, and National Institute of Health R21 EY028381-01. Xiao Wang and Siyue Wang contributed equally.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

## References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *OSDI* (Vol. *16*, pp. 265–283).

Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, *6*, 14410–14430. http://doi.org/10.1109/ACCESS.2018.2807385

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., . . . Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv: 1604.07316*.

Carlini, N., & Wagner, D. (2017a). Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (pp. 3–14). ACM.

Carlini, N., & Wagner, D. (2017b). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 39–57). IEEE.

Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Chen, L., Kounavis, M. E., & Chau, D. H. (2017). Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv: 1705.02900*.

Dhillon, G. S., Azizzadenesheli, K., Bernstein, J. D., Kossaifi, J., Khanna, A., Lipton, Z. C., & Anandkumar, A. (2018). Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1uR4GZRZ

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *2015 ICLR*.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., & McDaniel, P. (2017). On the (statistical) detection of adversarial examples. *arXiv preprint arXiv: 1702.06280*.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., & Dally, W. J. (2016). EIE: Efficient inference engine on compressed deep neural network. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* (pp. 243–254).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., . . . Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia* (pp. 675–678). ACM.

Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).

LeCun, Y. (1998). *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/

LeCun, Y. (2015). *LeNet-5, convolutional neural networks* (p. 20). http://yann.lecun.com/exdb/lenet

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv: 1706.06083*.

Metzen, J. H., Genewein, T., Fischer, V., & Bischoff, B. (2017). On detecting adversarial perturbations. *arXiv preprint arXiv: 1702.04267*.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016a). The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 372–387). IEEE.

Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016b). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 582–597). IEEE.

Parkhi, O. M., Vedaldi, A., & Andrew, Z. (2015). Deep face recognition. In *BMVC* (Vol. *1*, p. 6).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958. http://doi.org/10.5555/2627435.2670313

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv: 1312.6199*.

Wang, S., Wang, X., Ye, S., Zhao, P., & Lin, X. (2018a). Defending DNN adversarial attacks with pruning and logits augmentation. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (pp. 1144–1148). IEEE.

Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., & Lin, X. (2018b). Defensive dropout for hardening deep neural networks under adversarial attacks. In *Proceedings of the International Conference on Computer-Aided Design* (p. 71). ACM.

Wang, X., Wang, S., Chen, P.-Y., Wang, Y., Kulis, B., Lin, X., & Chin, P. (2019). Protecting neural networks with hierarchical random switching: Towards better robustness-accuracy trade-off for stochastic defenses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (pp. 6013–6019). AAAI Press.

Xie, C., Wang, J., Zhang, Z., Ren, Z., & Yuille, A. (2017). Mitigating adversarial effects through randomization. *arXiv preprint arXiv: 1711.01991*.

Zhao, A., Fu, K., Wang, S., Zuo, J., Zhang, Y., Hu, Y., & Wang, H. (2017). Aircraft recognition based on landmark detection in remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, *8*(2017), 1413–1417. http://doi.org/10.1109/LGRS.2017.2715858

**How to Cite:** Wang, X., Wang, S., Chen, P.-Y., Lin, X., & Chin, P. (2022). Block Switching: A Stochastic Approach for Deep Learning Security. *Journal of Computational and Cognitive Engineering* 1(4), 159–164, https://doi.org/10.47852/bonviewJCCE2202320