2023 IEEE/ACM International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest) | 979-8-3503-3812-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/DEEPTEST59248.2023.00010

DeepSHAP Summary for Adversarial Example Detection

1st Yi-Ching Lin
Management Information Systems
National Chengchi University
Taipei, Taiwan
110356019@g.nccu.edu.tw

2nd Fang Yu
Management Information Systems
National Chengchi University
Taipei, Taiwan
yuf@nccu.edu.tw

***Abstract*—The applications of deep learning are widely used across various fields. Additionally, Explainable AI (XAI) helps interpret model predictions, making them more reliable and trustworthy. Taking advantage of XAI techniques, we propose adopting decision logic from explanations to detect adversarial examples. We show that there exists a different interpretation distribution between normal and adversarial examples, as well as diverse decision logic of networks to distinguish them. Specifically, we first use DeepSHAP values to calculate the neuron contribution of the classification model layer-by-layer and show how to utilize this layer-wise explanation to distinguish between normal and adversarial examples. Second, we select critical neurons according to their SHAP values, generate a bitmap to represent the decision logic that interprets distribution of critical neurons, and propose a new approach that uses the decision logic instead of SHAP signature to detect adversarial examples. The preliminary results against the CIFAR-10 dataset demonstrate that the more SHAP layer information is given, the better accuracy can be achieved. Furthermore, using decision logic that concentrates on critical neurons can 1) outperform single or two layer SHAP signature detection approach, and 2) achieve competitive accuracy to all-layer SHAP signature detection with less resource requirement. We also show that the effectiveness of the proposed approach can be transferred to detect untrained attacks.**

***Index Terms*—Adversarial Example, Explainable AI, DeepSHAP, Critical Decision Logic**

I. Introduction

The applications of Deep Learning [1], [2] have been increasing dramatically in different fields from image classification, object detection, language translation to speech recognition and natural language processing. The recent progress on chatbot ChatGPT [3] which can interact with humans to answer questions, generate runnable codes, and write plausible essays has incredibly attracted trillions of active users in a few months. However, despite these advancements, deep learning models still have limitations: one of which is being fooled by adversarial examples [4].

Adversarial examples can be generated by various adversarial attack methods such as FGSM [5], DeepFool [6], and PGD [7]. These methods apply crafted perturbations to normal examples, creating adversarial examples. The post-attack example is undetectable to humans but can easily trick classification models into making incorrect predictions. While various defense methods attempt to correctly classify adversarial examples [7]–[11], most of these defenses are ineffective [12], [13]. Hence, turning to another way, some studies have focused on detecting adversarial examples [4], [13], [14]; Aldahdoo et al. [4] have summarized detection methods into the statistical approach [15], [16], auxiliary model approach [12], [17], network invariant approach [18], [19], feature squeezing approach [20], [21], and denoiser approach [22].

Explainable AI (XAI) can be used to interpret the deep learning model to improve trust and transparency. Many XAI methods have been developed in recent years, such as LRP [23], LIME [24], DeepLIFT [25], and SHAP [26]. The reason for applying XAI in the model is to justify, control, improve, and discover [27]; it can help to make a more trustworthy AI model. It has been applied in various domains, including medical [28]–[30], engineering [31], [32], aerospace [33], transportation [34], manufacturing [35], [36], and security [37], [38].

XAI has also been used for adversarial example detection [39]–[41]. Using SHAP values to detect the adversarial images has been proposed in [39] to detect adversarial images and in [40] to detect the word-level adversarial text attacks. In [41], LIME is used to extract the explanations from trained intrusion detection systems to detect adversarial cyber attacks.

As shown in [39]–[41] that normal and adversarial examples may differ in XAI interpretation, we continue this study line to build a detector based on DeepSHAP summary. We first extend the previous study [39] that adopts SHAP values to detect adversarial examples on a single layer to a combination of multi layers as a layer-wise SHAP signature approach. We show that for the layer-wise SHAP signature approach, more the SHAP layer information is given to interpret input characterization, better the detection accuracy can be achieved.

We further extend this understanding to decision logic. It has been shown [42] that the prediction of normal and adversarial examples may follow different decision logic, which can be represented as a critical neuron map based on neuron significance. We use SHAP values to identify critical neurons (similar to path extraction [42]) and propose a new decision logic approach to distinguish adversarial examples from normal ones via logic diverse.

We also summarize decision logic of multiple examples

979-8-3503-3812-6/23/$31.00 ©2023 IEEE
DOI 10.1109/DeepTest59248.2023.00010

as a decision graph (similar to path abstraction [42]) and observe the difference between the decision graph from normal examples and the graph from adversarial ones, providing evidence to concentrate on critical neurons for effectiveness of adversarial example detection.

We evaluate both layer-wise SHAP signature approach and decision logic approach against adversarial attacks on classification models, showing the effectiveness in detecting adversarial examples through the XAI techniques. In our preliminary results on CIFAR-10 dataset against FGSM [5], DeepFool [6], and PGD [7], we achieve 93.88% average accuracy with all-layer SHAP values and 93.10% with 95th decision logic. It is worth noting that accuracy can be transferred to unknown perturbation-based adversarial attacks including BIM [43], RFGSM [44], PGDL2 [7], APGD [45], and AutoAttack [45]. In the comparison against the CIFAR-10 testing dataset with FGSM, DeepFool, and PGD attacks, our detector outperforms previous detection approaches that have their implementation provided in [46]: KD+BU [47] (59.52%), NSS [48] (71.07%), FS [21] (56.08%), and MagNet [22] (50.01%).

II. Related Work

DeepSHAP [26] connects the Shapley value [49] and Deep Learning Important FeaTures (DeepLIFT) [25] to improve computational performance. The explanation is consistent with human intuition and has been widely used for model interpretation in various fields such as engineering, aerospace, and medical.

1) In Engineering: DeepSHAP has been used to interpret the DRL model in power system emergency control to assist in decision making [32]; to explain the classification model to detect the type of attack on cybersecurity threats data and identify the most influential features [38]; and to perform multiclassification on data in the manufacturing condition monitoring to provide reliable results, explain the importance of each sensor and assist humans in understanding then optimizing the model [36].

2) In Aerospace: Interpret the autonomous navigation path planning to allow users to understand and trust the model more, as well as to improve the model [33]; it has also been used to interpret and visualize the classification of fiber layup defects, which can increase the inspection efficiency and certification of inspection systems [50].

3) In Medical: It has been applied to an automatic detection platform for classifying different types of common fundus diseases and conditions based on images to simultaneously provide the heatmap indicating the important feature via DeepSHAP, allowing clinicians to verify the diagnosis [29]; it can also be used to interpret the task of schizophrenia classification based on fMRI and sMRI [30], and it is widely used in deep learning-based medical image analysis [51]; in addition, it has been implemented on the medical decision support system for non-communicable diseases (NCDs) to select suitable feature from time-series data for improving the system performance, readability and understanding [52].

To summarize the application of DeepSHAP, it can interpret the prediction of a model, let users trust the results, visualize feature contribution, give understandable decision-making and improve model reliability. Furthermore, Generalized DeepSHAP (G-DeepSHAP) modifies the calculation method allowing to formulate of a generalized rescale rule; this makes G-DeepSHAP can apply to linear, tree, and deep models [53].

Finally, using DeepSHAP to detect adversarial examples is not new. It has been applied to image detection to compute the SHAP values from internal layers of DNN to distinguish adversarial examples from normal ones [39]. Additionally, SHAP has been used to detect word-level adversarial attacks [40].

In this study, we extend the SHAP values adversarial detection from layer-wise signature [39] to critical decision graph [42]. Xie et al. [42] extract the decision graph from the LRP interpretation for computing Neuron Path Coverage (NPC), which reveals the internal decision-making process of the input and the effects of the neurons on the prediction via relevance. The authors show that NPC is a more stable coverage matrix for detecting adversarial examples. We show that the decision graph can be utilized to distinguish adversarial examples from normal ones. We comprehensively evaluate layer-wise signature and decision logic approaches and discuss their effectiveness and transferability.

III. Methodology

A. Phase 1: Data Collection

We apply PGD, DeepFool, and FGSM attack methods on the classification model to generate the adversarial examples from the normal dataset. Then, we use only the successful attack on the pair of normal and adversarial examples in our method. A successful attack denotes that the model classifies the normal examples correctly but classifies the adversarial examples wrong. After identifying the target data, we save both the normal and adversarial examples and label them accordingly to distinguish between them. These labeled examples will be our detection dataset.

B. Phase 2: DeepSHAP Computation

Lundberg and Lee [26] propose Shapley Additive exPlanation (SHAP) Values that take advantage of the concept of Shapley value [49] from game theory as a unified measure of feature importance for network models. It describes how to distribute the total contributions to each feature of inputs; in other words, the contribution value of the model's output is attributed to the contribution value of each input feature. The contribution value of each input feature is computed by measuring the change in the prediction when that feature is conditioned.

DeepSHAP [26] connects Shapley value [49] as a unified interpretation matrix with DeepLIFT [25] to compute input and neuron contribution as model prediction explanation. DeepLIFT [25] computes contribution scores by backpropagating the output of a deep learning network from a given input to determine the importance of all features of inputs. It

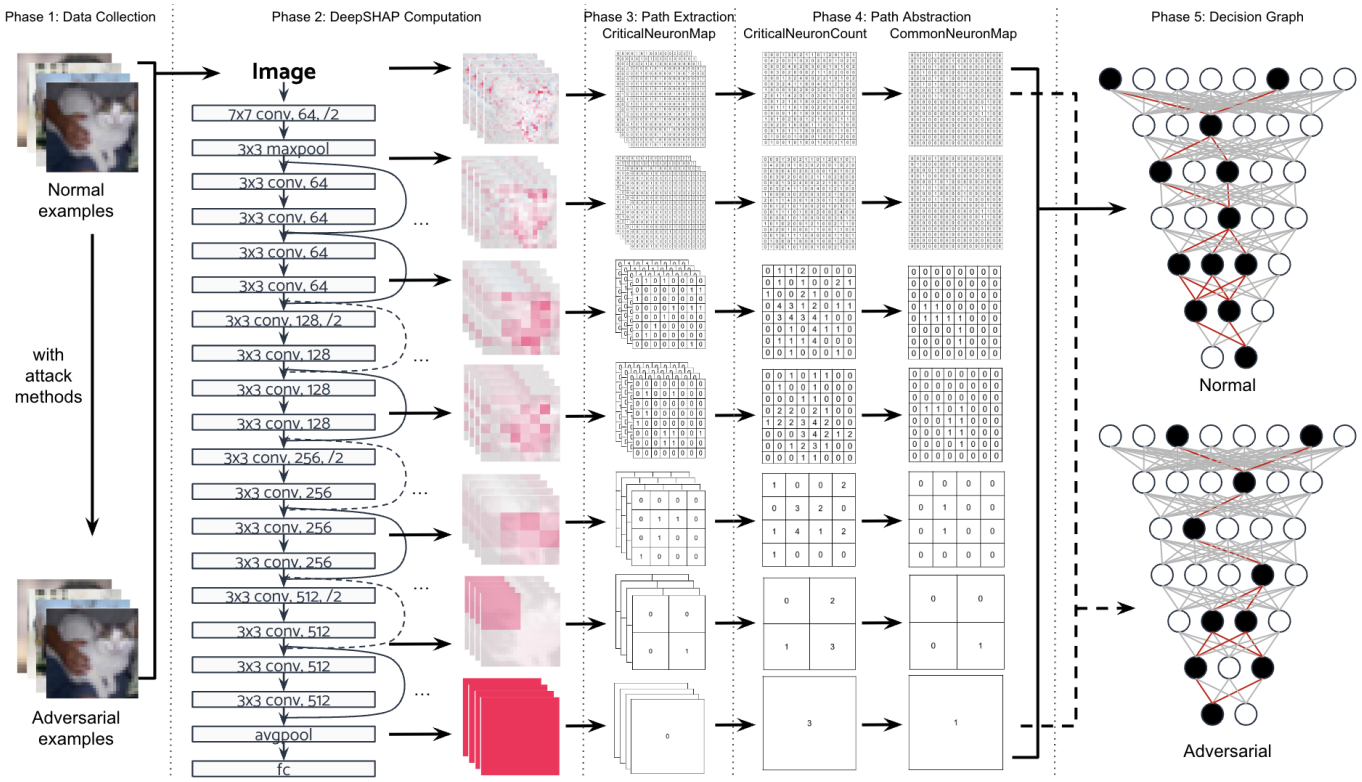
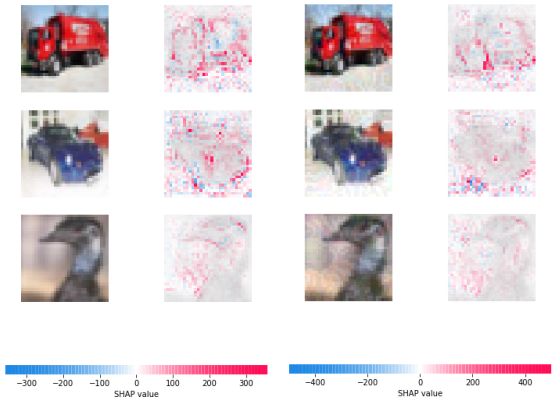


Fig. 1. Analysis Framework

uses the concept of reference, calculating importance based on the difference between the inputs and reference inputs, and the difference between the outputs and reference outputs. The contribution scores are calculated using the chain rule for the multiplier to compute the contribution of each neuron to a given neuron, making the computation more efficient through backpropagation. This results in a fast and efficient approximation of the entire model's SHAP values.

DeepSHAP improves computational performance by computing the SHAP value of smaller components of the network and using them to recursively propagate the DeepLIFT multiplier [25]. We use DeepSHAP as our XAI method by utilizing the DeepExplainer class from its SHAP package [54]. The package takes a background dataset (a few input samples) for integrating features. In a common setting, given input dimension M with output dimension N, the SHAP values' dimension on input is $N \times M$ for explaining each input feature importance to each output class. In our study, we only consider the predicted class as our target output and reduce the dimension to $1 \times M$. Fig. 2 shows three examples on input layer SHAP values against a normal example and its adversarial example (classified in a different category). As one can see, the SHAP color distribution has been changed in all three cases.

The SHAP values in fact are computed layer by layer. That is to say, the contribution on the outputs of each intermediate layer is also computed along the process. Based on this, one can extract the contribution value of any layer in the model.



(a) Normal Examples (b) Adversarial Examples

Fig. 2. The SHAP values on normal and adversarial examples

Fidel et al. [39] have shown that the layer distribution on SHAP values can be used to distinguish adversarial examples from normal ones. In this work, we generalize the approach as the layer-wise SHAP signature approach and further propose a new approach by summarizing SHAP values as decision logic to distinguish adversarial examples from normal ones.

C. Phase 3: αth Path Extraction

According to the XAI method, relevance represents the contribution of a neuron to the final output. The more significant the contribution of the neuron, the more critical it is.

The critical neuron is defined as one that provides a positive contribution to the decision, by sorting the contribution of all neurons in each layer to find the critical ones. Based on the features transformed layer-by-layer to make the prediction, we extract these critical neurons in each layer and then link them up as the Critical Decision Path (CDP) [42], which stands for the internal decision logic of the model.

In our approach, we use the concept of CDP to generate a critical neuron map and extract decision logic. First, we calculate the layer-by-layer contribution via DeepSHAP for each example. Then, we use the αth decision logic approach to generate the critical neuron map. We denote that each bit in the critical neuron map stands for whether the neuron is significant or not, where bit 1 represents the significant neuron and 0 represents the insignificant neuron. For the αth decision logic approach, we use the neurons larger than the αth quantile value of the data. We compute the αth quantile of each layer SHAP value (Q), and assign the bit to the critical neuron map based on whether the SHAP value is less than or greater than Q, indicating insignificance or significance, respectively. For the max decision logic approach, we use the most significant value of the data; we find the maximum value neuron of each layer SHAP value by setting a large α (1-1/N), and then we do the same thing. Through these approaches, we get the more critical neurons at each layer and concatenate them layer-by-layer to generate the decision logic of each example to be our critical neuron map data.

In eq1, we set Q as the αth quantile of the SHAP $value$ of layer l, and i, j, k represent the dimension of $value$.

$$CriticalNeuronMap[i][j][k] = \begin{cases} 1, \text{if } value[i][j][k] > Q \\ 0, \text{otherwise} \end{cases} \tag{1}$$

In the αth decision logic approach, we set the α parameter to calculate the value of the quantile of each layer to find the critical neurons and then generate the decision logic of each example. For instance, by setting the α to 95 in the 95th decision logic approach, we compute the value of the 95th quantile of each layer and select the neurons whose SHAP value is above the value of the 95th quantile, indicating that their SHAP values are in the top 5% of each layer. In the max decision logic approach, we simply find the maximum value of each layer and generate the decision logic. We use the decision logic of the input example as our critical neuron map data, which summarizes all layers of information.

D. Phase 4: β-agree Path Abstraction

We summarize the decision logic of multiple examples using path abstraction proposed in [42]. It assumes that inputs predicted to the same class should have a similar decision logic, and generates the common decision logic for these inputs. In our approach, we utilize this concept to generate the common decision logic by merging the respective layered critical neuron maps of the normal and adversarial examples. First, we extract the critical neuron map in the same way as before. Then, we calculate the weight of each neuron for each class, which represents the ratio of the critical neurons to the total neurons in the critical neuron maps. The following formula calculates the weight of each neuron by merging the critical neuron maps. In eq2, we define N_c as the total number of inputs in the set X_c, where the output prediction is class c, and $CriticalNeuronMap$ stores the bit of each neuron to represent whether the neuron is critical or not for each input x.

$$w_c[i][j][k] = \frac{\sum_{i,j,k,x \in I,J,K,X_c} CriticalNeuronMap[i][j][k]}{N_c} \tag{2}$$

Then, we set the threshold β to select the weights above it to identify critical neurons in the common decision logic. The common critical neurons are concatenated layer-by-layer to get the common decision logic for each class. The formulation in eq3 selects the common critical neurons of the respective class.

$$CommonNeuronMap[i][j][k] = \begin{cases} 1, \text{if } w_c[i][j][k] > \beta \\ 0, \text{otherwise} \end{cases} \tag{3}$$

E. Phase 5: β-agree Decision Graph

Finally, the decision graph comprises the common decision logic of each class. By adjusting the values of parameters α and β, we can generate different β-agree decision graphs for each class, enabling the comparison and analysis of the decision-making process and the different distributions of the common critical neurons between normal and adversarial examples.

Algorithm 1: Training the Detector

input : Normal Dataset N include image i and label l
input : Classification model $classifier$
input : Adversarial method adv of PGD, DeepFool, and FGSM
output: Detection model $detector$

$I = []$ /* an array of images */
$L = []$ /* an array of labels */
for $n \in N$ **do**
  if $classifier(n.i) = n.l$ **then**
    $i' \longleftarrow adv(classifier, n.i)$;
    if $classifier(i') \neq n.l$ **then**
      $I.add(n.i)$;
      $L.add(0)$;
      $I.add(i')$;
      $L.add(1)$;
    end
  end
end
for $idx \in |I|$ **do**
  $x \longleftarrow I[idx]$;
  $y \longleftarrow L[idx]$;
  $S \longleftarrow DeepSHAP(classifier, x)$;
  $B \longleftarrow \alpha DecisionLogic(S, \alpha)$;
  $detector \longleftarrow ModelTraining(B, y, detector)$;
end
return $detector$

IV. Experiment

A. Experiment Setup

1) Adversarial Attack on CIFAR-10: Using CIFAR-10 as the normal dataset, which contains 60,000 images. First, we

Algorithm 2: Adversarial Example Detection

input : The set of images I
input : Classification model $classifier$
input : Detection model $detector$
output: Detection result $result$

for $i \in I$ **do**
    $S \longleftarrow DeepSHAP(classifier, i)$;
    $B \longleftarrow \alpha DecisionLogic(S, \alpha)$;
    $result \longleftarrow detector(B)$;
end
return $result$

train our classification model using the pretrained ResNet18 model from PyTorch on the training dataset. Then, we use this model to generate the adversarial dataset of PGD, DeepFool, and FGSM attacks on the testing dataset. The classification accuracy of the normal dataset is 79.95%, and the classification accuracy of each adversarial dataset is 3.67%, 6.88%, and 6.28%. For the PGD attack, we get 7,628 pairs of successfully attacked images, while for the DeepFool attack, we get 6,989 pairs, and for the FGSM attack, we get 7,367 pairs. In total, we obtain 21,984 pairs of successfully attacked images, which include 21,984 normal images and 21,984 adversarial images, resulting in a total of 43,968 images for our detection dataset.

Then, we use the first 100 images of each attack method as the background and compute the SHAP value through DeepExplainer on the remaining 43,668 images to determine the contribution of each image in the classification for the layer-wise SHAP signature approach. Finally, we generate the critical neuron map by using the αth decision logic approach.

2) Detection model: Our detection model first flattens the data, then uses four linear layers with ReLU activation, whose output features are 256, 128, 16, and 2, to do the binary classification and add the BatchNorm1d and Dropout before the last linear layer. We train the model using the Cross-Entropy Loss function and the Adam optimizer with default parameters.

We observe that data can significantly impact the model's performance. To reduce the impact of randomness, we repeat the process ten times and shuffle the data each time. The detection dataset is split randomly into 8:1:1 for training, validation, and testing. After each training run, we record the test accuracy, and in the end, we average the test accuracy across all ten runs to evaluate the model's performance.

B. Layer-wise SHAP Signature Approach

The layers we selected from the classification model include the input layer, conv1, bn1, maxpool, layer1 to layer4, avgpool, and the output layer, for a total of 10 layers. We use the numbers 0 to 9 as the index of the layers to represent the selected layers. In this approach, we utilize the SHAP values calculated in Phase 2 to evaluate the detection. First, we use the input layer, which denotes the layer_0 SHAP signature approach, and the penultimate layer (avgpool layer), which denotes the layer_8 SHAP signature approach. The reason for using the input layer is that it is more intuitive in showing the impact of the input image on the classification. On the other hand, the penultimate layer is more relevant to the output prediction and has been used for detecting adversarial examples [39]. In our results, as shown in Fig. 3, the average test accuracy of the layer_0 SHAP signature approach is 88.74%, which is higher compared to the accuracy of 86.60% of the layer_8 SHAP signature approach. Moreover, we combine both layers of SHAP values as the layer_0_8 SHAP signature approach to see if providing more information would result in better performance. The average test accuracy is 89.32%, which is slightly higher than the result of the layer_0 SHAP signature approach. According to these results, we use all layers' SHAP values as the layer_0to9 SHAP signature approach and achieve an average test accuracy of 93.88%, which is a significant improvement and gives us more confidence in using the critical neuron map data.

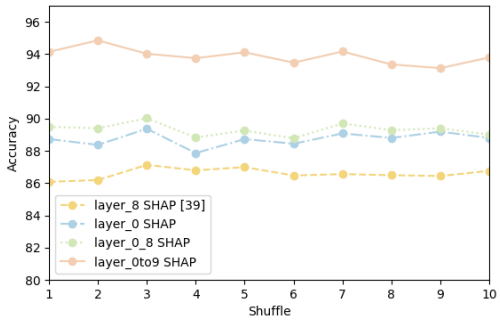


Fig. 3. Layer-wise SHAP Signature Approach detection results

C. αth Decision Logic Approach

In this approach, we utilize the critical neuron map generated by the αth decision logic approach defined in Phase 3 to compare it with the layer_0to9 SHAP signature approach. We evaluate the importance of the alpha setting in the approach by adjusting the α parameter to 5, 25, 50, 75, and 95 to find the optimal setting that achieves high accuracy while reducing unnecessary information. The results demonstrate in Fig. 4 show that the 95th decision logic approach achieves an average test accuracy of 93.10%, which is comparable to the performance of the layer_0to9 SHAP signature approach. Fig. 5 indicates that as the value of alpha decreases in the αth decision logic approach, the number of critical neurons increases, but the average test accuracy decreases. Conversely, when alpha is set too high, such as in the max decision logic approach, the number of critical neurons becomes too small, leading to a decrease in average test accuracy. The max decision logic approach has an average test accuracy of 79.54%, with the fewest neurons, while the 5th decision logic approach has 71.47%, with too many neurons leading to information disorder. Both of these results are even lower than the layer-wise SHAP signature approach. Based on the results, we can conclude that the critical neuron map data, which selects precise critical neurons, can obtain better results than the layer-wise SHAP values data.

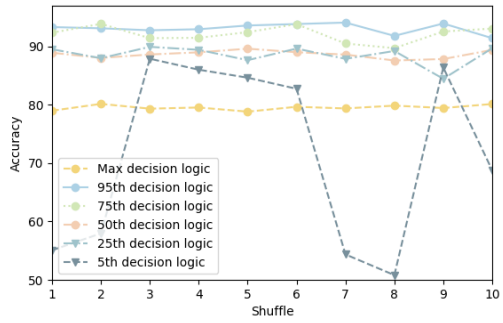


Fig. 4. αth Decision Logic Approach detection results

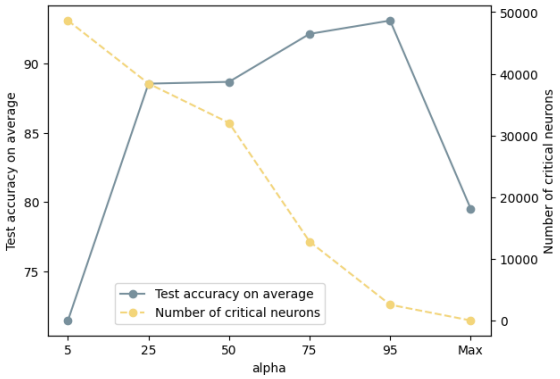


Fig. 5. The detection performance of alpha settings in the αth Decision Logic Approach

D. Comparison on other detection methods

We evaluate the performance of the proposed approach against previous detection approaches, including KD+BU [47], NSS [48], FS [21], and MagNet [22]. The performance of these methods (white box) has been discussed in [4]. We use FGSM, PGD and DeepFool attack methods in the torchattacks library [55] to generate the adversarial examples of the CIFAR-10 testing dataset. We use the implementation provided in [46] to retrain each detector under the original settings against the dataset with these adversarial examples. The accuracy is shown in Fig. 6. The results on detection accuracy, shown as (FGSM, PGD, DeepFool) respectively, are KD+BU (65.46%, 63.11%, 50.00%), NSS (84.18%, 78.70%, 50.34%), FS (59.29%, 58.38%, 50.58%), MagNet (50.05%, 49.98%, 50.01%), while our detectors outperformed them in all cases with layer_0to9 SHAP signature approach (93.99%, 98.28%, 88.40%), and 95th decision logic approach (93.78%, 98.28%, 88.25%).

E. Transferability

To evaluate the transferability of the proposed approach, we apply five other attack methods, including BIM [43], RFGSM [44], PGDL2 [7], APGD [45], and AutoAttack [45] to generate adversarial attacks on the CIFAR-10 testing dataset. We use 10,000 normal images from the CIFAR-10 testing

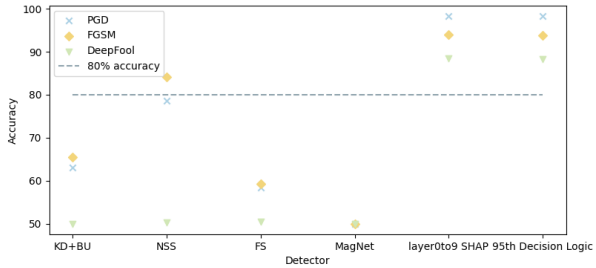


Fig. 6. The comparison on previous adversarial example detectors

dataset, and generate their adversarial examples using the implementation of each method provided in torchattacks [55]. Fig. 7 summarizes the detection results against the 95th decision logic approach. The test accuracy is 92.89% for BIM, 93.07% for RFGSM, 76.76% for PGDL2, 80.93% for APGD, and 85.56% for AutoAttack. In addition, by mixing all five unexplored adversarial examples and shuffling them during testing, we achieve a test accuracy of 85.85% with the 95th decision logic approach and 91.56% with the layer_0to9 SHAP signature approach. The results confirm that our detector is transferable to other adversarial attack methods.

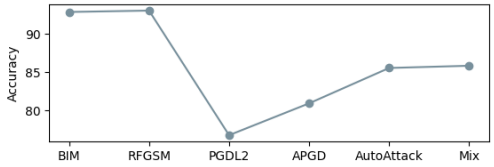


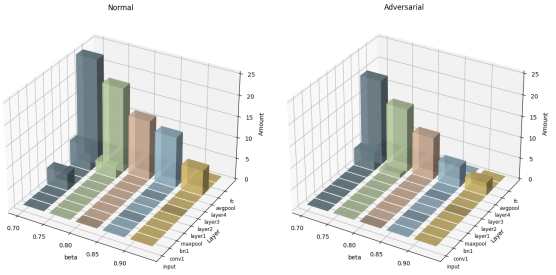
Fig. 7. Accuracy on untrained adversarial attacks

F. Decision Graph Analysis

To understand the differences in decision logic between normal and adversarial examples, we obtain the β-agree decision graph by merging the critical neuron maps. We apply the β-agree decision graph approach to the testing dataset based on the 95th decision logic approach, which has the most stable and best detection accuracy. For each class (normal and adversarial), we set β to pick common critical neurons from the decision logic of each image in the same class. As shown in Fig. 8, we observe that the number of common critical neurons differs, with normal examples having more common critical neurons than adversarial ones. Additionally, Fig. 9 shows that the distribution of common critical neurons also has significant differences, e.g., in 0.7-agree decision graph, we have common critical neurons of normal examples distributed in layer_3 (4), layer_6 (6), layer_7 (25) while adversarial examples in layer_6 (4), layer_7 (20). In summary, the decision graph is dissimilar between normal and adversarial ones, making it useful for detecting adversarial examples. This finding provides evidence to our decision logic approach where we can pay attention to critical neurons instead of all to detect adversarial examples.

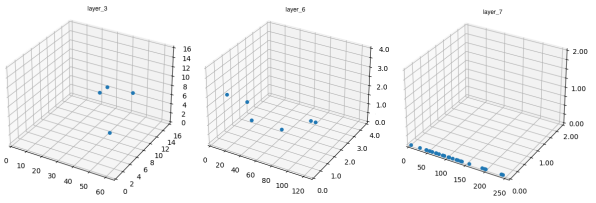
G. Layer Selection Performance

We observe the impact of varying β from 0.9 to 0.7 on the β-agree decision graph approach based on the 95th decision logic approach. In the 0.9-agree decision graph, all of the common critical neurons are located at layer_7, with 6 for normal examples and 3 for adversarial examples (as shown in Fig. 8), suggesting that layer4 has a greater impact than other layers. In the 0.7-agree decision graph, there are more common critical neurons, but still concentrated in some layers, and Fig. 9 shows that their distribution has some differences. Thus, we use layer4 as our data in layer_7 SHAP signature approach. Fig. 10 shows the comparison against other layers. Using the layer_7 SHAP signature approach the average test accuracy is 90.51%, which is better than using the layer_0 SHAP signature approach or layer_8 SHAP signature approach, even with their combination in the layer_0_8 SHAP signature approach. It shows that selecting the layer that has more critical neurons in the decision graph can achieve better performance, and hence providing an effective way of layer selection for layer-wise SHAP signature approaches.

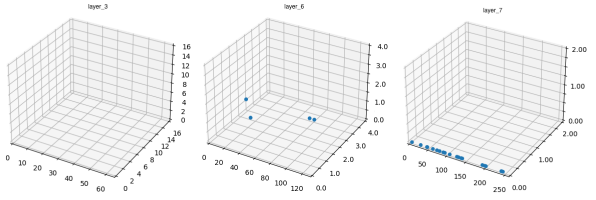


(a) Normal Examples (b) Adversarial Examples

Fig. 8. Number of common critical neurons across different beta settings



(a) Normal Examples (layer_3, layer_6, layer_7)



(b) Adversarial Examples (layer_3, layer_6, layer_7)

Fig. 9. Distribution of common critical neurons in 0.7-agree decision graph

V. Conclusion

We leverage XAI techniques to detect adversarial examples. While using layer-wise SHAP values, a combination of layers can achieve better accuracy than a single layer but requires computation on all layers of SHAP values and using them as input to train a classification model to detect adversarial examples. We propose a new approach based on the β-agree decision graph, where we use the critical neuron map to summarize the impacts of neurons based on their SHAP values. The αth decision logic approach requires fewer input data during the training/testing process since it takes critical neuron distribution as its input instead of the distribution of SHAP values among all neurons in the selected layers. We show that the αth decision logic approach outperforms single/two layer SHAP value detection and achieves competitive accuracy compared to all layer SHAP values detection with much less training and testing computation costs, providing a lightweight detection approach against adversarial examples.

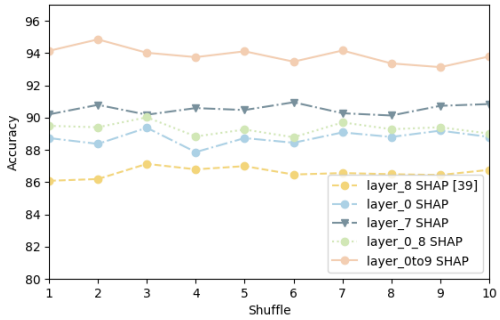


Fig. 10. The layer_7 SHAP signature approach detection results

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[3] OpenAI, "Chatgpt language model," https://openai.com, 2023.

[4] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges, "Adversarial example detection for dnn models: A review and experimental comparison," *Artificial Intelligence Review*, vol. 55, no. 6, pp. 4403–4462, 2022.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[6] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[8] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv preprint arXiv:1412.5068*, 2014.

[9] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[10] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 25–32.

[11] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 4480–4488.

[12] T. Pang, C. Du, Y. Dong, and J. Zhu, "Towards robust detection of adversarial examples," *Advances in neural information processing systems*, vol. 31, 2018.

[13] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.

[14] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

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

[16] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5764–5772.

[17] Z. Gong, W. Wang, and W.-S. Ku, "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960*, 2017.

[18] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[19] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 446–454.

[20] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 72–85, 2018.

[21] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[22] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.

[23] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[25] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[27] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[28] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," *Journal of Imaging*, vol. 6, no. 6, p. 52, 2020.

[29] L.-P. Cen, J. Ji, J.-W. Lin, S.-T. Ju, H.-J. Lin, T.-P. Li, Y. Wang, J.-F. Yang, Y.-F. Liu, S. Tan *et al.*, "Automatic detection of 39 fundus diseases and conditions in retinal photographs using deep neural networks," *Nature communications*, vol. 12, no. 1, pp. 1–13, 2021.

[30] J. Reiter, "Developing an interpretable schizophrenia deep learning classifier on fmri and smri using a patient-centered deepshap," in *in 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)(Montreal: NeurIPS)*, 2020, pp. 1–11.

[31] S. Mangalathu, S.-H. Hwang, and J.-S. Jeon, "Failure mode and effects analysis of rc members based on machine-learning-based shapley additive explanations (shap) approach," *Engineering Structures*, vol. 219, p. 110927, 2020.

[32] K. Zhang, J. Zhang, P.-D. Xu, T. Gao, and D. W. Gao, "Explainable ai in deep reinforcement learning models for power system emergency control," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 2, pp. 419–427, 2021.

[33] L. He, N. Aouf, and B. Song, "Explainable deep reinforcement learning for uav autonomous path planning," *Aerospace science and technology*, vol. 118, p. 107052, 2021.

[34] A. B. Parsa, A. Movahedi, H. Taghipour, S. Derrible, and A. K. Mohammadian, "Toward safer highways, application of xgboost and shap for real-time accident detection and feature analysis," *Accident Analysis & Prevention*, vol. 136, p. 105405, 2020.

[35] H. Wu, A. Huang, and J. W. Sutherland, "Layer-wise relevance propagation for interpreting lstm-rnn decisions in predictive maintenance," *The International Journal of Advanced Manufacturing Technology*, vol. 118, no. 3, pp. 963–978, 2022.

[36] A. T. Keleko, B. Kamsu-Foguem, R. H. Ngouna, and A. Tongne, "Health condition monitoring of a complex hydraulic system using deep neural network and deepshap explainable xai," *Advances in Engineering Software*, vol. 175, p. 103339, 2023.

[37] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck, "Evaluating explanation methods for deep learning in security," in *2020 IEEE european symposium on security and privacy (EuroS&P)*. IEEE, 2020, pp. 158–174.

[38] R. Alenezi and S. A. Ludwig, "Explainability of cybersecurity threats data using shap," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2021, pp. 01–10.

[39] G. Fidel, R. Bitton, and A. Shabtai, "When explainability meets adversarial learning: Detecting adversarial examples using shap signatures," in *2020 international joint conference on neural networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[40] E. Mosca, L. Huber, M. A. Kühn, and G. Groh, "Detecting word-level adversarial text attacks via shapley additive explanations," in *Proceedings of the 7th Workshop on Representation Learning for NLP*, 2022, pp. 156–166.

[41] E. Tcydenova, T. W. Kim, C. Lee, and J. H. Park, "Detection of adversarial attacks in ai-based intrusion detection systems using explainable ai," *HUMAN-CENTRIC COMPUTING AND INFORMATION SCIENCES*, vol. 11, 2021.

[42] X. Xie, T. Li, J. Wang, L. Ma, Q. Guo, F. Juefei-Xu, and Y. Liu, "Npc: N euron p ath c overage via characterizing decision logic of deep neural networks," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 3, pp. 1–27, 2022.

[43] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[44] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[45] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*. PMLR, 2020, pp. 2206–2216.

[46] A. Aldahdooh, https://github.com/aldahdooh/detectors_review, 2022.

[47] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[48] A. Kherchouche, S. A. Fezza, W. Hamidouche, and O. Déforges, "Detection of adversarial examples in deep neural networks with natural scene statistics," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[49] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.

[50] S. Meister, M. Wermes, J. Stüve, and R. M. Groves, "Investigations on explainable artificial intelligence methods for the deep learning classification of fibre layup defect in the automated composite manufacturing," *Composites Part B: Engineering*, vol. 224, p. 109160, 2021.

[51] B. H. Van der Velden, H. J. Kuijf, K. G. Gilhuijs, and M. A. Viergever, "Explainable artificial intelligence (xai) in deep learning-based medical image analysis," *Medical Image Analysis*, p. 102470, 2022.

[52] K. Davagdorj, J.-W. Bae, V.-H. Pham, N. Theera-Umpon, and K. H. Ryu, "Explainable artificial intelligence based framework for non-communicable diseases prediction," *IEEE Access*, vol. 9, pp. 123 672–123 688, 2021.

[53] H. Chen, S. M. Lundberg, and S.-I. Lee, "Explaining a series of models by propagating shapley values," *Nature communications*, vol. 13, no. 1, p. 4512, 2022.

[54] S. Lundberg, "shap," https://github.com/slundberg/shap, 2022.

[55] H. Kim, "Torchattacks: A pytorch repository for adversarial attacks," *arXiv preprint arXiv:2010.01950*, 2020.