Efficient Defense Against First Order Adversarial Attacks on Convolutional Neural Networks

Subah Karnine[1†], Sadia Afrose[1†], Hafiz Imtiaz[1*]

[1*]Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, 1205, Bangladesh.

*Corresponding author(s). E-mail(s): hafizimtiaz@eee.buet.ac.bd;
Contributing authors: 1706174@eee.buet.ac.bd; 1706161@eee.buet.ac.bd;
†These authors contributed equally to this work.

Abstract

Machine learning models, especially neural networks, are vulnerable to adversarial attacks, where inputs are purposefully altered to induce incorrect predictions. These adversarial inputs closely resemble benign (unaltered) inputs, making them difficult to detect, and posing significant security risks in critical applications, such as autonomous vehicles, medical diagnostics, and financial transactions. Several methods exist to improve neural network model's performance against these adversarial attacks, which typically modify the network architecture or training procedure. However, often times these *adversarial training* techniques provide robustness against specific attack types and/or require substantial computational resources, making them impractical for real-world applications with limited resources. In this work, we propose a computationally-efficient adversarial fine-tuning approach to enhance the robustness of Convolutional Neural Networks (CNNs) against adversarial attacks and attain the same level of performance as the conventional adversarial training. More specifically, we propose to identify specific parts of the neural network model, in this case convolutional filters, that are more vulnerable to adversarial attacks. Our analysis reveals that only a small portion of these vulnerable components accounts for a majority of the model's errors caused by adversarial attacks. As such, we propose to selectively fine-tune these vulnerable components using different adversarial training methods to develop an effective and resource-efficient approach to improve model robustness. We empirically validate our proposed approach for one real datasets and demonstrate that our approach can achieve similar performance as the more resource-intensive conventional adversarial training method. We note that enhancing model robustness against adversarial attacks is crucial for ensuring

the reliability and safety of machine learning applications, ultimately promoting societal security and well being.

Keywords: Adversarial attacks, machine learning model security, convolutional neural networks, fast gradient sign method (FGSM), projected gradient descent (PGD)

1 Introduction

Deep neural networks have proven to be highly effective in solving complex machine learning tasks, such as image recognition [1, 2], speech recognition [3], natural language processing [4], and even computer games [5, 6]. These networks have achieved remarkable success in recognizing images with accuracy levels close to (and sometimes exceeding) that of humans. However, researchers have recently discovered that these networks are prone to adversarial attacks. More specifically, these attacks intentionally perturb samples to *simulate* worst-case scenarios, leading the network to output incorrect results with high confidence levels.

Adversarial examples were first discovered in the image classification domain by Szegedy et al. [7]. Their research showed that it is possible to transform the classification output corresponding to an image by making minimal alterations to it. This means that given an input $\mathbf{x}$ and any target classification t, it is possible to discover a new input $\tilde{\mathbf{x}}$ that is very similar to the original input $\mathbf{x}$, but classified as the target $t' \neq t$. The quantity of change required is often so small that it is difficult for humans to detect, making it a significant challenge to use neural networks in security-sensitive areas. In other words, adversarial examples pose a significant concern, since they can limit the domains in which neural networks can be safely used. For example, using neural networks in self-driving vehicles can be risky because an attacker could exploit adversarial examples to cause the car to take actions that it is not supposed to take [8]. As a result, constructing robust neural networks, that are resistant to such attacks, is a top priority for researchers in the field.

As such, robust defense mechanisms against such attacks on modern machine learning models have been the topic of extensive research. Gu and Rigazio [9] and Chalupka et al. [10] have started the journey towards adversarial resistant models. Since then, numerous techniques have been proposed to enhance the robustness of neural networks against adversarial threats. These include approaches similar to adversarial training – where the model is trained on adversarial examples, defensive distillation methods – which aim to smooth the model's decision boundaries [11], and certified defenses – which provide theoretical guarantees against specific attack types. However, existing solutions often have practical limitations, such as being tailored to specific attacks, requiring substantial computational resources, and offering incomplete protection against the wide range of possible adversarial attacks.

Our Contributions. In this work, we propose a novel and computationally efficient method for ensuring adversarial robustness of convolutional neural networks (CNNs). We achieve this by proposing a *selective adversarial training* approach. More specifically, our proposed approach identifies the model components that are most vulnerable

to adversarial perturbations, and then ensures the model's robustness against adversarial attacks by selectively fine-tuning those components. To this end, we identify the convolutional filters of a CNN model that are highly susceptible to adversarial attacks. We show that our proposed *adversarial fine-tuning* of these filters enables the resulting model to maintain a high accuracy on benign inputs, while exhibiting similar, if not better) resilience against adversarial inputs. Our contributions are summarized below:

- We show that the effect of first order adversarial attacks on a CNN model is neither uniform nor random. In fact, certain parts of the model are more susceptible to an attack, regardless of data class. We empirically show this by identifying the filters in the convolutional layers across different datasets.
- Since certain specific components of the model are more vulnerable to adversarial attacks, we argue that focusing on those components are crucial for ensuring model robustness against those attacks. To this end, we propose to split the model into trainable and non-trainable sections. We empirically demonstrate that performing adversarial fine-tuning of the vulnerable components in this way provides a model that performs just as well as existing adversarial training methods. Additionally, this enables a much simpler and computationally light adversarial training.
- We increase the trainable part of the model and show that results do not significantly improve with it, re-enforcing our claim that the whole model does not need to be trained for appropriate adversarial security.

Notations. For vector, matrix, and scalar, we used bold lower case letter ($\mathbf{v}$), bold capital letter ($\mathbf{V}$), and unbolded letter (M), respectively. We used the symbol $\mathbf{v}_n$ for the n-th column of the matrix $\mathbf{V}$; and v_{ij} denotes the (i, j)-th entry of matrix $\mathbf{V}$. We sometimes denote the set $\{1, 2, \ldots, N\}$ as $[N]$. Inequality $\mathbf{V} \geq 0$ apply entry-wise. We denoted $\mathcal{L}_2$ norm (Euclidean norm) with $\|.\|_2$, the $\mathcal{L}_\infty$ norm with the $\|.\|_\infty$, and the Frobenius norm with $\|.\|_F$.

2 Background and Related Works

As mentioned before, neural networks that are commonly used in practice, such as computer vision and speech recognition applications, are susceptible to adversarial attacks that manipulate the model into predicting the wrong output. As such, protection against such attacks has garnered particular research interest [12–15]. In computer vision, adversarial attacks are of particular interest as very small and undetectable perturbations can be added to a benign image in order to fool a model with high probability [7, 16–18]. To counter this, several works proposed defensive techniques against adversarial attacks. Some examples include feature squeezing [19], defensive distillation [11, 20, 21], and other detection methods [22]. All of these methods, however, are computationally expensive.

Individual input features' precision can frequently be restricted in a variety of circumstances. For instance, since digital photographs are frequently only recorded with 8 bits per pixel, any data that descends below 1/255 of the dynamic range is lost [16]. This restriction may have significant effects on the precision and resilience

of machine learning models, particularly when it comes to adversarial examples that can take advantage of the vulnerabilities presented by such restrictions. Therefore, when constructing and assessing machine learning models for various applications, researchers must consider these limits. If the precision of the features is limited, it is not reasonable for a classifier to respond differently to an input $\mathbf{x}$ than to an adversarial input $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}$, as long as every element of the perturbation $\boldsymbol{\eta}$ is smaller than the precision of the features. In other words, if the changes made to the input features are smaller than the resolution of the input, the classifier should not treat the adversarial input any differently than the original input [16].

Ideally, for datasets with well-separated classes, it is expected for the classifier model to assign the same class to both the original input $\mathbf{x}$ and the adversarial input $\tilde{\mathbf{x}}$, as long as $\|\boldsymbol{\eta}\|_\infty \leq \epsilon$, where ϵ is small enough to be discarded by the sensor or data storage device, which considers it to be noise or measurement error and not impact the classification decision [16]. If $\mathbf{w}$ represents the weights of the linear model then the dot product between $\mathbf{w}$ and adversarial example $\tilde{\mathbf{x}}$ is $\mathbf{w}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} + \mathbf{w}^\top \boldsymbol{\eta}$. After going through the model, the previously imperceptible noise marker $\boldsymbol{\eta}$ causes the activation to grow by $\mathbf{w}^\top \boldsymbol{\eta}$. This increased activation can be maximized by assigning $\boldsymbol{\eta} = \text{sign}(\mathbf{w})$, while making sure the constraint on $\boldsymbol{\eta}$ still holds. For a weight vector containing elements with an average magnitude of m and having n dimension, this perturbation results in an activation increase of ϵmn. While the norm of the perturbation $\boldsymbol{\eta}$ does not grow with the dimensionality of the problem, the change in activation caused by the perturbation ϵ can grow linearly with the dimensionality. As a result, in high-dimensional problems, it is possible to make many infinitesimal changes to the input that add up to one large change in the output [16, 23].

Fast Gradient Sign Method (FGSM). For a neural network model, let $\boldsymbol{\theta}$ be the model parameters, y be the target associated with input sample $\mathbf{x}$, and $J(\boldsymbol{\theta}, \mathbf{x}, y)$ be the cost function. Goodfellow et al. [16] showed that the cost function can be linearized around the current $\boldsymbol{\theta}$ value, obtaining an optimal max-norm constrained perturbation of $\boldsymbol{\eta} = \epsilon \text{sign}(\Delta * J(\boldsymbol{\theta}, \mathbf{x}, y))$. Goodfellow referred this approach as the Fast Gradient Sign Method (FGSM) of generating adversarial perturbations, and thereby, adversarial samples. Primarily designed to be fast, instead of a close adversarial estimate, FGSM is optimized for $\mathcal{L}_\infty$ distance metrics to ensure the amount of perturbation remains within the fixed bound of ϵ. Here the $\mathcal{L}_\infty$ norm measures the largest absolute difference between an every element of an input and it's perturbed counterpart. The adversarial example can be calculated as $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \text{sign}(\Delta * J(\boldsymbol{\theta}, \mathbf{x}, y))$, where ϵ is often chosen to be small enough to be imperceptible to humans. FGSM is a simple one-step algorithm for maximizing the inner part of the saddle point formulation of the loss function [23].

Projected Gradient Descent (PGD). Madry et al. [23] proposed a multi step variant of FGSM. The Projected Gradient Descent (PGD) scheme for generating adversarial examples is a more powerful iterative attack that performs multiple gradient descent steps to find the perturbation with maximum loss while ensuring that the adversarial input stays within the constraint typically imposed by the $\mathcal{L}_\infty$ norm. It is shown to produce more effective attacks compared to FGSM. Instead of taking a single step of size ϵ in the direction of the gradient-sign, multiple smaller steps are taken. More specifically, at t-th iteration, the adversarial sample is given

Table 1 Details of the CNN Model

Layer	Output Shape	Param #
Conv2D	$28 \times 28 \times 32$	832
MaxPooling2D	$14 \times 14 \times 32$	0
Dropout	$14 \times 14 \times 32$	0
Conv2D	$14 \times 14 \times 64$	18,496
MaxPooling2D	$7 \times 7 \times 64$	0
Dropout	$7 \times 7 \times 64$	0
Flatten	3136	0
Dense	32	100,384
Dense	10	330

by $\mathbf{x}^{t+1} = \Pi_{\mathbf{x}+\mathbf{S}}(\mathbf{x}^t + \alpha \text{sign}(\Delta_x J(\boldsymbol{\theta}, \mathbf{x}, y)))$. Here, $\boldsymbol{S}$ is the set of allowed perturbations chosen such that it maintains perceptual similarities between an input and its perturbed counterpart, and α is the step size.

3 Proposed Approach Against Adversarial Attacks

Adversarial training is crucial for any neural network model deployed in applications, where security is a priority. Nevertheless, computation cost, model complexity and memory usage, and inference times are crucial factors that need to be considered during adversarial training. Moreover, the adversarial training of the model must be robust enough to minimize the effect of different adversarial attacks. Our work is motivated by such need for efficient and robust adversarial training scheme. More specifically, we focus on a CNN trained on the MNIST handwritten digits dataset [24]. To that end, we are interested in investigating:

- Does an adversarial attack affect different parts of a network equally?
- If not, how can we take advantage of this while performing adversarial training of the network?

We utilized the Cleverhans [25] open source library to generate the adversarial examples, and demonstrate the vulnerability of neural network models. The library offers a selection of attacks and countermeasures for testing how susceptible machine learning models are to adversarial examples.

3.1 Model

As mentioned before, we consider a CNN for image classification and trained the model on the MNIST dataset. After training, the CNN captures the spatial relationships present in an image fairly accurately. Because fewer parameters are needed and weights can be reused, this architecture yields superior results than fully-connected neural networks. We present the details of the model we used in Table 1. Except for the output layer, we have used the ReLU activation. The structure of the base model is shown in Fig. 1. This model has a total of 120,042 trainable parameters. Note that, the MNIST dataset's training partition consists of 60000 gray scale images of size 28×28 pixels, and the test partition consists of 10000 gray scale images of the same size.

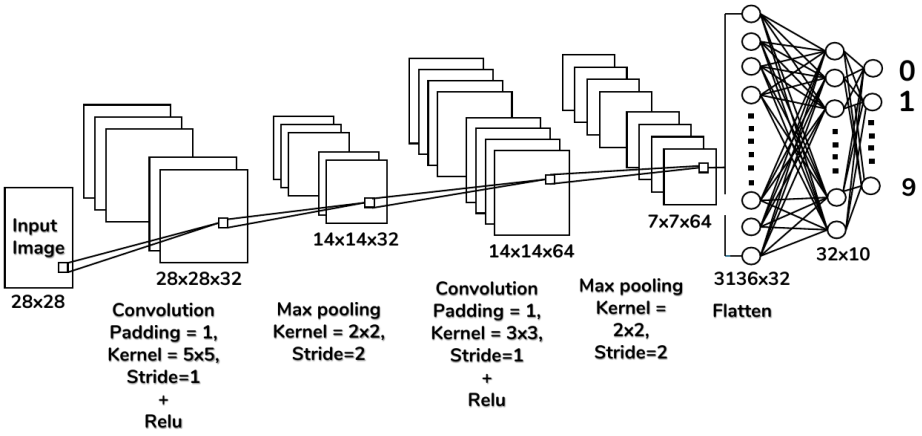

Fig. 1 CNN Model under consideration

The model is trained with RMSprop algorithm and 0.001 learning rate. Early stopping is employed to prevent over-fitting by monitoring the accuracy of the set over a patient of 2 epochs. A dropout layer after every convolutional layer was used further prevent overfitting. The accuracy and loss plots of the training and validation set, as shown in Figure 2, demonstrate that the model has managed to properly learn without overfitting to the dataset. Evaluating it against the test set gives us an accuracy of 99.33%.

3.2 Generating Adversarial Examples

To test the vulnerability of the model, adversarial examples are generated using both the FGSM and PGD approaches. Given a valid input data $\mathbf{x}$ and a target classification, $t = C(\mathbf{x})$, it is possible to find a similar input $\tilde{\mathbf{x}}$ such that $C(\tilde{\mathbf{x}}) = t$. Here $C(\mathbf{x}) = \arg\max F(\mathbf{x})$ is the classifier function, and $F(\mathbf{x})$ is the neural network loss. Additionally, $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are close with respect to some distance metric. The adversarial example $\tilde{\mathbf{x}}$ with this property is known as a targeted adversarial example [23]. A less powerful attack, or un-targeted attack, classifying $\mathbf{x}$ as a given target class searches only for an perturbed input $\tilde{\mathbf{x}}$ so that $C(\tilde{\mathbf{x}}) \neq C(\mathbf{x})$, and that $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are close spatially. Carlini and Wagner [26] considered three different approaches to choosing the target class in a targeted attack:

- Average Case – target class selected uniformly at random among the incorrect labels
- Best Case – targeting the class least difficult to attack
- Worst Case – targeting the class most difficult to attack

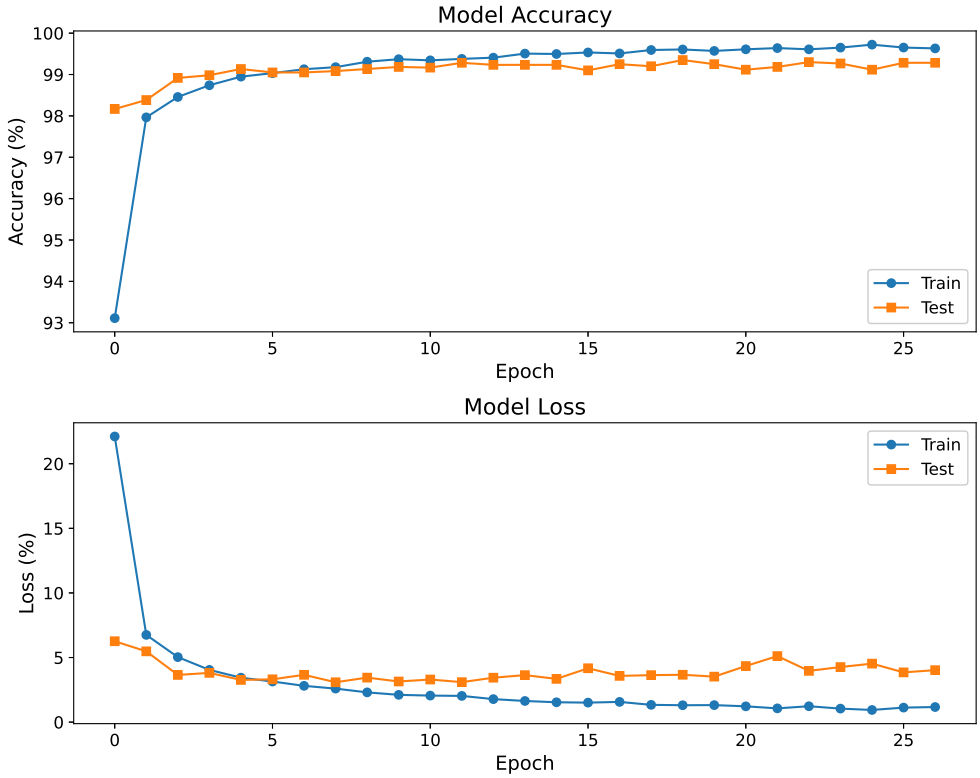

Fig. 2 Accuracy and loss plots of the base model.

3.3 Model Behaviour Under FGSM and PGD Attacks

As mentioned before, a dataset of adversarial examples using FGSM and PGD approaches is generated for the corresponding MNIST dataset with different ϵ values. We employ a white box targeted FGSM and PGD attack on the base model. Intuitively, higher ϵ values result in higher attack success rates, i.e., the model performance should have higher error rates. The model was evaluated using a test set of 10,000 adversarial examples for different values of ϵ.

For the FGSM attack, each image from a class is used to produce adversarial examples targeting the remaining classes. We observed that the FGSM targeted attack was successful 11.68% of the time. Table 2 and 3 show whether an adversarial attack where the perturbation is constructed to output a predetermined target class is successful with it's intention. In Table 2, we show that targeted FGSM attacks on our model are not very successful, even though the adversarial accuracy of the model for the same amount of perturbation stands at 21.26%. On the other hand, the targeted PGD attacks are more successful, as PGD is a stronger attack. We show the details of the targeted PGD attacks on our model in Table 3. The base model can correctly identify a PGD adversarial attack 0.58% of time.

Table 2 Targeted FGSM Attack

	Target Class									
Source Class	0	1	2	3	4	5	6	7	8	9
0	-	x	x	x	x	√	√	x	x	x
1	x	-	x	x	√	x	x	x	x	x
2	x	√	-	x	x	x	x	x	x	x
3	x	x	x	-	x	√	x	x	√	x
4	x	x	√	x	-	x	x	√	x	x
5	x	x	x	x	x	-	√	x	x	x
6	x	x	x	x	x	√	-	x	x	x
7	x	x	√	x	x	x	x	-	x	x
8	x	x	√	x	x	x	x	x	-	x
9	x	x	x	x	√	x	x	x	x	-

Table 3 Targeted PGD Attack

	Target Class									
Source Class	0	1	2	3	4	5	6	7	8	9
0	-	x	x	x	x	√	√	x	x	√
1	√	-	√	√	√	√	√	√	√	√
2	x	√	-	x	x	x	x	x	x	x
3	x	x	√	-	x	√	x	x	√	√
4	x	x	√	x	-	x	x	√	√	√
5	x	x	x	x	x	-	√	x	x	x
6	x	x	x	x	x	√	-	x	x	x
7	√	x	√	√	x	√	x	-	√	√
8	x	x	√	√	x	x	x	x	-	x
9	x	x	x	x	√	x	x	x	x	-

To further test the effect of the magnitude of this perturbation of the model with a shallow and a deeper networks, we built two more models trained on MNIST. The shallow model has two convolution layers with number of filters 16 and 32, respectively. The deeper network has one added convolutional layer, making it three convolutional layers with 32, 64 and 128 filters. The other parameters of the networks are kept consistent with that of our base model. The results are summarized in Table 4 and 5. As expected, all the models perform worse with higher perturbation attack. In Figure 3, we show the model performance on adversarial examples for different ϵ values. It is evident from this figure that PGD is the stronger attack approach of the two.

Table 4 Adversarial Accuracy before Adversarial Training VS Magnitude of Perturbation of the FGSM Attack

Model Type	**Value of ϵ**			
	0.1	0.2	0.3	0.5
Shallow Model	62.10%	21.10%	10.61%	6.71%
Base Model	70.36%	34.01%	21.26%	16.75%
Deeper Model	74.15%	25.88%	11.64%	6.98%

Table 5 Adversarial Accuracy before Adversarial Training VS Magnitude of Perturbation of the PGD Attack

Model Type	**Value of ϵ**			
	0.1	0.2	0.3	0.5
Shallow Model	60.00%	1.44%	0.89%	0.89%
Base Model	64.34%	0.82%	0.58%	0.58%
Deeper Model	67.95%	5.51%	1.20%	1.14%

3.4 Filter Identification

We argue that the adversarial examples are generated by exploiting particular filters in the convolution layers. In this section, we investigate this hypothesis and identify the filters more susceptible to be exploited during the FGSM or PGD attacks. We extract the output features of the convolutional layers of the model, and observe their individual effects on different inputs. As such, the difference in output of the convolutional layer between a benign input image and the corresponding adversarial input image provides the relative affect of an adversarial attack on the layer.

From the test set of the MNIST dataset, we selected 100 images from each class, and generated adversarial images corresponding to these benign images with a white box attack targeting the remaining nine classes with $\epsilon = 0.3$. As a result, 900 targeted adversarial examples from the 100 images of each class (9 adversarial examples for a single image) are generated. We extracted the outputs of the convolutional layers, and calculated the difference in the output between a benign image and its adversarial counterpart. As hypothesized, we observe that some filters are effected more than others. We selected the top 10% of the affected filters for each of the 900 benign-adversarial pairs. Frequency of the filters appearing in the top 10% of the search for the first convolutional layer for input class 0 on MNIST is plotted on a histogram shown in Figure 4. As is clear from the results, some filters cause more difference in convolutional layer output. We repeat this investigation for all the other nine classes, and observed that the top 10% filters remain mostly the same. That is, the same filters in a convolutional layer are affected the most in an adversarial attack regardless of the input class as can be seen from Figure 5. The second convolutional layer also exhibited a similar behaviour. The identified filters for the second layer are shown in Figure 6. Therefore, employing defensive measures on these particular filters should provide strong defense against adversarial attacks, while not sacrificing too much of

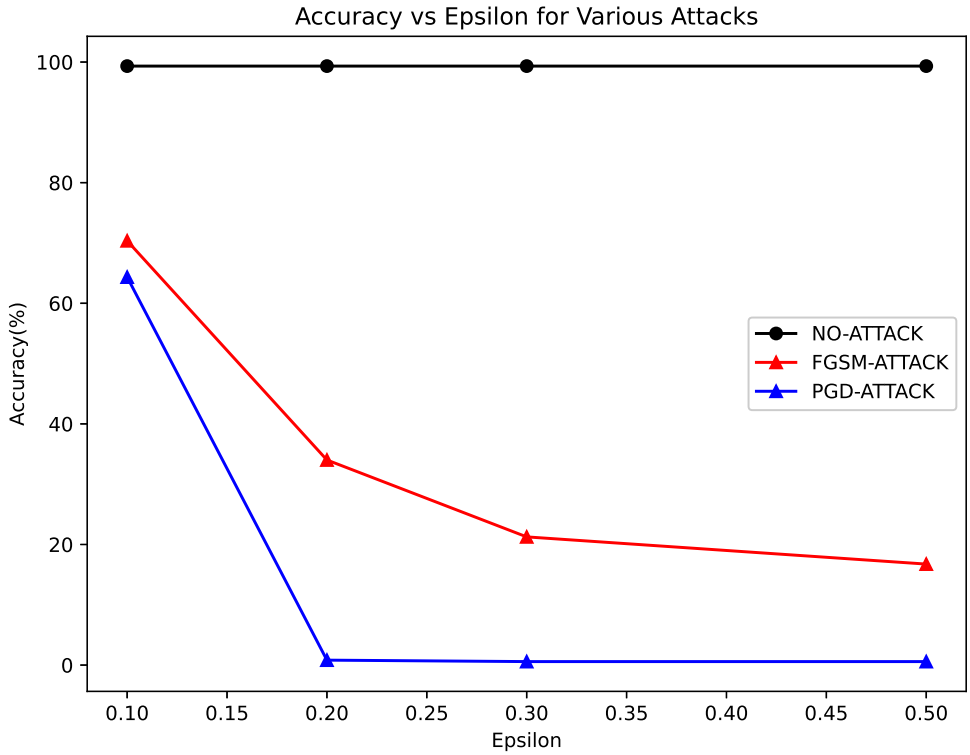

Fig. 3 Adversarial Accuracy Vs Magnitude of Perturbation

the model performance. Additionally, this approach possess the added advantage of acquiring adversarial defense, while re-training much less parameters.

3.5 Model Splitting and Adversarial Training

The existing approach for defense against adversarial attacks is to re-train the model with adversarial samples accompanied by correct class labels. Training the model with a certain form of attack gives it the necessary defense for that attack and weaker (to some extent) attacks [7, 23], but the training also makes the model lose some of it's initial performance capabilities. We hypothesize that re-training the parts of the model that are exploited the most during adversarial attacks should provide a balanced outcome on both requirements. We term the most susceptible filters, as described in the previous section, as *dominant filters*. We intend to re-train or fine-tune these filters, while keeping other parts of the model frozen, for defense against adversarial attacks. This can be accomplished with a split model, as shown in Figure 7.

Note that, we are proposing an adversarial fine-tuning. That is, the weights of the trained base model are transferred to this new split model, and everything except for the dominant filters are kept frozen during the fine-tuning. To this end, each convolutional layer is split into two parallel layers – one with the dominant filters

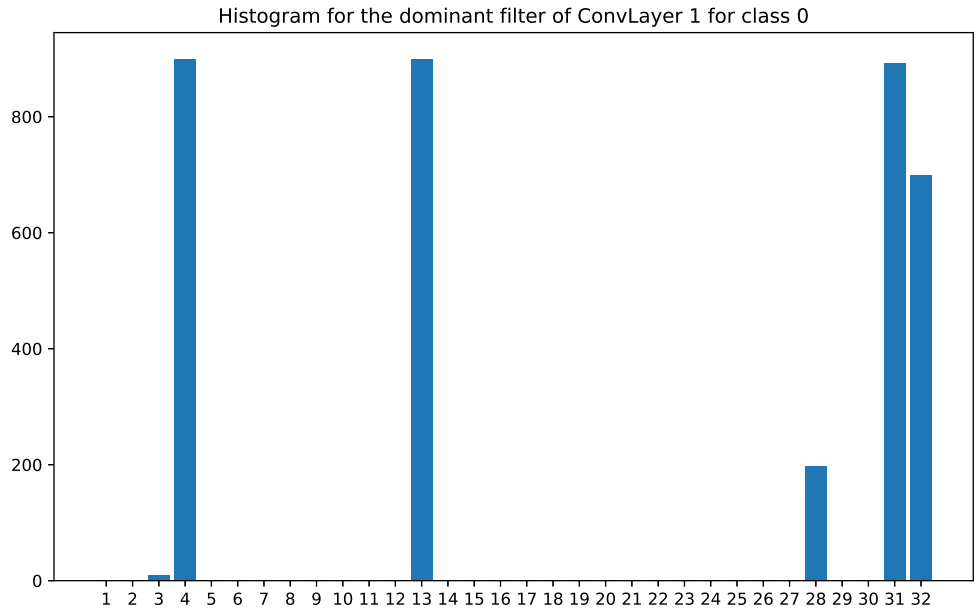

Fig. 4 The top 10% of the dominant filters for the first class

and the other with the non-dominant ones. Weights are assigned accordingly and the non-dominant filters are kept frozen. The output of the layer is reconstructed back before feeding it to the next layer. The base model's adversarial training is performed according to [23].

4 Experimental Results

As mentioned before, we generated adversarial images for the MNIST training set with $\epsilon = 0.3$ using both FGSM and PGD. After the proposed adversarial fine-tuning of the split model, and conventional adversarial training of the base model, we evaluate the models for both benign accuracy and adversarial accuracy. We recall that the accuracy on adversarial images with $\epsilon = 0.3$ for the base model was 21.26% for FGSM and 0.58% for PGD.

For comparison, we consider three models: a shallow model consisting of two convolution layers with 16 and 32 filters, respectively; a base model featuring two convolution layers with 32 and 64 filters, respectively; and a dense model comprising three convolution layers with 32, 64 and 128 filters, respectively. In Table 6 and 7, we show the performance of the three models under consideration for conventional adversarial training against FGSM and PGD attacks respectively. As mentioned before, we evaluate the models for both benign accuracy and adversarial accuracy. The adversarial accuracy increases, as expected, with adversarial training for both PGD and FGSM. But there is a drop in the benign accuracy for both cases, as can be seen in the fourth column for both adversarial training against FGSM and PGD attacks. To address this, Goodfellow et al. [16] proposed using a training set containing a mixture

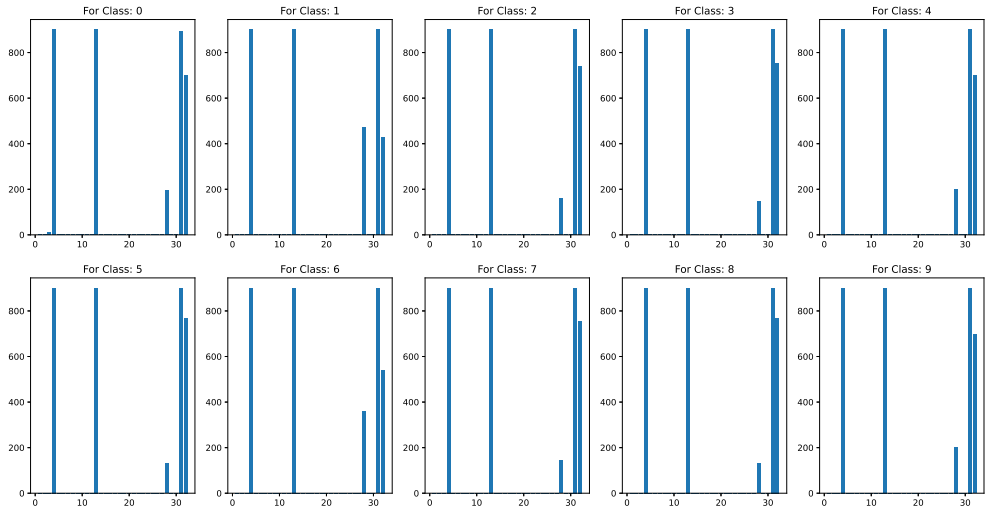

Fig. 5 10% dominant filters for convolutional layer 1

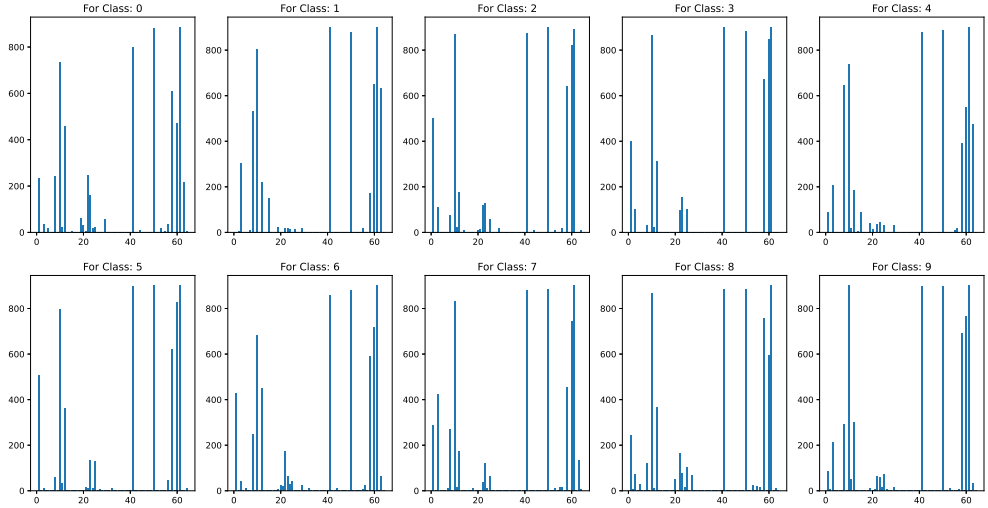

Fig. 6 10% dominant filters for convolutional layer 2

of benign and adversarial images. We follow this approach as well – we take 42,000 benign images and their adversarial counterparts, and perform adversarial training of the models. After adversarial training, the adversarial accuracy of the models over an average of five training's remains essentially the same for both FGSM and PGD attacks. For the base model, benign accuracy drops from 99.33% to 96.00% for FGSM training and from 99.33% to 87.79% for PGD training. If adversarial training is performed with a mixture of benign and adversarial training data, we can bring up the benign accuracy to satisfactory levels without having to sacrifice adversarial accuracy

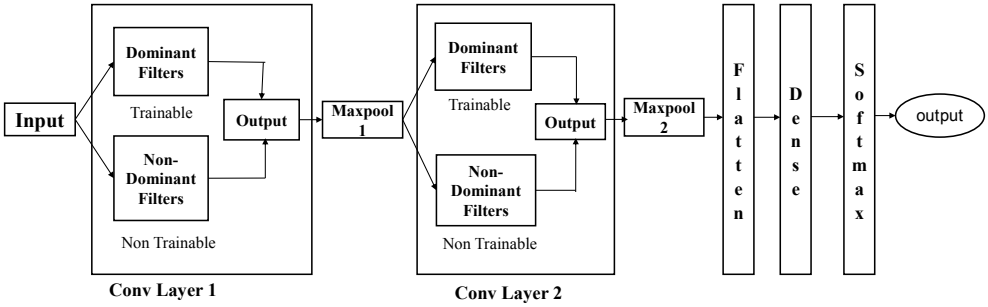

Fig. 7 Split Model for adversarial fine-tuning of dominant filters.

Table 6 Performance of the conventional adversarial training of the entire model (FGSM)

Model	Test Image Type	Before Adversarial Training	Adversarial Training with FGSM	Adversarial Training with Benign + FGSM
Shallow	Benign	98.94%	93.00%	98.86%
	Adv (FGSM)	10.61%	98.60%	98.60%
Base	Benign	99.33%	96.00%	98.83%
	Adv (FGSM)	21.26%	98.95%	98.98%
Dense	Benign	99.36%	90.11%	98.67%
	Adv (FGSM)	11.64%	98.91%	98.75%

(see the fifth column). That is, for all models under consideration, the accuracy on benign images for both FGSM and PGD adversarial training can be attained near 99.00%.

In Table 8 and 9, we show the performance of the models under consideration for our proposed adversarial fine-tuning training against FGSM and PGD attacks respectively. As we can observe from the table, our adversarial fine-tuning approach by splitting the model provides similar results as the conventional adversarial training, even though a smaller number of parameters were fine-tuned. As the conventional approach, performance of the proposed approach is better when the adversarial fine-tuning is done with a mixture of benign and adversarial training data. For the base model, benign accuracy for both FGSM and PGD training is approximately 99.00%, and the adversarial accuracy is approximately 98.50% – this is essentially the same as the conventional adversarial training results on the base model.

From Table 6 to 9, we observe that adversarial training with a mixture of benign and adversarial samples provides approximately 98% accuracy, regardless of the model type. This indicates that the dominant filters are indeed the most vulnerable portion of the model and securing just those filters provides as good a result as conventional adversarial training. However, our proposed adversarial fine-tuning is more

Table 7 Performance of the conventional adversarial training of the entire model (PGD)

Model	Test Image Type	Before Adversarial Training	Adversarial Training with PGD	Adversarial Training with Benign + PGD
Shallow	Benign	98.94%	95.94%	99.04%
	Adv (PGD)	0.89%	98.78%	98.63%
Base	Benign	99.33%	87.79%	99.13%
	Adv (PGD)	0.58%	98.89%	98.86%
Dense	Benign	99.36%	95.68%	99.05%
	Adv (PGD)	1.20%	98.72%	98.59%

Table 8 Performance of the proposed adversarial fine-tuning of the split model (FGSM)

Model	Test Image Type	Adversarial Fine-tuning with FGSM	Adversarial Fine-tuning with Benign + FGSM
Shallow	Benign	94.03%	98.77%
	Adv (FGSM)	98.77%	98.30%
Base	Benign	95.45%	98.91%
	Adv (FGSM)	98.60%	98.61%
Dense	Benign	64.67%	98.89%
	Adv (FGSM)	98.20%	98.00%

computation-friendly, since it involves a smaller amount of trainable parameters. The advantage of this method becomes greater with denser, and more complex networks containing many trainable parameters.

Thus far, we chose the top 10% filters as dominant filters. We investigate the effect of choosing more dominant (and therefore, trainable) filters. As shown in Table 10 and 11, performance does not change noticeably for either the PGD or the FGSM. The accuracy findings show the benign and adversarial accuracies for the three models when the percentage of trainable filters in the convolutional layers are increased. We argue that with this combination of dataset and attack model, the vulnerability of the model lies mostly within the top 10% of the dominant filters, which once again proves that adversarial training of the entire network is somewhat wasteful.

In Table 12, we show the percentage of trainable parameters reduced when using the proposed adversarial fine-tuning of the split model. For the dense model, more than half of the model does not need adversarial training for robust performance against adversarial attacks. As before, we argue that the proposed method of adversarial fine-tuning is more advantageous as models get bigger. These results are consistent even when the level of perturbation of the attacking images is changed.

Table 9 Performance of the proposed adversarial fine-tuning of the split model (PGD)

Model	Test Image Type	Adversarial Fine-tuning with PGD	Adversarial Fine-tuning with Benign + PGD
Shallow	Benign	93.93%	98.86%
	Adv (PGD)	98.15%	97.75%
Base	Benign	94.11%	99.01%
	Adv (PGD)	98.76%	98.56%
Dense	Benign	76.97%	98.01%
	Adv (PGD)	95.67%	95.05%

Table 10 Effect of Dominant Filter Percentage for FGSM

Dominant Filters	Test Image Type	Adversarial Fine-tuning with FGSM	Adversarial Fine-tuning with Benign + FGSM
10%	Benign	95.99%	99.13%
	Adv (FGSM)	98.69%	98.53%
20%	Benign	96.33%	99.10%
	Adv (FGSM)	98.64%	98.52%
50%	Benign	95.92%	99.07%
	Adv (FGSM)	98.86%	98.77%

Remarks. Adversarial training may not be an *one-shot* approach for the model to be robust against future adversarial attacks. If the attackers gets access to a model information; fully or partially; adversarial examples can be generated even if a model is trained against it. Especially when models are trained for FGSM attacks, since FGSM is a one-step algorithm that computes the perturbation with a single step in the direction of the gradient of the loss function. It does not explore the gradient of the loss function in its entirety. Attackers can take advantage of this characteristic by computing new adversarial attacks the model is not trained against, if model information is revealed. Thus, a model trained against FGSM attacks may need further training if it is suspected that model information has been leaked. This underlines one advantage of our proposed split model fine-tuning, where we can achieve almost similar results using much lower computational burden. Additionally, PGD training provides better defense it takes an iterative approach to the gradient descent in order to maximize the loss. A stronger adversarial training, such as, with PGD adversarial images, will not only secure the model against PGD attacks, but also provide protection against other weaker attacks (such as FGSM) to a certain extent. Implementing our proposed method with second order adversarial attacks (i.e., the Carlini-Wagner

Table 11 Effect of Dominant Filter Percentage for PGD

Dominant Filters	Test Image Type	Adversarial Fine-tuning with PGD	Adversarial Fine-tuning with Benign + PGD
10%	Benign	94.11%	99.01%
	Adv (PGD)	98.76%	98.56%
20%	Benign	93.40%	99.08%
	Adv (PGD)	98.70%	98.67%
50%	Benign	91.29%	99.17%
	Adv (PGD)	98.88%	99.00%

Table 12 Percentage of Trainable Parameters Reduced

Dominant Filters	Shallow Model	Base Model	Dense Model
10%	7.96%	14.33%	64.07%
20%	7.08%	12.81%	56.93%
50%	4.54%	8.05%	35.73%

[26] approach) is deferred for future research. Finally, our current research focused on simple neural network models, trained on the MNIST dataset. We defer the extension to other complicated models for future work.

5 Conclusion

Adversarial attacks take advantage of *the excess capacity* of the neural network models in such a way that makes subliminal adjustments to the inputs (imperceptible to humans), and thereby causes the model to make inaccurate predictions. Since neural network based models are deployed in several critical applications, strong defense mechanisms against such adversarial attacks are rightfully warranted. However, existing approach for attaining model robustness against adversarial attacks is adversarial training, which typically provides defense against specific attack types and requires substantial computational resources. In this work, we showed that by algorithmically identifying specific vulnerable parts of the neural network model, and performing adversarial fine-tuning of those parts, we can attain the same level of performance as the conventional adversarial training. Our analysis reveals that only a small portion of the vulnerable components accounts for a majority of the model's errors caused by adversarial attacks. As such, we propose to selectively fine-tune these vulnerable components, which ensures significant computational-load savings. We empirically validate our proposed approach on the MNIST dataset, and demonstrate that our approach can achieve similar performance as the more resource-intensive conventional adversarial training method.

Acknowledgments. The authors would like to express their sincere gratitude towards the Department of Electrical and Electronic Engineering (EEE) of Bangladesh University of Engineering and Technology (BUET) for providing support for research.

References

[1] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)

[2] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[3] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., *et al.*: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine **29**(6), 82–97 (2012)

[4] Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M.: Globally normalized transition-based neural networks. arXiv preprint arXiv:1603.06042 (2016)

[5] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)

[6] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., *et al.*: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484–489 (2016)

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

[8] Chernikova, A., Oprea, A., Nita-Rotaru, C., Kim, B.: Are self-driving cars secure? evasion attacks against deep neural networks for steering angle prediction. In: 2019 IEEE Security and Privacy Workshops (SPW), pp. 132–137 (2019). https://doi.org/10.1109/SPW.2019.00033

[9] Gu, S., Rigazio, L.: Towards Deep Neural Network Architectures Robust to Adversarial Examples (2015). https://arxiv.org/abs/1412.5068

[10] Chalupka, K., Perona, P., Eberhardt, F.: Visual Causal Feature Learning (2015). https://arxiv.org/abs/1412.2309

[11] Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In:

2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597 (2016). https://doi.org/10.1109/SP.2016.41

[12] Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial Machine Learning at Scale (2017). https://arxiv.org/abs/1611.01236

[13] Rozsa, A., Gunther, M., Boult, T.E.: Towards Robust Deep Neural Networks with BANG (2018). https://arxiv.org/abs/1612.00138

[14] Torkamani, M.A.: Robust large margin approaches for machine learning in adversarial settings. PhD thesis, University of Oregon (2016)

[15] Fawzi, A., Fawzi, O., Frossard, P.: Analysis of classifiers' robustness to adversarial perturbations. Machine learning **107**(3), 481–508 (2018)

[16] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015) https://doi.org/10.48550/arXiv.1412.6572

[17] Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 427–436 (2015)

[18] Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 Acm Sigsac Conference on Computer and Communications Security, pp. 1528–1540 (2016)

[19] He, W., Wei, J., Chen, X., Carlini, N., Song, D.: Adversarial example defense: Ensembles of weak defenses are not strong. In: 11th USENIX Workshop on Offensive Technologies (WOOT 17) (2017)

[20] Papernot, N., McDaniel, P.: On the effectiveness of defensive distillation. arXiv preprint arXiv:1607.05113 (2016)

[21] Carlini, N., Wagner, D.: Defensive distillation is not robust to adversarial examples. arXiv preprint arXiv:1607.04311 (2016)

[22] Carlini, N., Wagner, D.: 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 (2017)

[23] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks https://doi.org/10.48550/arXiv.1706.06083

[24] Keras: MNIST digits classification dataset. https://keras.io/api/datasets/mnist/

[25] Toronto., C.L.: Cleverhans-lab. https://tinyurl.com/488xaf2h (2021)

[26] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 Ieee Symposium on Security and Privacy (sp), pp. 39–57 (2017). Ieee

1706161 - Sadia Afrose

From: Signal, Image and Video Processing <sankari.gireesa@springernature.com>
Sent: Monday, November 18, 2024 1:18 PM
To: 1706161 - Sadia Afrose
Subject: Signal, Image and Video Processing - Receipt of Manuscript 'Efficient Defense Against...'

Ref: Submission ID 4d662a2a-17f3-4f8d-a9a3-d871cb347d60

Dear Dr Afrose,

Please note that you are listed as a co-author on the manuscript "Efficient Defense Against First Order Adversarial Attacks on Convolutional Neural Networks", which was submitted to Signal, Image and Video Processing on 18 November 2024 UTC.

If you have any queries related to this manuscript please contact the corresponding author, who is solely responsible for communicating with the journal.

Kind regards,

Editorial Assistant
Signal, Image and Video Processing