The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

MTDeep: Boosting the Security of Deep Neural Nets Against Adversarial Attacks with Moving Target Defense

Sailik Sengupta, Tathagata Chakraborti, Subbarao Kambhampati
Arizona State University, United States
{sailiks,tchakra2,rao}@asu.edu

Abstract

Recent works on gradient based attacks and universal perturbations can adversarially modify images to bring down the accuracy of state-of-the-art classification techniques based on deep neural networks to as low as 10% on popular datasets like MNIST and ImageNet. The design of general defense strategies against a wide range of such attacks remains a challenging problem. In this paper, we derive inspiration from recent advances in the fields of cybersecurity and multi-agent systems and propose to use the concept of *Moving Target Defense (MTD)* for increasing the robustness of a set of deep networks against such adversarial attacks. To this end, we formalize and exploit the notion of *differential immunity* of an ensemble of networks to specific attacks. To classify an input image, a trained network is picked from this set of networks by formulating the interaction between a Defender (who hosts the classification networks) and their (Legitimate and Malicious) Users as a repeated *Bayesian Stackelberg Game (BSG)*. We empirically show that our approach, MTDeep reduces misclassification on perturbed images for MNIST and ImageNet datasets while maintaining high classification accuracy on legitimate test images. Lastly, we demonstrate that our framework can be used in conjunction with any existing defense mechanism to provide more resilience to adversarial attacks than those defense mechanisms by themselves.

Introduction

State-of-the-art systems for image classification based on Deep Neural Networks (DNNs) are used in many important tasks such as recognizing handwritten digits on cheques (Jayadevan et al. 2012), object classification for automated surveillance (Javed and Shah 2006) and autonomous vehicles (De La Escalera et al. 1997). Adversarial intent to make these classification systems misclassify inputs can lead to dire consequences. For example, being able to make a classifier misclassify the digit '1' as '9' might help an adversary withdraw more money from the bank than the amount handwritten on a cheque. In fact, in (Papernot et al. 2016a) and (Szegedy et al. 2013) authors show how models for handwritten digit recognition built using the MNIST dataset can be easily attacked. In (Papernot et al. 2016a), road signs saying 'stop' are misclassified, which can make an autonomous vehicle behave dangerously. Such attack mechanisms also exist for state-of-the-art vision systems that recognize faces, which may be used for authentication, target identification etc. as shown in (Sharif et al. 2016). Furthermore, the manipulated image generated by an adversary is, in almost all cases, indistinguishable from the original image when viewed by a human observer.

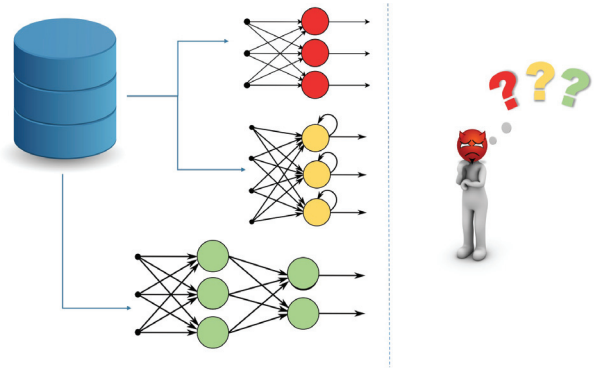


Figure 1: To classify an input image, the MTDeep system uses a network selected randomly from an ensemble of networks. As an attacker is not sure which network will classify their malicious input, the success rate of an attack that affects only one network in the ensemble is reduced.

These adversarial attacks exploit the fact that Deep Neural Networks (DNNs) have high biases in certain regions of the high dimensional space onto which input data is projected. Adversaries try to perturb legitimate input data towards a decision boundary so that it is misclassified by the network. Formally, if $\hat{D}(x)$ denotes the class of an image x output by a Deep Neural Network $\hat{D}$, an adversarial perturbation p when added to the image x tries to ensure that $\hat{D}(x) \neq \hat{D}(x+p)$. Minimum perturbations, in addition, try to minimize some norm of p, which ensures that the changed image $x+p$ and the original image x are indistinguishable to humans. Fast Gradient (FGSM) attack (Szegedy et al. 2013) and Jacobian Saliency Maps (JSM) (Papernot et al. 2016a) are examples of methods used for such purposes.

Popular defenses against such attacks, at a high level, generate adversarial samples using an attack algorithm and incorporate them into the training set (with the correct labels) so that the classifier is now trained on data that gets pro-

jected into empty regions of the higher dimensional space. This process is known as adversarial training.

Recent work on universal adversarial perturbation generates a single perturbation for a Neural Network classifier, which when added to any input images, can adversely bring down the classification accuracy of even the state-of-the-art classifiers (ResNet-152 (He et al. 2016)) from 95.5% to as low as 14.6% on the Imagenet dataset (Moosavi-Dezfooli et al. 2016). Unfortunately, the authors demonstrate that adversarial training (termed as 'fine-tuning' in the paper) is an ineffective defense mechanism for this attack.

In this paper, we propose Moving Target Defense (MTD) as a general technique for defending Deep Neural Networks against all class of attacks (see Figure 1) that can be used in conjunction with existing defense mechanisms. In the section on related works, we categorize the state-of-the-art attacks on Deep Neural Networks for image classification into three classes, followed by introduction to the relevant literature in the fields of cybersecurity and multi-agent systems that form the backbone of our approach. In this paper, we

(i) Propose MTDeep - an MTD framework for DNNs that can be used as 'security-as-a-service' to bootstrap any existing defense mechanism to increase the robustness of a classification system to adversarial attacks.

(ii) Formalize the notion of differential immunity for MTD systems and propose the problem of developing differentially immune network configurations, which in turn seeks to reduce the transferability of perturbed images, as an open research problem.

(iii) Formulate the interaction between MTDeep and its users as a Repeated Bayesian Game and find the optimal switching strategy in order to exploit the differential immunity of its composing classifiers. We show that this is necessary as simple strategies like uniform random selection might even be worse in terms of security than using single classifiers (rendering the MTD useless).

(iv) Show that finding the Stackelberg equilibrium of this game seeks to solve a multi-objective optimization problem that maximizes classification accuracy on legitimate input data and reduces misclassification error on adversarially modified data.

We empirically demonstrate the effectiveness of our contributions on MNIST and Imagenet datasets.

Related Work

Attacks on Deep Neural Networks and Existing Defense Strategies

In this section we will do a brief review of existing work on crafting adversarial attacks on deep neural networks and efforts being made to combat them.

• **Gradient-based perturbations:** Recent literature has shown multiple ways of generating adversarial samples for a test image input to a DNN (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Papernot et al. 2016a; Szegedy et al. 2013). In these works, either (i) the input features whose partial derivatives on the DNN's Loss Functions are high are modified by a small amount to make the DNN misclassify them, or (ii) the geometric space around a point is examined to find the closest class-separation boundary. Apart from assuming that the test image which is to be modified is available beforehand, similar to a chosen ciphertext attack, these attacks often make further assumptions pertaining to knowledge about the network that is being used for classification.

• **Black-box attacks:** Black-box attacks against DNNs train a (smaller) substitute model by assuming that the network being attacked provides test labels for a list of images the adversary provides (Papernot et al. 2017), similar to chosen plaintext attacks. Gradient based attacks on this substitute model have been shown to generalize to the actual network. Recent work on zeroth order optimization has further shown it is possible to create black-box attacks without the need for substitute models (Chen et al. 2017).

A popular defense against these types of attacks involves first generating adversarial images by modifying the training inputs using one (or all) of the attack methods described. The generated images along with the expected labels are then used to fine tune the parameters of the neural network in the training phase. This helps the DNN to reduce its bias in the unexplored high dimensional space, reducing the effectiveness of the adversarial perturbations. Ensemble adversarial training (Tramèr et al. 2017) and stability training (Zheng et al. 2016) are improvements on this defense technique. Besides these, other methods like defensive distillation (Papernot et al. 2016b) and dimensionality reduction & 'anti-whitening' (Bhagoji, Cullina, and Mittal 2017) also exist but we refrain from discussing these in detail since our proposed framework can be used in conjunction to any of these to provide stronger security guarantees.

• **Universal perturbations:** In this attack, a single perturbation image for a particular network is constructed. This DNN-specific perturbation when added to any input image, makes the network misclassify it (Moosavi-Dezfooli et al. 2016). Thus, although it might be time consuming to generate a single perturbation, only one "universal" perturbation image per network needs be computed. Interestingly, the authors show that adversarial training is ineffective in increasing robustness against these attacks (and the other works on developing defense mechanisms, mentioned above, have not shown that they can mitigate this either).

Besides these, there has been some effort in trying to protect machine learning systems against attacks using randomization techniques such as in (Biggio, Fumera, and Roli 2008). Unfortunately, the framework proposed is not generic enough to be used for DNNs. Furthermore, while such works try to prevent misclassification under attack, they land up reducing the classification accuracy significantly. In contrast, work on using ensemble models for DNNs simply tries to increase classification accuracy for legitimate users but provide no protection against adversarially modified test images (Ioffe and Szegedy 2015). There has also been work on using ensemble models to detect adversarial samples for the MINST dataset (Abbasi and Gagné 2017). This requires the distribution of how a sample from a specific class is likely to be misclassified into another class, which is either unavailable or difficult to obtain in most cases. Furthermore, the idea of using a classifier (or learning a controller) to classify

an input image as legitimate or adversarial is highly insecure since that classifier (or controller) itself can be adversarially attacked. Thus, existing works for securing DNNs against adversarial attacks are either mostly classifier/attack/dataset specific or fail to reason about attacker strategies.

Adoption of Moving Target Defense (MTD) for Boosting the Security of DNNs

Moving Target Defense (MTD) is a paradigm used in software security that tries to reduce the success rate of an attack by constantly switching between multiple software systems (Zhuang, DeLoach, and Ou 2014). Practical use of MTDs in Web Application Systems have been shown to enhance system security (Taguinod et al. 2015). Based on the principles of MTD, we design a general purpose security framework for Neural Networks (MTDeep) in this paper.

Devising strategies for MTD systems have been shown to be a difficult problem. In order to provide formal guarantees about the security of such systems, one needs to reason about these attacks in a multi-agent game theoretic fashion (Sengupta et al. 2017). They show that this leads to defense strategies that outperform trivial randomization strategies.

Thus, we compile the interaction between the image classification system driven by an ensemble of DNNs (MTDeep) and its users into a Repeated Bayesian Game, providing provable guarantees on the expected performance and security of the system. In MTDeep, an input image is classified by one of the networks in the ensemble chosen randomly based on a strategy generated using game-theoretic reasoning in the formulated multi-agent setup.

Although our framework is motivated to provide security for DNNs, it can also be used for boosting security of any Machine Learning (ML) model against any form of attack, since the switching strategies can be easily generated for any 'known' set of attacks. The game theoretic reasoning ensures that the attacker cannot increase the misclassification rate by strategic manipulation over the space of these known attacks. Most importantly, MTDeep reasons about trying to balance between providing security while affecting the accuracy for legitimate users of the system only by a small amount. Such a consideration is often absent in present works on the design of security mechanisms for DNNs.

Moving Target Defense for Deep Neural Networks (MTDeep)

In a Moving Target Defense (MTD) system, the defender has multiple system configurations. The attacker has a set of attacks that it can use to affect some of the configurations in the defender's system. Given an input to the system, the defender selects one of the configurations to run the input and returns the output generated by that system. Since the attacker does not know which system was specifically selected, it is no longer as effective as before (Figure 1). Thus, randomization in selecting a configuration for classification (on each input) is paramount. A potential downside of such a framework is that it might land up reducing the accuracy of the overall system in classifying non-perturbed images. Thus, we want to retain good classification performance while guaranteeing high security.

In this section, we first describe the agents in our framework and actions they can execute, which include describing the defender along with their DNN configurations for the MTD framework and the user–their types (adversarial and legitimate) and the action set of each type. Lastly, we show that randomized switching over the set of defender's configurations needs to reason about the MTDeep system in a game theoretic fashion. An equilibria of this game gives us an optimal selection strategy that maximizes classification accuracy and security for the defender's system.

Defender Configurations

The configuration space for the defender in the MTD framework for DNNs are a ensemble of DNNs that are trained on the same task but ideally not affected by the same attack. For classifying images, Convolutional Neural Networks (CNNs) are known to produce the best results. Thus, although the different DNN configurations used by the defender might differ in the number of layers, parameters, hyperparameters or activation functions, they will likely have to use CNN units to produce comparable results. Formally, let N denote the set of defender configurations. In our experiments, N is an ensemble of three and six neural networks for MNIST (see Table 1) and ImageNet (see Table 3) respectively.

User Types and Action Sets

Our second player, namely the users, are of two types– Legitimate User ($\mathcal{L}$) and the Adversary ($\mathcal{A}$). $\mathcal{L}$ tries to use the MTDeep system for classifying images for a specific task without any adversarial intent. These are the target users of most the present machine learning 'as-a-service' providers. The second type, i.e. the adversary $\mathcal{A}$, is essentially trying to perturb input images to make the DNNs misclassify the label for these inputs. $\mathcal{L}$ has a single action that represents inputting an image for classification, where as the attacker may use multiple attack actions.

In our threat model, we consider a strong adversary who knows the different architectures we use in our MTDeep system. Thus, they can generate an attack (say, using the Fast Gradient Method or Universal Perturbations) for each of the networks in our system. We let U denote this set of attacks the attacker generated against our system. Note that an attack ($\in U$) generated for a specific network in MTDeep, may or may not be effective for the other configurations which brings us to the concept of *Differential Immunity*.

Differential Immunity

For Neural Networks, the effectiveness of an attack is directly proportional to its misclassification (or fooling) rate, i.e, the probability of test samples it can make the network mis-classify. Formally, let $E : N \times U \rightarrow [0, 100]$ denote this fooling rate function where, $e_{n,u}$ is the fooling rate when an attack $u(\in U)$ is used against a network $n(\in N)$.

For an attack ($u \in U$), we would like it to be effective for only one particular configuration and ineffective for all the others. Whenever this property holds for all attacks against

MTDeep	Legitimate User ($\mathcal{L}$) Classification Image
CNN	$(99.1, 99.1)$
MLP	$(98.3, 98.3)$
Hierarchical-RNN	$(98.7, 98.7)$

Table 1: Pay-off matrix for the defender and the Legitimate User ($\mathcal{L}$) type for MNIST. The reward for both the players is the accuracy of classification on non-perturbed images.

MTDeep	Adversarial User ($\mathcal{A}$) $FGSM_{CNN}$	$FGSM_{MLP}$	$FGSM_{HRNN}$
CNN	$(11.63, 88.37)$	$(47.54, 52.46)$	$(74.65, 25.35)$
MLP	$(36.37, 63.63)$	$(1.96, 98.04)$	$(38.10, 61.90)$
HRNN	$(35.72, 64.28)$	$(24.08, 75.92)$	$(9.65, 90.35)$

Table 2: Table showing the normal form game matrix for the defender and the Adversarial User ($\mathcal{A}$) type. The reward values for $\mathcal{A}$ is equal to the fooling rate and the reward for the defender is the accuracy of the system under attack.

an MTD system, there is something to be gained by switching between multiple configurations (and thus using MTD). The property called *differential immunity* aims to capture this. We now define differential immunity δ formally as,

$$\delta(U, N, E) \quad = \quad \min_{u} \frac{\max_n e_{n,u} - \min_n e_{n,u} + 1}{\max_n e_{n,u} + 1}$$

where $\min_n e_{n,u}$ and $\max_n e_{n,u}$ denote the minimum and maximum impact that an attacker can cause when using the attack u against the MTD system. Notice that if the maximum and minimum impact differ by a wide margin, then the differential immunity of the MTD system should be higher. This is represented in the numerator. The denominator ensures that an attack which has high impact reduces the differential immunity of a system compared to a low impact attack even when their maximum and minimum values differ by the same margin. The $+1$ factor in the denominator of the function prevents division by zero. The $+1$ in the numerator is to ensure that when $\min_n e_{n,u} = \max_n e_{n,u}$, the higher the value of $\max_n e_{n,u}$, lesser the δ and *vice versa*. Notice that $0 \leq \delta \leq 1$ since $0 \leq e_{n,u} \; \forall \; n, u$.

Constructing an ensemble N of DNNs with has high differentially immunity is non-trivial in light of existing work (Szegedy et al. 2013). The authors show that ideas like partitioning the training data and training the different networks ($\in N$) on the disjoint data sets does not seem to make the networks differentially immune. The notion of transferability of an attack against DNNs introduced by the authors seems to be opposite to the notion the differential immunity of an ensemble of networks. We show later that even an ensemble with various architectures, we can provide only a weak level of differential immunity.

MTD as a Repeated Bayesian Game

For MTDeep to be secure, it should randomly pick a network n each time to classify an input image. If we use a naive strategy like uniform random to pick a network, we will have equal chances of choosing networks that have low classification accuracy or high vulnerability to perturbed images. Also, with time the attacker will eventually infer the defender's strategy and exploit the highly vulnerable configurations which might lead to worse security guarantees for the MTDeep than using a single network (as we show in our experiments). We now formulate our system as a Repeated Bayesian Game to design an effective strategy.

The use of our MTDeep framework in safety critical systems should ensure that the accuracy of classification for

MTDeep	Legitimate User ($\mathcal{L}$) Classification Image
VGG-F (Chatfield et al. 2014)	$(92.9, 92.9)$
CaffeNet (Jia et al. 2014)	$(83.6, 83.6)$
GoogLeNet (Szegedy et al. 2015)	$(93.3, 93.3)$
VGG16 (Simonyan and Zisserman 2014)	$(92.5, 92.5)$
VGG19 (Simonyan and Zisserman 2014)	$(92.5, 92.5)$
ResNet-152 (He et al. 2016)	$(95.5, 95.5)$

Table 3: Table showing the normal form game matrix for the defender and the Legitimate User ($\mathcal{L}$) type for ImageNet. The reward values for both the players is same as the classification accuracy.

legitimate users is not affected. In essence, we want MTDeep to be effective for the legitimate users and alongside increase the accuracy of classification for the adversary generated images, making this a multi-objective optimization. Fortunately, this can easily be captured by using the probability of player types in our game theoretic framework that lets us associate relative importance to each of the user types ($\mathcal{L}$ and $\mathcal{A}$). Thus, the two types of users have a probability associated with them, making this a Bayesian Game.

An (intelligent) adversary can infer the switching strategy of the defender by probing or observing the traffic on the system for a reasonable amount of time. They can then reason about their attacks when attacking the system. In our threat model, the defender has to account for this behavior before choosing their switching strategy, making this a Repeated Game. We now formulate the game as a non-zero sum game where the player utilities are defined as follows:

- For the Legitimate User, $\mathcal{L}$ and the defender both get a reward value that represents the accuracy of the DNN system. Thus, for using a network n in the ensemble N with classification accuracy (say) 93% for an input image both the defender and $\mathcal{L}$ get a reward of 93 (see Tables 1 & 3).
- For the Adversary, the reward values for an attack u against the network n is given by $e_{n,u}$, which corresponds to the fooling rate. The defender's reward in this case is the accuracy n when classifying the input image perturbed using u, which is $(100 - e_{n,u})$ (see Tables 2 & 4).

Defender's Strategy for Switching

A defender has to launch the MTDeep system first to begin the game. This imparts a leader-follower paradigm to

	Adversarial User ($\mathcal{A}$)					
MTDeep	UP_{VGG-F}	$UP_{CaffeNet}$	$UP_{GoogLeNet}$	UP_{VGG-16}	UP_{VGG-19}	$UP_{ResNet-152}$
VGG-F	$(6.3, 93.7)$	$(28.2, 71.8)$	$(51.6, 48.4)$	$(57.9, 42.1)$	$(57.9, 42.1)$	$(52.6, 47.4)$
CaffeNet	$(26.0, 74.0)$	$(6.7, 93.3)$	$(52.3, 47.7)$	$(60.1, 39.9)$	$(60.1, 39.9)$	$(52.0, 48.0)$
GoogLeNet	$(53.8, 46.2)$	$(56.2, 43.8)$	$(21.1, 78.9)$	$(60.8, 39.2)$	$(60.2, 39.8)$	$(54.5, 45.5)$
VGG-16	$(36.6, 63.4)$	$(44.2, 55.8)$	$(43.5, 56.5)$	$(21.7, 78.3)$	$(26.9, 73.1)$	$(36.6, 63.4)$
VGG-19	$(36.0, 64.0)$	$(42.8, 57.2)$	$(46.4, 53.6)$	$(26.5, 73.5)$	$(22.2, 77.8)$	$(42.0, 58.0)$
ResNet-152	$(53.7, 46.3)$	$(53.7, 46.3)$	$(49.5, 50.5)$	$(53.0, 47.0)$	$(54.5, 45.5)$	$(16.0, 84.0)$

Table 4: Table showing the normal form game matrix for the defender and the Adversarial User ($\mathcal{A}$) type. The reward values for $\mathcal{A}$ is equal to the fooling rate and the reward for the defender is the accuracy of the system under attack.

the formulated Repeated Bayesian Game where the defender leads and the attacker follows over a repeated time frame to infer the leader's strategy. Satisfying the multi-objective criterion, mentioned above, is now equivalent to finding the Stackelberg Equilibrium of this game.

We now find the Stackelberg equilibrium in our game by using the optimization problem formulated in (Paruchuri et al. 2008). Let us denote the strategy vector for the defender as $\vec{x}$ and their reward as $R^D_{n,u}$ when the defender uses the network n and user selects the action u. Similarly, the strategy vectors for the adversary and the legitimate user types are $\vec{q}^{\mathcal{A}}$ and $\vec{q}^{\mathcal{L}}$ and their rewards are $R^{\mathcal{A}}_{n,u}$ and $R^{\mathcal{L}}_{n,u}$ respectively. We seek to maximize the defender's reward while allowing the attacker to choose the most effective attack, which can be described as follows,

$$\begin{aligned}
\max_{x,q} \sum_{n\in N} (\alpha \cdot \sum_{u\in U} R^D_{n,u}\, x_n q^{\mathcal{A}}_u \quad &+ \quad (1-\alpha)\cdot R^D_{n,u}\, x_n q^{\mathcal{L}}_u) \\
\text{s.t.} \quad \sum_{n\in N} x_n &= 1 \\
\sum_{u\in U} q^{\mathcal{D}}_u &= 1 \\
0 \le \; x_n \; &\le 1 \quad \forall\, n \in N \\
q^{\mathcal{D}}_u \; &\in \{0,1\} \\
0 \le v^{\mathcal{D}} - \sum_{n\in N} R^{\mathcal{D}}_{n,u} x_n \; &\le (1 - n^{\mathcal{D}}_a) M \\
\forall\, u \in U^{\mathcal{D}} \qquad &\forall\, \mathcal{D} \in \{\mathcal{A}, \mathcal{L}\}
\end{aligned}$$

where α, the probability of the adversary $\mathcal{A}$ attacking a MTDeep system and M is a large positive number. Equation 1 maximizes defender's expected reward (i.e, classification accuracy) over $\vec{x}$ and the strategy vector of the user types weighted by the relative importance assigned to the attacker, denoted as α. The first four constraints ensure that the strategy vectors sum up to 1 since individual x_n and q_u represent probability values for the defender to select network n and user to select action u. The last constraint represents the dual of the attacker's optimization problem which tries to maximize their expected reward $v^{\mathcal{D}}$ over the defender's strategy. This constraint captures the fact that the attacker knows $\vec{x}$ and uses it to select its attack strategy $\vec{q}$. Notice that the second constraint forces the users $\mathcal{L}$ and $\mathcal{A}$ to select a pure strategy. As the authors in (Paruchuri et al. 2008) show, this constraint is not limiting for the attacker because for the attacker $\mathcal{A}$ there always exists a pure strategy in support of any mixed strategy it can select that gives $\mathcal{A}$ the same reward. For the attack the attacker selects, the right side of the last constraint becomes 0 making $v^{\mathcal{D}} = \sum_{n\in N} R^{\mathcal{D}}_{n,u}$.

The defender's strategy, in the worst case, will be a pure strategy that directs it to use a single network for classification. This is equivalent to most modern day classifiers.

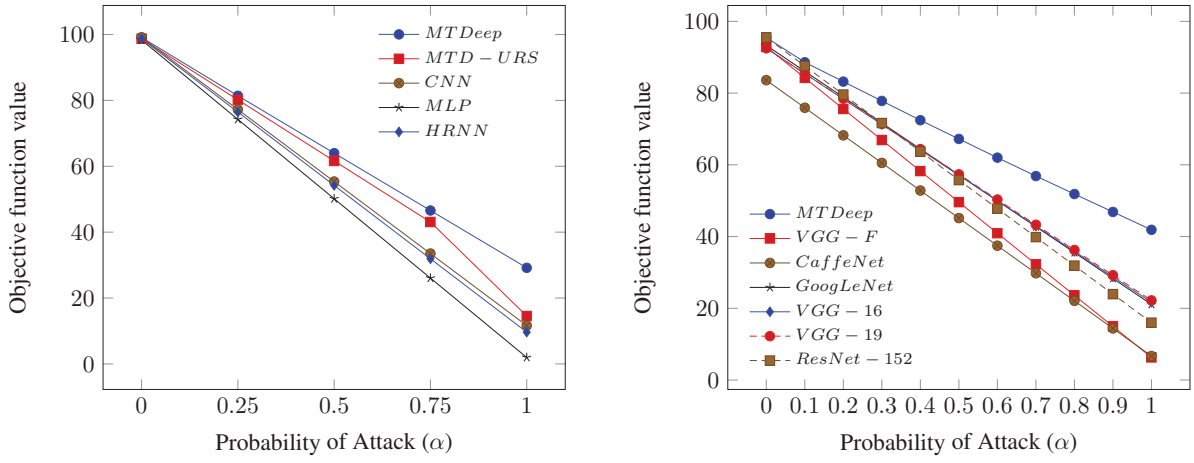
Experimental Results

In this section, we first compare the effectiveness of MTDeep on non-adversarially trained ensemble of networks for MNIST and ImageNet datasets against the constituent networks and a baseline MTD with DNNs that uses Uniform Random Strategy we call MTD-URS. We then show that piggybacking an existing defense mechanism (like adversarial training) with MTDeep results in robustness gains against adversarial attacks when using the BSG formulation.

MTDeep with Non-Adversarilly Trained Networks

MNIST For our experiments, we trained three networks in the MNIST dataset which were based on (and hence called) Convolution Neural Net (CNN), Multi Layer Perceptron (MLP) and Hierarchical Recurrent Neural Net (HRNN) to produce relatively high classification accuracies. The size of the training and testing sets were 50000 and 10000 images respectively. We then developed Fast Gradient Based attacks (with perturbation bounded by $\epsilon = 0.3$) for each of these networks and calculated its misclassification rate on the test set for all the networks. These values and the classification accuracies of the networks are were used to obtain the utilities of the players in our game (2 & Table 1 respectively). The value of δ for this system is only 0.29. Interestingly, we notice that even with different types of fundamental units (like CNN, MLP and RNN) for the individual networks the transferability of $FGSM$ attacks is high.

In Figure 2a, we plot the objective values of using MTDeep *Vs.* any of the single networks as α varies from 0 to 1. When $\alpha = 0$, the MTDeep always selects the CNN to produce maximum classification accuracy of 99.1%. In contrast, MTD-URS has a classification accuracy of 98.2% because it uses the lesser accurate classifiers as well. When $\alpha = 1$, the single networks misclassify 98.0% (in the worst case) and 88.4% (in the best case) of the input images. MTDeep misclassifies only 70% of the time even when it



(a) MNIST

(b) Imagenet

Figure 2: Expected accuracy of the non-adversarially trained Deep Neural Network for the legitimate user ($\alpha = 0$) and the attacker ($\alpha = 1$) when using MTDeep *vs.* any one of the original networks.

uses such highly vulnerable networks as a part of its ensemble. The mixed strategy of the defender in this case is $\vec{x} = (0.274, 0.061, 0.665)$. MTD-URS misclassifies 76% of the time because it picks more vulnerable configurations with equal probability. With a misclassification rate of 70%, MTDeep is not meant to be a stand-alone solution but as a service that has the potential to improve the security of classification systems based on DNNs in the front end especially when no effective defense mechanisms exist.

Imagenet For our experiments, we use six different networks that have excelled in ILSVRC 2012 (Russakovsky et al. 2015) validation set ($50,000$ images) in the defender's ensemble (see Table 3). Since generating Fast Gradient Attacks on these large networks for every single images is time consuming, we use Universal Perturbations (UP) developed for each network in (Moosavi-Dezfooli et al. 2016), which have to be generated only once. These are the attack actions of the adversary $\mathcal{A}$ for this game. These UPs were generated by making sure that the $l-\infty$ norm of the perturbations were less than a bound $\xi = 10$. The utilities for this game is shown in Table 4 for $\mathcal{A}$ and Table 3 for $\mathcal{L}$. As state earlier, researchers have shown that adversarial training are ineffective against this attack (Moosavi-Dezfooli et al. 2016) and no other proven defense mechanisms exist.

The differential immunity of this ensemble, although greater than 0.29 for the MTDeep system for MNIST, is only 0.34. In Figure 2b, we plot the objective function value (given by Equation 1) for the MTDeep and ignore MTD-URS (as it is will always performs worse than MTDeep) along with the objective values of each of the constituent networks when the probability of an adversary type α varies. When $\alpha = 0$, MTDeep uses the most accurate network that maximizes the classification accuracy. Hence, the plots for ResNet-152 and MTDeep start at the same place. As adversarial test samples become more ubiquitous, the accuracy against perturbed inputs drops for all the constituent networks of the ensemble. Thus, to stay protected, MTDeep starts to switch between the networks.

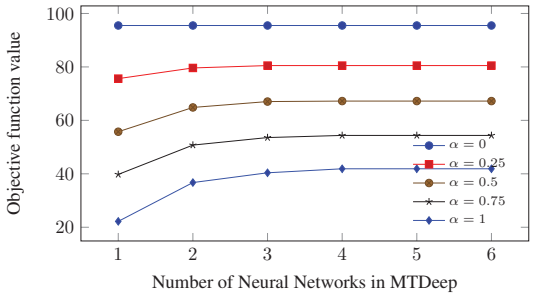


Figure 3: Accuracy of the MTDeep as the number of networks in the defender's configuration vary.

When the system receives only adversarial samples, i.e. $\alpha = 1$, the accuracy of MTDeep is 42% compared to 20% for most of the single DNN architectures. The strategy in this case is $\vec{x} = (0, 0.171, 0.241, 0, 0.401, 0.187)$ which does not use the configurations that have high vulnerability. If we used MTD-URS, it would switch to the vulnerable networks with equal probability, reducing the security of the ensemble. Although the 20% accuracy bump for modified images might not seem to be a drastic improvement, note that the individual networks have (i) high misclassification rates against Universal Perturbations designed for them (ii) present defense mechanisms (like adversarial training) are shown to be ineffective against Universal Perturbations and (iii) the differential immunity of our system is only 0.34 and there is no way of generating ensembles with high differential immunity and comparable classification accuracy.

We now explore the participation of individual networks in the equilibria for MTDeep in Figure 3. The results clearly show that while it is useful to have multiple networks providing differential immunity (as testified by the improvement of

	Adversarial User ($\mathcal{A}$)			
MTDeep	$FGSM_C$	$FGSM_M$	$FGSM_H$	$CWL2$
CNN	94.2, 5.8	97.8, 2.2	97.6, 2.4	80.0, 20.0
MLP	96.0, 4.0	87.0, 13.0	63.2, 36.8	90.0, 10.0
HRNN	95.9, 4.1	87.9, 12.1	93.2, 6.8	60.0, 40.0

Table 5: Table showing utilities for the defender and the User types when for classification of MNIST images with adversarially trained networks.

accuracy in adversarial conditions), the leveling-off of the objective values with more DNNs in the mix does underline that there is much room for research in actively developing DNNs that can provide greater *differential immunity*. An ensemble of such networks equipped with MTD can provide significant gains in both security and accuracy.

MTDeep with Adversarially Trained Networks

Adversarial training has emerged to be an effective defense mechanism against a multitude of different attacks. For this process, an attack algorithm is used to generate perturbed images, which are used (with their correct labels) in the training phase of the network. Unfortunately, the adversarially trained nets are only immune to perturbed imaged generated by this algorithm and may still be vulnerable to other (more expensive) adversarial manipulations like (Carlini and Wagner 2017). Even though one might think of adversarially training a network using images generated by all attack algorithms invented till date, it is (i) an expensive process and (ii) does not provide guarantees that a new algorithm will not render the network vulnerable.

We now show how MTDeep can be used in conjunction with adversarially trained neural network configurations to reduce misclassification rates. We adversarially train (with perturbations generated by the $FGSM$ method) the three networks described before– CNN, MLP and HRNN– on the MNIST dataset. The adversary not only has the FGSM attacks against which the adversarial training provides some immunity, but also a new iterative attack devised using (Carlini and Wagner 2017) that finds robust attacks against existing defenses. The matrix for game between MTDeep and $\mathcal{A}$ is shown in Table 5. We generate the misclassificaion rates on the entire test data for the $FGSM$ attacks and on 20 test data points using 1000 iterations for the new attack developed in (Carlini and Wagner 2017) ($CWL2$).

In Figure 4, we plot the objective function values for the MTDeep systems *vs.* its constituent networks as the value of α varies from 0 to 1. As mentioned earlier in the contributions, in this case, *MTD-URS is worse than having a classification system with a single network (CNN)*. When chances of attack is high (i.e. $\alpha = 1$), the misclassification rate of MTDeep is about 24% whereas, that of CNN, the most robust among the constituent networks, is 20%.

In the worst case, MTDeep uses a single network. Thus, it can never be insecure that than any of its constituent network. Thus, MTDeep can be used to bootstrap existing defense mechanisms of Neural Networks and provides higher robustness to adversarially manipulated images.

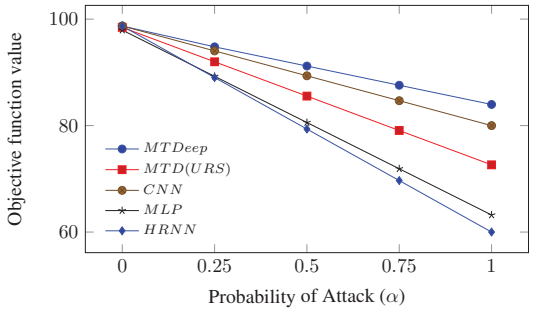


Figure 4: Objective function values on Adversarially Trained Networks for MNIST as α goes from 0 to 1.

Conclusion and Future Work

In this paper, we introduced MTDeep, a framework inspired by Moving Target Defense (MTD) in cyber security, as 'security-as-a-service' to help boost the security of existing classification systems based on Deep Neural Networks (DNNs). We defined the concept of differential immunity for an ensemble of networks, exploiting it to design switching strategies for MTDeep. In essence, we construct a Repeated Bayesian Game for capturing the interaction between MTDeep and its users, showing that the Stackelberg equilibrium of our game provides the optimal switching strategy for MTDeep that tries to reduce the misclassification on adversarially modified images while maintaining high classification accuracy for the legitimate users of the system. We show that this formulation is necessary when using MTDeep because naive switching strategies with an ensemble can lead to more vulnerable systems than using single networks. We empirically show the effectiveness of MTDeep against selected attacks for the MNIST and Imagenet datasets. Lastly, we demonstrate that using MTDeep with existing defense mechanisms for DNNs provide higher security guarantees those using the defense mechanisms by themselves.

In our experiments, the differential immunity of the ensemble of networks turns out to be the limiting factor in increasing the robustness of MTDeep against adversarial attacks. Thus, this work not only demonstrates the relevance of MTD in the context of transferability of attacks, but also brings to the table the open research problem of developing networks with high differential immunity. Given that MTD boosts the security of DNN based systems, we are exploring directions to incorporate the differential immunity measure into a trainable loss function such that we can train ensembles that have high differential immunity.

Acknowledgments This research is supported in part by the NASA grant NNX17AD06G and the ONR grants N00014161-2892, N00014-13-1-0176, N00014-13-1-0519, N00014-15-1-2027. The second author is also supported in part by the IBM Ph.D. Fellowship 2017-18.

References

Abbasi, M., and Gagné, C. 2017. Robustness to adversarial examples through an ensemble of specialists.

arXiv:1702.06856.

Bhagoji, A. N.; Cullina, D.; and Mittal, P. 2017. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *CoRR* abs/1704.02654.

Biggio, B.; Fumera, G.; and Roli, F. 2008. Adversarial pattern classification using multiple classifiers and randomisation. *Structural, Syntactic, and Statistical Pattern Recognition* 500–509.

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

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv:1405.3531*.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *arXiv preprint arXiv:1708.03999*.

De La Escalera, A.; Moreno, L. E.; Salichs, M. A.; and Armingol, J. M. 1997. Road traffic sign detection and classification. *IEEE transactions on industrial electronics* 44(6):848–859.

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

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*.

Javed, O., and Shah, M. 2006. Tracking and object classification for automated surveillance. *Computer VisionECCV 2002* 439–443.

Jayadevan, R.; Kolhe, S. R.; Patil, P. M.; and Pal, U. 2012. Automatic processing of handwritten bank cheque images: a survey. *International Journal on Document Analysis and Recognition (IJDAR)* 15(4):267–296.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, 675–678. ACM.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2016. Universal adversarial perturbations. *arXiv:1610.08401*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2574–2582.

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

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

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the Conference on Computer and Communications Security*, 506–519. ACM.

Paruchuri, P.; Pearce, J. P.; Marecki, J.; Tambe, M.; Ordonez, F.; and Kraus, S. 2008. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, 895–902. International Foundation for Autonomous Agents and Multiagent Systems.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.

Sengupta, S.; Vadlamudi, S. G.; Kambhampati, S.; Doupé, A.; Zhao, Z.; Taguinod, M.; and Ahn, G.-J. 2017. A game theoretic approach to strategy generation for moving target defense in web applications. AAMAS.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the SIGSAC Conference on Computer and Communications Security*, 1528–1540. ACM.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

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

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.

Taguinod, M.; Doupé, A.; Zhao, Z.; and Ahn, G.-J. 2015. Toward a Moving Target Defense for Web Applications. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI)*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Zheng, S.; Song, Y.; Leung, T.; and Goodfellow, I. 2016. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4480–4488.

Zhuang, R.; DeLoach, S. A.; and Ou, X. 2014. Towards a theory of moving target defense. In *Proceedings of the First ACM Workshop on Moving Target Defense*, 31–40. ACM.