A Moving Target Defense against Adversarial Machine Learning

[1]Abhishek Roy
[2]Anshuman Chhabra
[3]Charles A. Kamhoua
[2]Prasant Mohapatra
abroy,chhabra@ucdavis.edu
charles.a.kamhoua.civ@mail.mil
pmohapatra@ucdavis.edu
[1]Department of Electrical and Computer Engineering, University of California, Davis
[2]Department of Computer Science, University of California, Davis
[3]Network Security Branch, U.S. Army Research Laboratory (ARL)

ABSTRACT

Adversarial Machine Learning has become the latest threat with the ubiquitous presence of machine learning. In this paper we propose a Moving Target Defense approach to defend against adversarial machine learning, i.e., instead of manipulating the machine learning algorithms, we suggest a switching scheme among machine learning algorithms to defend against adversarial attack. We model the problem as a Stackelberg game between the attacker and the defender. We propose a switching strategy which is the Stackelberg equilibrium of the game. We test our method against rational, and boundedly rational attackers. We show that designing a method against a rational attacker is enough in most scenarios. We show that even under very harsh constraints, e.g., no attack-cost, and availability of attacks which can bring down the accuracy to 0, it is possible to achieve reasonable accuracy in the context of classification. This work shows, that in addition to switching among algorithms, one can think of introducing randomness in tuning parameters, and model choices to achieve better defense against adversarial machine learning.

KEYWORDS

Adversarial Machine Learning, Moving Target Defense, Bounded Rationality, Cybersecurity

ACM Reference Format:
[1]Abhishek Roy, [2]Anshuman Chhabra, [3]Charles A. Kamhoua, and [2]Prasant Mohapatra. 2019. A Moving Target Defense against Adversarial Machine Learning. In *Proceedings of Second ACM/IEEE Workshop on Security and Privacy in Edge Computing (EdgeS&P '19)*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

It has been shown in [1] that an adversarial attack designed against a classification algorithm transfers well to other classification algorithms. Even an ensemble of multiple algorithms is not safe due to transferability of attack from one algorithm to another. We know that, in case of a white-box attack, i.e., if the attacker has full knowledge and access to the neural network being used, given enough time, highly effective adversarial attacks can be designed [2]. Even in a back-box environment where the attacker can only observe the output generated by a neural network for a given input but has no control over the true deep neural network (DNN), very notorious adversarial examples can be generated. It has been shows that accuracy can be brought down to 10% by a well-designed black-box attack [3]-[4]. Ref. [5] shows that if the attacker has limited attack budget, one can introduce random noise in DNN models to break the transferability of attacks without compromising accuracy. If the transferability is low the defender may benefit from changing the algorithm intermittently if we assume that the attacker does not know the exact algorithm being implemented instantly. This is a valid assumption as the attacker needs time to design adversarial examples. Switching among algorithms may fail if algorithms with high enough accuracy are unavailable. Though now-a-days normally we have lot of highly efficient algorithms for common machine learning jobs, e.g., classification, clustering, and prediction. Our main goal is to develop an efficient defense strategy against transferable adversarial attacks by switching among algorithms of various accuracy.

In this paper we model the interaction between the attacker, and the defender as a Stackelberg game where the defender is the leader. The defender declares the probability vector of using different algorithms next. The attacker reacts accordingly. We consider a general setting where transferability of different attacks, and accuracy of the available algorithms can assume any possible value. We allow the attacker to be potent enough to bring down the accuracy to zero if he knows exactly which algorithm is being deployed. This is not necessarily the case all the time, but it is useful to assume this as it is the worst case from defender's perspective. Defender incurs cost while switching among algorithms. If this cost is high, switching often among algorithms may not be beneficial for the defender. The objective of this work is to optimally decide the moving strategy taking the above trade-off into account.

EdgeS&P '19, November 07–09, 2019, Washington DC

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

Table 1: Notation Meaning

Notation	Meaning
$A = \{A_i\}_{i=1}^K$	Set of available algorithms
$B = \{B_i\}_{i=1}^K$	Set of attacks corresponding to the algorithms
a_i	Accuracy of algorithm A_i
$V_{d(att)}(a)$	Value of accuracy a to the defender (attacker)
C_M	Defender's cost of transition from A_i to A_j $(i \neq j)$
$T = \left[\tau_{ij}\right]_{K..K}$	Transferability matrix
$a_{i\vert j}$	Accuracy of A_i when attacked by B_j
$U_d(\pi)$	Defender's expected utility with transition probability
$U_{att}(B_i)$	Attacker's utility of B_i
$\pi = [\pi_i]_K$	Vector of probabilities of algorithms going to be deployed next
δ, γ	Bounded rationality model parameters
$w\,(p)$	Probability weighting function

We find the optimal strategies for players and characterize the optimal solutions under certain structures of the game. We show on real datasets how we can mislead attacker using our approach and achieve higher effective efficiency.

In most of the game theoretic study, the players are assumed to be rational. But human beings rarely are *rational*. A defense method designed against a rational attacker may not work well against an attacker which does not confirm to the sense of rationality as posed by the defender. In this work we attempt to answer the question of how detrimental a *boundedly rational* attacker can be to the defender who is using a defense mechanism designed against a rational attacker. To the best of our knowledge, this is the first work to show how switching among machine learning algorithms can be used to defend against a boundedly rational adversary. Note that our approach is similar to [6] but we extend the technique to boundedly rational attackers.

The rest of the paper is organized as follows: Section 2 contains the details of the game where we derive optimal strategies for players. Section 3 presents different bounded rationality models. Section 4 presents a toy example illustrating our approach. In Section IV contains the simulation results, and performance results of our method on a real dataset. Section 5 concludes the paper.

2 DETAILS OF THE GAME

2.1 Preliminaries

We model the moving target defense (MTD) as a Stackelberg game with the defender as the leader and the attacker as the follower. First, the defender declares the vector π containing probabilities of using different algorithms. Observing the probability, the attacker decides on attacking. Given the accuracy a of an algorithm, we denote the defender's (attacker's) payoff by $V_{d(att)}\,(a)$.

Assumption 1. We assume that the defender's (attacker's) payoff is an increasing (decreasing) function of accuracy, i.e., $\frac{dV_d}{da} \geq 0 \left(\frac{dV_{att}}{da} \leq 0\right)$.

Assumption 2. Every algorithm A_i has a corresponding best attack B_i which brings down the accuracy of A_i to 0.

If $a_{i|j}$ is the accuracy of algorithm A_i when attacked by B_j, assumption 2 implies, $a_{i|i} = 0$. Assumption 2 is rational as this assumption represents the worst case for the defender.

It has been observed that an attack which is designed against a particular algorithm may work well against another algorithm. We introduce a performance metric called transferability to measure the damage caused to the performance of one algorithm when attacked by an attack designed for another algorithm.

Definition 1: Transferability τ_{ij} of attack B_j to algorithm A_i is defined as $\tau_{ij} = \frac{a_{i|j}}{a_i}$.

The lower the value of τ_{ij} the better is the transferability. Note that $0 \leq \tau_{ij} \leq 1$. Let $\pi = (\pi_1,\ \pi_2,\ \cdots\ ,\pi_K)$ be the probability of using different algorithms as declared by the defender. The attackers utility is given by

$$U_{att}\,(B_i, \pi) = \pi_i V_a\,(0) + \sum_{j \neq i} \pi_j V_{att}\,\left(\tau_{ji} a_j\right) \tag{1}$$

The action taken by the attacker on observing π is given by

$$B_*\,(\pi) = argmax_{B_i} U_{att}\,(B_i, \pi) \tag{2}$$

If there B_* is not unique, without loss of generality, we assume that the attacker chooses the attack with smaller index. Assuming the attacker is rational, and hence going to follow the above attack method, the defender has to maximize its utility given as

$$U_d\,(\pi) = \pi_1 C_M + \sum_j \pi_j \left[V_d\,\left(\tau_{j*} a_j\right) - C_M\right] \tag{3}$$

The defender chooses π^* such that,

$$\begin{aligned} &\pi^* = argmax_\pi U_d\,(\pi) \\ &\sum_{i=1}^K \pi_i = 1 \\ &\pi_i \geq 0 \qquad \forall i = 1, 2, \cdots, K \end{aligned} \tag{4}$$

The solution to the above optimization problem constitutes the Stackelberg equilibrium of the game. The feasible region of the above optimization problem is the $(K-1)$-simplex. Depending on B_* the coefficients of the objective function changes. For a fixed B_* the optimization is a Linear Program (LP) on $(K-1)$-simplex which can be solved in polynomial time in the number of variables [10]. There are K distinct possible values of B_*; hence, the optimization can be solved in polynomial of K time. Next, we are going to consider more structured forms of this game to gain more insight into how the optimal strategies vary with different parameters.

We will specifically consider two extreme cases mentioned before in the introduction: firstly, the case where the transferability of attacks among algorithms is constant but algorithms have different accuracies; secondly, the case where the attacks have different transferability, but all the algorithms achieve a fixed accuracy. For this work, we assume $V_{att}\,(x) = 1 - x$, and $V_d\,(x) = x$. Moreover, without loss of generality, we assume that the defender is using A_1 before the beginning of the game, and this is known to the attacker. In the first of the above mentioned cases $\tau_{ij} = \tau_j$ when $i \neq j$, and $\tau_{ii} = 0$. Under these assumptions, $U_{att}\,(B_i, \pi) = 1 - a\tau_i + a\tau_i\pi_i$. So the attacker uses attack i with minimum $\tau_i\,(1 - \pi_i)$ value. $U_d\,(\pi) = \pi_1 C_M +$

$\sum_j \pi_j \left[\tau_* a - C_M\right] = a\tau_* \left(1 - \pi_*\right) - C_M \left(1 - \pi_1\right)$. Let, $\pi_i' = \frac{1-\pi}{K-1}$. Then the attacker uses the attack with minimum $\tau_i \pi_i'$, and the defender's utility changes to $U_d\left(\pi\right) = \left(K-1\right)\left(a\tau_*\pi_*' - C_M\pi_1'\right)$. For better exposition we assume $\tau_{ij} = \tau_j$ when $i \neq j$, and $\tau_{ii} = 0$. These assumptions lead to $U_{att}\left(B_i, \pi\right) = 1 - \tau \sum_j \pi_j a_j + \tau\pi_i a_i$, and $U_d(\pi) = \pi_1 C_M + \sum_j \pi_j \left[\tau_(j*) a_j - C_M\right] = \tau \sum_j \pi_j a_j - C_M \left(1 - \pi_1\right) - \tau\pi_* a_*$. So for a given transition probability vector and $\left(a_1,\ a_2,\ \cdots,\ a_K\right)$, the attacker attacks the algorithms with maximum $\pi_i a_i$.

2.2 Characterization of the Solution

We are trying to find the solution to the above optimization on the probability simplex. We will characterize the solution in case of $K = 3$ for better exposition. The characterization is similar for any general K.

Note that maximizing $U_d\left(\pi\right)$ can be thought of finding the maximum of the optimal points of the following optimization problems on probability simplex : for each $i = 1, 2, 3$,

$$\max_{\pi} \tau \sum_{j \neq i} \pi_j a_j - C_M \left(1 - \pi_1\right) \tag{5}$$

$$\pi_i a_i \geq \pi_j a_j \qquad \forall j = 1, 2, 3. \tag{6}$$

As each of the above optimization problems is a linear program, the optimal solution will be at one of the vertices of the feasible regions. We enlist few properties of the solution of this problem in a case where τ is constant. Very similar conclusions can be made for a scenario where transferabilities vary with algorithms but the accuracy is same across algorithms.

(1) If $C_M \geq \max\left(\tau a_2, \tau a_3\right)$, the best strategy of the defender is not to change the current algorithm.
Let us assume w.l.o.g. $a_3 \geq a_2$.

(2) When $a_2 \leq \frac{a_3}{1+a_3}$, if $a_1 \geq \frac{a_2}{1-\frac{a_2}{a_3}}$, for $0 \leq C_M \leq \tau a_3$, the optimal solution is $\left(\frac{a_3}{a_1+a_3}, 0, \frac{a_1}{a_1+a_3}\right)$.

(3) If $\frac{1}{a_1} \geq \frac{1}{a_2} + \frac{1}{a_3}$, and $C_M \leq \tau\left[\frac{a_2 a_3}{a_2+a_3} - a_1\right]$, then the optimal solution is $\left(0, \frac{a_3}{a_2+a_3}, \frac{a_2}{a_2+a_3}\right)$.

(4) In all other scenarios the solution will be $\left(\frac{a_2 a_3}{a_1 a_2+a_2 a_3+a_3 a_1}, \frac{a_3 a_1}{a_1 a_2+a_2 a_3+a_3 a_1}, \frac{a_1 a_2}{a_1 a_2+a_2 a_3+a_3 a_1}\right)$.

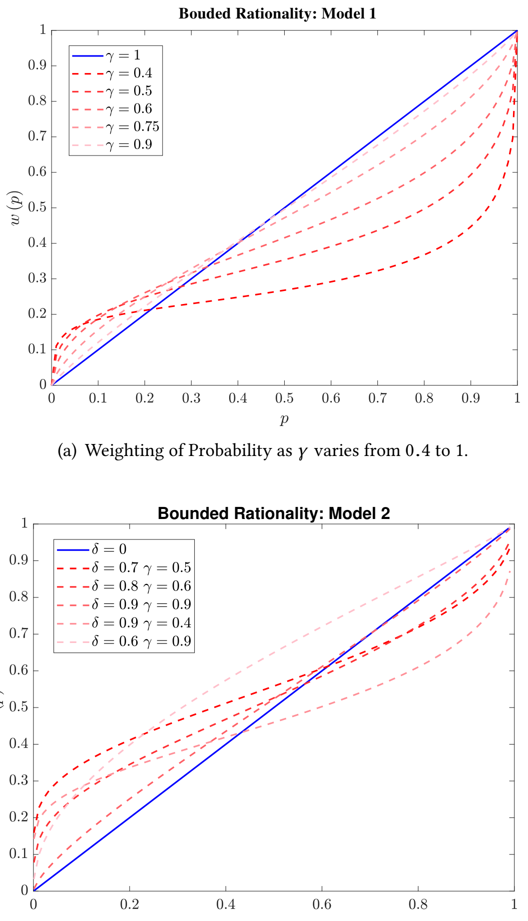
3 BOUNDEDLY RATIONAL ATTACKER

In this section we discuss the effects of a boundedly rational attacker. It has been well studied that human begins do not tend to interpret probability values as it is [7],[8]. To be precise, they tend to overvalue lower probabilities and undervalue higher probabilities as shown in Fig. 1. The interpretation of probability also depends on how inclined an individual is towards gambling. We denote the interpreted value of the probability as $w\left(p\right)$. We consider two well-known bounded rationality models given by[7]:

$$\text{Model 1:} \qquad w\left(p\right) = \frac{p^{\gamma}}{\left(p^{\gamma} + \left(1-p\right)^{\gamma}\right)^{\frac{1}{\gamma}}} \tag{7}$$

$$\text{Model 2:} \qquad w\left(p\right) = \exp\left(-\delta\left(-\log p\right)^{\gamma}\right) \tag{8}$$

where δ, and γ are model parameters. Figure 1(a) shows that as γ increases the attacker becomes more rational. Figure 1(a) shows that δ indicates how optimistic the attacker is, and γ indicates the curvature of the weighting function. We will see the effects of bounded rationality in the next section.



(a) Weighting of Probability as γ varies from 0.4 to 1.

(b) Weighting of Probability for different choices of δ, and γ

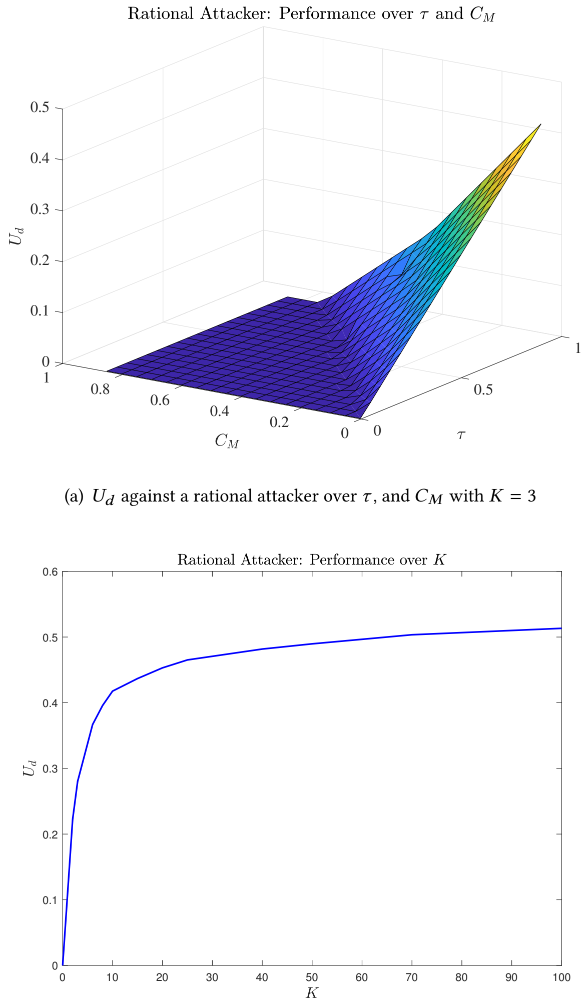
Figure 1: Bounded Rationality Models

4 RESULTS

4.1 Simulation Results

In all the following simulations we have chosen the accuracies of the algorithms to be randomly between 0.6 and 1.

4.1.1 Rational Attacker. In this section we show how the defender performance varies with moving cost, and transferability of attacks. Figure 2 shows that for low C_M and high τ, using the MTD approach can ensure an accuracy of up to 0.5 which is about 65% of the average algorithm accuracy of 0.8 instead of 0 accuracy. Note that, these results are obtained in an extreme scenario where the attacker does not incur any cost which is unrealistic. In reality, due to cost, if the effective utility is non-positive the attacker will abort the attack resulting in a much higher effective accuracy for the defender. Figure 2(b) shows an interesting trend. Adding more algorithms to



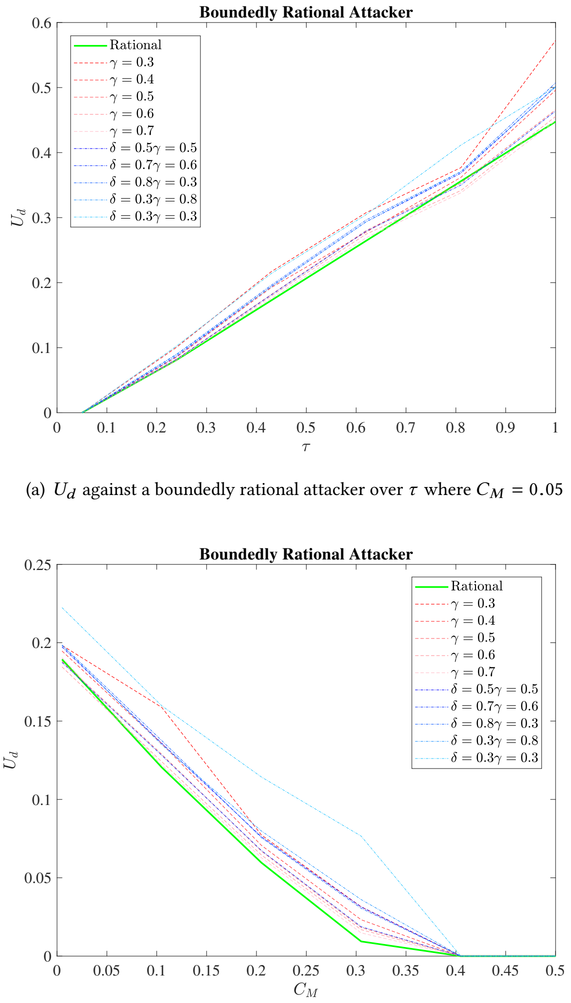
(a) U_d against a rational attacker over τ, and C_M with $K = 3$

(b) U_d against a rational attacker over K with $\tau = 0.6$, and $c_M = 0.03$

Figure 2: Performance of the defender against a rational attacker under different model parameters

the arsenal has a diminishing return property in terms of accuracy. So having a few algorithm can be enough to achieve any meaningful accuracy.

4.1.2 Boundedly Rational Attacker. In this section we discuss how a boundedly rational attacker can be more beneficial to the defender. Figure 3(a) shows that if the attacker is boundedly rational then the defender obtains higher effective accuracy for most values of τ. Same behavior is observed over the range of C_M. As one can see from the figure that the accuracy can be upto 25% more if the attacker is not fully rational. This implies that designing defense strategies assuming the attacker is rational works best for the defender in most situation. We leave the quantification of the gain of the defender as a function of *boundedness* of rationality of attacker to future work.



(a) U_d against a boundedly rational attacker over τ where $C_M = 0.05$

(b) U_d against a boundedly rational attacker over C_M where $\tau = 0.4$

Figure 3: Performance of the defender against different boundedly rational attacker under different model parameters

4.2 Real Dataset Results

To evaluate how MTD performs against real-world attacks on machine learning algorithms, we implement the following attack algorithms: Carlini and Wagner (CW) attack [2] against a Convolutional Neural Network (CNN), Fast Gradient Sign Method [9] against a Logistic Regression classifier, and the adversarial Support Vector Machine (SVM) attack as described in [1]. All the classifiers are trained on the MNIST dataset [10], and the CNN model is based on the AlexNet architecture [11].

The CW attack used on the CNN architecture is an iterative L_2 attack that uses gradient descent to solve the following minimization problem:

$$minimize : \|x - x'\| + cl(x')$$

Table 2: Transferability Results

SOURCE/TARGET	Multi-class SVM	Logistic Regression	CNN
Multi-class SVM	1.0	0.12	0.006
Logistic Regression	0.93	1.0	0.856
CNN	0.42	0.284	1.0

where $l(x') := \max(\max\{Z(x')_{i\neq t}\} - Z(x')_t, \kappa)$, Z are the logits of the neural network, and c, κ are parameters.

The FGSM attack uses the gradient of the cross-entropy loss obtained from the Logistic Regression classifier (denoted as $C(x)$) to compute an optimal yet small perturbation $\rho(x)$ such that $C(x + \rho(x)) \neq C(x)$. Moreover, the perturbation is computed as follows:

$$\rho(x) = \epsilon.sign\left(\nabla_x L(x, C(x))\right)$$

where ϵ is a threshold parameter, and $L(x, C(x))$ is the loss function used to train the classifier on.

Finally, the SVM attack is a simple adversarial attack that crafts a perturbed sample for the one-vs-the-rest multi-class SVM classifier by moving the original input (x) orthogonal to the decision boundary hyperplane. Thus the perturbed sample (x') can be denoted by $x' = x - \epsilon \frac{w_k}{\|w_k\|}$, where w_k is the weight vector normal to the hyperplane for the k^{th} binary SVM making up the multi-class SVM, and ϵ is a threshold parameter.

We implement all the attacks and classifiers in Python, and 100 adversarial images for each source attack on the MNIST dataset. The accuracy values obtained on the test-set for each classifier are: Logistic Regression: 0.8975, Support Vector Machine: 0.9156, and for Convolutional Neural Network: 0.9888. We use each of the 100 generated images for each source, and attack the other target classifiers. The obtained transferability values averaged over 5 runs are shown in Table 2. Figure 4 shows the performance of our method in this dataset. It can be seen here that boundedly rational attackers are not much of a threat compared to a rational attacker. We observe that when the cost is low, we can achieve accuracy as high as 0.5 instead of 0. For boundedly rational attackers the gain is high even if the moving cost is large. As we consider the worst case possible, that the adversary has no cost, never aborts, and the accuracy decreases to 0 when attacked by a targetted algorithm, this result is promising.

5 CONCLUSION

In this paper we propose a Moving Target Defense approach to defend against adversarial machine learning. We test our method against rational, and boundedly rational attackers. We show that designing a method against a rational attacker is enough in most scenarios. We show that even under very harsh constraints, e.g., no attack-cost, and availability of attacks which can bring down the accuracy to 0, it is possible to achieve reasonable accuracy for classification. In future we plan to extend this work to other varieties of machine learning tasks. This also shows, that in addition to switching among algorithms, one can think of introducing randomness in tuning parameters, and model choices to achieve better defense against adversarial machine learning.

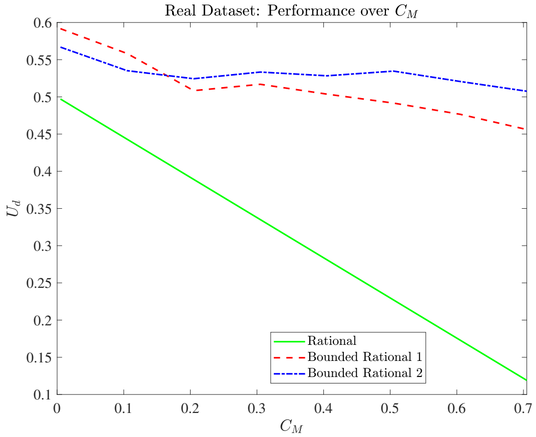


Figure 4: The performance of MTD against optimally designed attack against classification on real dataset

REFERENCES

[1] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[3] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.

[4] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017.

[5] Yan Zhou, Murat Kantarcioglu, and Bowei Xi. Breaking transferability of adversarial samples with randomness. *arXiv preprint arXiv:1805.04613*, 2018.

[6] Sailik Sengupta, Tathagata Chakraborti, and Subbarao Kambhampati. Mtdeep: boosting the security of deep neural nets against adversarial attacks with moving target defense. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] Richard Gonzalez and George Wu. On the shape of the probability weighting function. *Cognitive psychology*, 38(1):129–166, 1999.

[8] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[10] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.