Adversarial Machine Learning for Network Security

Yi Shi
Intelligent Automation, Inc.
Rockville, MD, USA
yshi@i-a-i.com

Hui Zeng
Intelligent Automation, Inc.
Rockville, MD, USA
hzeng@i-a-i.com

Tung T. Nguyen
Intelligent Automation, Inc.
Rockville, MD, USA
tnguyen@i-a-i.com

***Abstract*—With the rapid growth of machine learning applications in communication networks, it is essential to understand the security issues associated with machine learning. In this paper, we choose a flow-based Deep Neural Network (DNN) classifier as a target and study various attacks on this target classifier. The target classifier detects malicious HTTP traffic (i.e., bots, C&C, etc.). We first launch an exploratory attack under a black box assumption against the target DNN classifier. We start from a simple case that the attacker can collect the same set of features used in the target classifier and then consider the case that the attacker can only collect a set of features based on its judgement. We also design the attacks with conditional Generative Adversarial Network (cGAN) to reduce the requirement on the amount of collected data. We show that the attacker can build its own classifier to predict the target classifier's classification results with about 93% accuracy. Once the exploratory attack is successful, we can perform further attacks, e.g., evasion attack and causative attack. We show that these attacks are very effective. Evasion attack can identify samples to double error probability of the target classifier while under causative attack, the new classifier makes classification errors on more than 60% of samples.**

Keywords—Machine learning; network security; exploratory attack; evasion attack; causative attack; conditional GAN.

I. INTRODUCTION

Recent research [1]-[4] has demonstrated that machine learning (ML) models are often vulnerable to information leakage as well as adversarial manipulation of their input. However, current efforts are mainly focused on the image, sensor and text domains, and many deal with simple attacks such as model inversion attacks. The topic of vulnerabilities in ML system utilized in the cyber defense domain has not been sufficiently explored. The users of ML systems in other domains, such as recommendation systems, in general do not have much incentive or opportunity to mislead the system. In the cyber defense domain, attackers and defenders each improve their tools in response to each other devising new techniques. Hence, compared to the counter-parts in other domains, the attacks to ML systems in the cyber defense domain are more complicated, dynamic and associated with higher cost.

In this paper, we design an adversarial machine learning toolset in the cyber defense domain, including the evaluation of ML concepts, methods, attack methodology, and the impact in the cyber security. The results of this effort will aid in better understanding of machine learning based network security solutions as well as enhanced penetration testing and cyber defensive techniques. Figure 1 shows the schematic of the adversarial ML toolset.

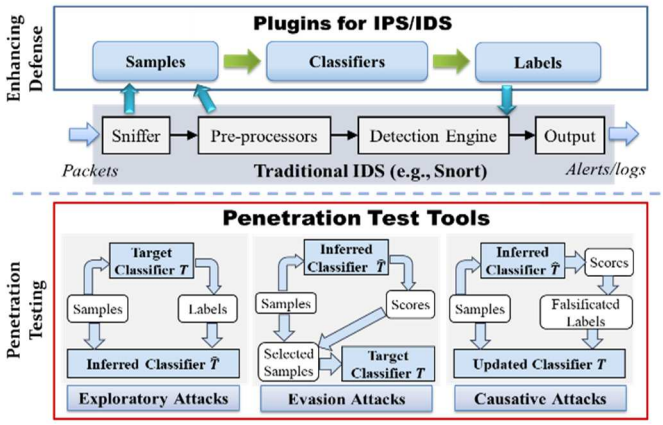

Figure 1: Schematic of our Adversarial ML tool

The key components can be categorized into Plugins for Intrusion Prevention/Detection System (IPS/IDS, i.e., the **defense** side) shown in the blue box on the

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-1-7281-5092-5/19/$31.00 ©2019 IEEE

top of diagram, and Penetration Test tools (i.e., the **assessment** side) in the red box on the bottom of diagram.

Currently we focus on the assessment side, which analyzes the attack surfaces and potential vulnerabilities of the ML-dependent cyber security systems, by tapping into the attacker's viewpoint to spear the shield of the defense. To counter the attack vectors discovered from the assessment, the impact will be carefully evaluated, and the countermeasures will be devised and incorporated into the evolving update of the model or design in the defense side. Specifically, at the **defense** side, the traditional IPS/IDS (e.g., Snort) sniffs the network traffic, identifies packets/flow "behavior", compares each packet/flow against a predefined ruleset (or statistics/ patterns), and provides the alerts and/or log as the output. The ML based plugins to the traditional IPS/IDS will enhance its anomaly detection and security analysis functions. Many of these functions can each be modelled as a classifier T, which accepts features provided by the data collection module as input, performs analysis based on machine learning algorithms, and generates labels (e.g., normal or malicious) for each sample based on the pre-trained or continuously updated model. For the model training and evaluation, we can use public datasets, such as the CTU-13 dataset [5][6], CIC IDS 2018 [7], ISCX IDS 2012 [8], DEFCON [9], etc., and synthetic data generated based on the above datasets.

The **assessment** side contains a set of penetration test (pentest) tools to examine the ML related vulnerabilities of the defense side, including the ML concepts, methods, and data. Three types of attacks are shown at the bottom of Figure 1, including:

- ***Exploratory attack*** [3][14][17][18][19] aims to gain the knowledge of target classifier T. We started from a *black box* attack assumption, i.e., the adversary has no prior information on T. The attacker, however, is assumed to be able to observe the input (samples) and infer the output (labels) of T. Note that although samples can be observed, the attacker does not know which features are used to describe a sample and thus determine its own set of features, which may not be the same as that used by T. Over a learning period, an attacker can collect a set of (sample, label) as its training data and apply deep learning (DL) to build a functionally equivalent classifier $\hat{T}$. Using generative adversarial network (GAN) [10] or conditional GAN (cGAN) [12], an attacker can build $\hat{T}$ with a short learning period and improve it over time. Once $\hat{T}$ is obtained, an attacker can predict the label generated by T for any sample. An attacker can analyze the vulnerability of T and then perform other attacks more effectively, such as evasion attack and causative attack (which do not have to rely on the exploratory attack, though).
- ***Evasion attack*** [15][16][19] aims to discover samples that T may mis-classify. For example, if T is used to find malicious users, an attacker can perform evasion attack such that T cannot detect it as a malicious user. Evasion attack is done by analyzing the output of DL algorithm in $\hat{T}$, which provides not only labels, but also the confidence of classification. An attacker will select samples with low classification confidence such that the error probability of T on selected samples is high. We developed this attack and examined the error probability of T under evasion attack to evaluate the vulnerability of T.
- ***Causative attack*** [20] alters the training data to influence T. We started with the causative attack on the retraining process. The adversary is assumed to be able to access the training samples, manipulate (some of) the samples, and observe or infer the output (i.e., labels) of T. For any sample, the adversary can either provide a falsificated label *directly* (i.e., alter its label in the training data) or *indirectly* (i.e., alter the sample to mislead the target classifier). To avoid being detected, an attacker should not falsificate all data. We again studied the output of DL algorithm in $\hat{T}$. An attacker will change the sample or the label only for those data with high classification confidence to maximize the impact on T. We examined the error probability of T under causative attack to evaluate the vulnerability of T.

The rest of the paper is organized as follows. Section II describes the example target classifier. Section III demonstrates how to apply deep learning to launch the exploratory attack. Section IV shows the application of GAN and cGAN to generate synthetic data for enhancing the attack. Sections V and VI develops the evasion attack and the causative attack, respectively. Section VII concludes the paper.

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

II. TARGET CLASSIFIER

We chose a flow-based Deep Neural Network (DNN) classifier developed in our previous effort as the target classifier T, which detects malicious HTTP traffic (i.e., bots, C&C, etc.). This DNN classifier was built and evaluated mainly on the CTU-13 dataset [6] in Stratosphere [5]. To increase the size of normal data for our study, we also used another dataset, the "Hands-on Network Forensics" dataset from the 2015 FIRST conference [11].

Among many flow features, seven were selected for the training of this target classifier:

- *Largest time interval bin (MaxBin)*: The time intervals of the consecutive flows are grouped in the flow window into 1-minute bins. It represents the bin that contains the largest number of flows.
- *Ratio of the largest time interval bin (MaxBinRatio)*: The percentage of the flows that fall into MaxBin.
- *Average time interval (AvgInterval)*: The average time interval in seconds.
- *Average flow duration (AvgDuration)*: The average flow duration in seconds.
- *Standard deviation of flow durations (StdDuration)*: The standard deviation of the flow durations in seconds.
- *Average source data amount (AvgSrcBytes)*: The data amount transferred from the source to the destination in bytes.
- *Average destination data amount (AvgDstBytes)*: The average data transferred to the destination in bytes.

Our DNN classifier was constructed with the following: Two hidden layers, each of which has 64 neurons with the rectified linear unit (ReLU) as activation function; An output layer with 2 neurons and "softmax" as activation function; Categorical cross-entropy as loss function; Adaptive Moment Estimation (ADAM) as the optimizer with learning rate of 0.001. Dropout regularization technique was applied to prevent overfitting. In the evaluation, our DNN classifier achieves a training loss at around 3% and a categorical accuracy at around 98%.

Although details on how to build this target classifier is presented in this section, the attacks discussed in the following sections are in general black box attacks, i.e., they do not require any of the design details described in this section.

III. DL BASED EXPLORATORY ATTACK

We considered an exploratory attack on the target classifier described in Section II, under a black box attack assumption. Instead, the attacker can collect data from T, including flow data and the classification on each flow. Then the attacker will launch an exploratory attack based on deep learning to build a classifier $\hat{T}$ such that it can provide the same classification as T for most flow scenarios.

A. Data Collection

We started from a simple case that the attacker can collect the same set of features (i.e., the same seven features) used in T. We further assumed that the attacker can detect the classification results made by T. We first collected data for 1,040 "flows" (each as an aggregation of 50 flows), including 199 negative samples (i.e., normal flows) and 841 positive samples (i.e., malicious flows). Later on we collected additional 1,149 negative and 10 positive samples from CTU-13 dataset, which makes the total to be 2,189. In our case, we divided our data by half for training data and test data, respectively.

In a more challenging case, each sample can be described by many features and an attacker does not know which features are used by T. In this case, the attacker needs to identify a set of features based on its own knowledge. We considered a scenario that an attacker uses 9 features for each sample.

B. Performance Metrics

In reality, $\hat{T}$ and T may provide different class information for a flow and we want to minimize the probability of these flows. We can calculate two error probabilities for a given test data as follows.

- *Misdetection probability* e_{MD} that is the number of positive flows (by T) detected as negative by $\hat{T}$ divided by the number of positive flows.
- *False alarm probability* e_{FA} that is the number of negative flows (by T) detected as positive by $\hat{T}$ divided by the number of negative flows.

To take both misdetection and false alarms into consideration, we use $max\{e_{MD}, e_{FA}\}$ as the performance metric.

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

C. Deep Learning Algorithm

We applied deep learning algorithm to build classifier $\hat{T}$ (see Figure 2). We divided the collected data into training data and test data, and fed them to the DL algorithm. The DL algorithm tunes its network under a set of hyper-parameters based on the training data. In addition, hyper-parameters, such as number of layers, size of a layer, learning algorithm, and cost function, are tuned based on test data and the performance metric $max\{e_{MD}, e_{FA}\}$.

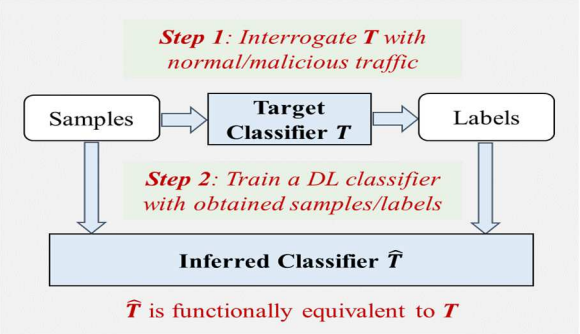

Figure 2: DL-based Black-Box Exploratory Attack

The trained classifier achieves: $e_{MD} = 8.70\%$, $e_{FA} = 4.72\%$, and $max\{e_{MD}, e_{FA}\} = 8.70\%$.

D. Different View at the Attacker

Now consider the case that the attacker collects a different set of 9 features, among which four features are used by the target classifier, including: *Largest time interval bin (MaxBin), Ratio of the largest time interval bin (MaxBinRatio), Average source data amount (AvgSrcBytes)*, and *Standard deviation of flow durations (StdDuration)*.

And five new features are listed below:

- *Standard deviation of time intervals (stdInterval)*: The standard deviation of time intervals in seconds.
- *Average total data amount (avgTotBytes)*: The data amount transferred from the source to the destination in bytes.
- *Standard deviation of total data amount (stdTotBytes)*: The standard deviation of the data amount transferred from the source to the destination, in bytes.
- *Standard deviation of source data amount (stdSrcBytes)*: The standard deviation of the data amount transferred from the source, in bytes.
- *Standard deviation of destination data amount (stdDstBytes)*: The standard deviation of data transferred to the destination, in bytes.

Table 1 lists the attack results in this case, compared with the case that the attacker knows the same set of features (i.e., 7 features) used to train the target classifier.

Table 1: Attack Results using Different Set of Features

	e_{MD}	e_{FA}	$\max\{e_{MD}, e_{FA}\}$
7 features	8.70%	4.72%	8.70%
9 features	5.19%	4.73%	5.19%

It is shown in Table 1 that even with a different set of features, the attacker can still launch an exploratory attack. In fact, with more features, the error probability is even reduced.

IV. SYNTHETIC DATA GENERATION USING GAN AND CGAN

One major challenge to DL-based exploratory attack is its need for a large amount of training data to build a good classifier (and testing data for performance evaluation). This also applies to further attacks such as evasion and causative attacks.

One effective approach to solve this problem is to use generative adversarial network (GAN) [10]. GAN includes a generator G and a discriminator D. Once GAN is trained, generator G can generate synthetic data very similar to real data. Figure 3 shows the DL based black-box exploratory attack using GAN.

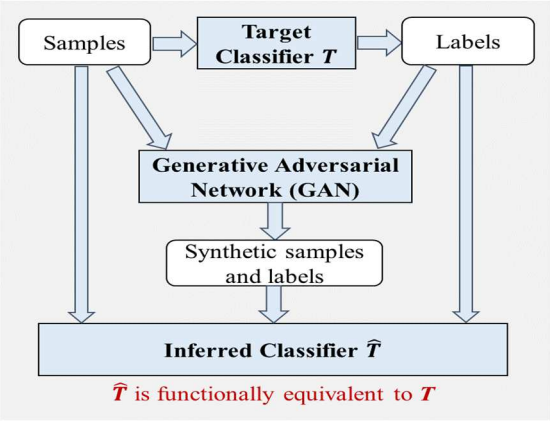

Figure 3: GAN-based Black-Box Exploratory Attack

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

In this section, we first applied a standard GAN to generate additional data and evaluated the difference between its generated data and the real data. We then designed and developed a conditional GAN (cGAN) [12] that generates data with the same distribution as that for training data, and evaluated its performance compared to the standard GAN.

The synthetic data can be used to improve various attacks. We use exploratory attack as an example. If we only use real data, our previous results show that $e_{MD} = 8.70\%$ and $e_{FA} = 4.72\%$ can be achieved, and the performance metric $max\{e_{MD}, e_{FA}\}$ is 8.70% (using 7 features). Table 2 shows the attack results using additional 500 synthetic data generated by either standard GAN or cGAN, respectively, as part of the training data to build the attacker's classifier $\hat{T}$, and then half of the original 2,199 real data for testing. We can see that both standard and conditional GAN can reduce the performance metric to 6.78%. Conditional GAN has better performance on false alarm probability than standard GAN, and is more realistic (i.e., feasible) and stealthy (i.e., usually in the normal behavior zone).

Table 2: Exploratory Attack Results with Synthetic Data

	e_{MD}	e_{FA}	$max\{e_{MD}, e_{FA}\}$
Without GAN	8.70%	4.72%	8.70%
Standard GAN	6.78%	4.67%	6.78%
cGAN	6.78%	3.74%	6.78%

V. EVASION ATTACK

Besides exploratory attack, there are another two adversarial machine learning attacks, namely evasion attack and causative attack [15]. Evasion attack aims to discover samples that target classifier T may mis-classify and causative attack alters the (re-)training data to influence T. Although we again consider black-box attacks, i.e., the adversary does not know T and thus cannot analyze it directly, the adversary can study $\hat{T}$ to discover T's weakness.

The evasion attack, as shown in Figure 4, can be done as follows.

1. Run $\hat{T}$ on each test data.
2. If the score τ for a sample is larger than τ_0 and smaller than τ_1, where τ_0 is a parameter less than τ and τ_1 is a parameter more than τ, then this sample is selected for evasion attack.
3. Generate instances (i.e., flows) for selected samples and observe the performance of evasion attack, which is measured by e_{MD} and e_{FA} of T on selected samples.

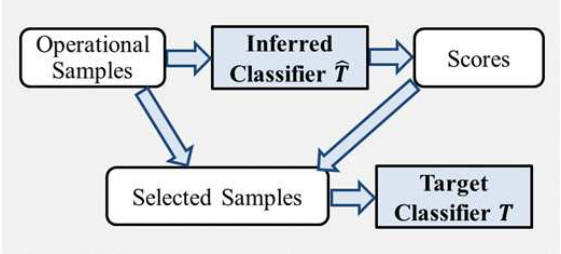

Figure 4: Evasion Attack

In the simulation result, we select $\tau_0 = 0.5\tau$ and $\tau_1 = 0.5 + \tau_0$ such that approximately half of the test data is selected. Also note that the observation on the effect of evasion attack needs the knowledge of ground truth, which may not be available to the adversary. To show performance in simulation, we assume the adversary has such knowledge. Table 3 summarizes the attacks results.

Table 3: Performance Comparison of Evasion Attacks

	e_{FA}	e_{MD}	$\max\{e_{MD}, e_{FA}\}$
Attack w/o synthetic data	8.00%	3.57%	8.00%
Attack with GAN data	33.33%	0.00%	33.33%
Attack with cGAN data	16.66%	9.56%	16.66%

For the first case in Table 3, the adversary does not use synthetic data in exploratory attack. Due to limited training data in exploratory attack, the evasion attack (based on exploratory attack results) is not very successful, i.e., T still has good performance. The second case is using 520 real data and 500 synthetic samples generated by GAN as training data to build $\hat{T}$ and then select samples for evasion attack. Among 520 samples in test data, 48

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

samples are selected causing T not working well on negative samples, i.e., $e_{FA} = 33.33\%$. However, T does not make any error on positive samples. This is caused by different distributions in synthetic data generated by GAN and in real data. The third case is using 520 real data and 500 synthetic samples generated by cGAN as training data to build $\hat{T}$ and then select samples for evasion attack. Among 520 samples in test data, 144 samples are selected and they make T not working well on both negative and positive samples. In other words, evasion attack with cGAN can substantially increase both misdetection and false alarm of the target classifier. Note the purpose here is to demonstrate the capability of evasion attack. In reality, the attacker might have different performance requirements, e.g., to maximize the misdetection probability e_{MD}.

VI. CAUSATIVE ATTACK

We now consider causative attack for the scenario that the target classifier is updated by user feedback to further improve accuracy. The attacker alters the retraining data such that target classifier T is updated and has a worse performance. The goal is to maximize the performance metric $max\{e_{MD}, e_{FA}\}$. The adversary is assumed to be able to access the training samples, manipulate (some of) the samples, and observe or infer the output (i.e., labels) of T. To maximize the influence, the attacker may change the label for each sample. However, it is easy to detect and defend this naïve attack. Thus, the attacker needs to carefully select samples and change labels on selected samples only.

We used the deep learning score to select samples for the causative attack, as shown in Figure 5.

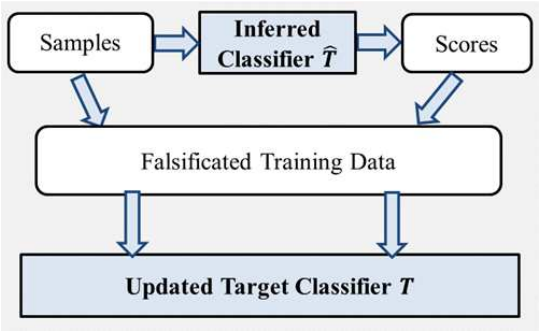

Figure 5: Causative Attack

The causative attack has the following steps:

1. Run $\hat{T}$ on each test data.
2. If the score τ for a sample is smaller than τ_0 or larger than τ_1, where τ_0 is a parameter less than τ and τ_1 is a parameter more than τ, then this sample is selected for causative attack.
3. Generate an instance for each selected sample and assign it an incorrect label. Use this data as a user feedback to update T.

In the simulation result, we select $\tau_0 = 0.5\tau$ and $\tau_1 = 0.5 + \tau_0$ such that approximately half of the test data is selected. The causative attack results are summarized in Table 4.

Table 4: Causative Attack Results

	e_{MD}	e_{FA}	$\max\{e_{MD}, e_{FA}\}$
GAN	69.83%	61.47%	69.83%
cGAN	60.34%	72.48%	72.48%

Table 4 shows that both GAN and cGAN can increase T's classification errors to more than 60%. Note that the entire process of the causative attack needs a large amount of data. Thus, an attacker must use synthetic data from GAN or cGAN to launch the causative attack.

VII. CONCLUSION

This paper presented the design of various attacks in our toolset using deep learning on a target classifier to classify flows are either normal or malicious. We showed that under black box assumption, the attacker can successfully launch an exploratory attack to build its classifier, which can make similar classification as the target classifier. We developed the cGAN approach to generate synthetic data to reduce the requirement on the amount of collected data for the exploratory attack. We also developed further attacks, i.e., evasion attack and causative attack, either to identify samples that cannot be accurately classified by the target classifier or to re-train the target classifier using tampered samples. The results in this paper show the critical need to investigate and explore the security issues associated with the deep learning systems, and accordingly enhance the defense mechanisms. We will perform our future work along four directions: (1) attacks under even more realistic settings (e.g., what if attacker cannot collect the same feature values as those observed by T; (2) better understand attack results and use them to manipulate malicious programs to falsify the output

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

of target classifier, e.g., generate flow instances for selected traffic; (3) the other types of attacks; and (4) defense mechanism to mitigate the attacks.

ACKNOWLEDGMENT

The authors thank the U.S. Army Combat Capabilities Development Center (CCDC) for their support in this effort. The authors also thank Metin Ahiskali, Michael De Lucia and Stephen Raio for their guidelines directing this research.

REFERENCES

[1] L. Huang, A. Joseph, B. Nelson, B. Rubinstein, J. Tygar, "Adversarial Machine Learning," in Proceedings of 4th ACM Workshop on Artificial Intelligence and Security, Oct. 2011, pp. 43-58.

[2] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," in International Conference on Learning Representations (ICLR) 2017.

[3] Y. Shi, Y. E. Sagduyu, and A. Grushin, "How to Steal a Machine Learning Classifier with Deep Learning," in IEEE Symposium on Technologies for Homeland Security (HST), 2017.

[4] Y. Shi and Y.E. Sagduyu, "Evasion and Causative Attacks with Adversarial Deep Learning," in Proceedings of IEEE Military Communications Conference (MILCOM), Baltimore, MD, October 23–25, 2017.

[5] Stratosphere IPS project, https://stratosphereips.org/.

[6] The CTU-13 dataset in Stratosphere IPS project, https://mcfp.felk.cvut.cz/publicDatasets/CTU-13-Dataset.

[7] CIC IDS 2018 dataset, http://www.unb.ca/cic/datasets/ids-2018.html.

[8] ISCX IDS 2012 dataset, https://www.unb.ca/cic/datasets/ids.html.

[9] DEFCON dataset, https://media.defcon.org/.

[10] I. Goodfellow, et al, "Generative Adversarial Nets," Advances in Neural Information Processing Systems, 2014.

[11] Forum of Incident Response and Security Teams (FIRST), https://www.first.org/resources/papers/2015

[12] Gauthier, Jon. "Conditional generative adversarial nets for convolutional face generation." Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, no. 5 (2014): 2.

[13] Shu, Rui, Hung Bui, and Stefano Ermon. "AC-GAN Learns a Biased Distribution." NIPS Workshop on Bayesian Deep Learning. 2017.

[14] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, Oct. 12–16, 2015.

[15] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv preprint arXiv:1607.02533, 2016.

[16] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Prague, Czech Republic, Sept. 23–27, 2013.

[17] G. Ateniese, L. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," International Journal of Security and Networks, vol. 10, issue 3, pp. 137–150, Sept. 2015.

[18] F. Tramer, F. Zhang, A. Juels, M. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," USENIX Security, Austin, TX, Aug. 10–12, 2016.

[19] Y. Shi, T. Erpek, Y.E. Sagduyu, and J.H. Li, "Spectrum data poisoning with adversarial deep learning," in Proc. IEEE Military Communications Conference (MILCOM), Los Angeles, CA, pp. 407–412, Oct. 29–31, 2018.

[20] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," International Conference on International Conference on Machine Learning, Edinburgh, Scotland, June 26 - July 1, 2012.

978-1-7281-5092-5/19/$31.00 ©2019 IEEE