Model Evasion Attack on Intrusion Detection Systems using Adversarial Machine Learning

Md. Ahsan Ayub*, William A. Johnson†, Douglas A. Talbert‡, and Ambareen Siraj§
Department of Computer Science
Tennessee Technological University
Cookeville, USA
{mayub42*, wajohnson43†}@students.tntech.edu
{dtalbert‡, asiraj§}@tntech.edu

***Abstract*—Intrusion Detection Systems (IDS) have a long history as an effective network defensive mechanism. The systems alert defenders of suspicious and / or malicious behavior detected on the network. With technological advances in AI over the past decade, machine learning (ML) has been assisting IDS to improve accuracy, perform better analysis, and discover variations of existing or new attacks. However, applications of ML algorithms have some reported weaknesses and in this research, we demonstrate how one of such weaknesses can be exploited against the workings of the IDS. The work presented in this paper is twofold: (1) we develop a ML approach for intrusion detection using Multilayer Perceptron (MLP) network and demonstrate the effectiveness of our model with two different network-based IDS datasets; and (2) we perform a model evasion attack against the built MLP network for IDS using an adversarial machine learning technique known as the Jacobian-based Saliency Map Attack (JSMA) method. Our experimental results show that the model evasion attack is capable of significantly reducing the accuracy of the IDS, *i.e.,* detecting malicious traffic as benign. Our findings support that neural network-based IDS is susceptible to model evasion attack, and attackers can essentially use this technique to evade intrusion detection systems effectively.**

***Index Terms*—Adversarial Machine Learning, Evasion Attack, Intrusion Detection System, Neural Network**

I. Introduction

The use of Machine Learning (ML) in Intrusion Detection System (IDS) is widespread and has demonstrated remarkable performance as a robust and effective defense mechanisms [37], [41]. An IDS provides detection capabilities over malicious traffic by generating alerts with network logs such that further intelligence can be derived as and when needed. Based on its placement in the network infrastructure, a network-based IDS monitors the communication that travels into and out of the network, while a host-based IDS scans a particular host (*e.g.*, server), to notify the network administrator for possible security threats. The two types of IDSs are (1) Signature-based IDS that analyze network traffic is for known malicious signatures and (2) Anomaly-based IDS that compares the network traffic against a user's known patterns and raises an alert if it deviates from the pattern. Researchers have leveraged various ML based classifiers, such as, Artificial Neural Networks, Decision Trees, Support Vector Machine (SVM), Fuzzy Logic, and Bayesian Networks to detect malicious traffic as well as discover unseen attacks that deviate from normal profile [10], [39]. In our study, we focus on network-based IDS, also referred as NIDS, and Artificial Neural Network (ANN) as our machine learning algorithm.

Despite success of machine learning for intrusion detection, the advent of Adversarial Machine Learning has recently emerged as a significant threat to the effectiveness of such applications. An adversary can exploit vulnerabilities in the machine learning algorithm itself or the trained ML model to compromise network defense [16]. There are various ways this can be achieved, such as, Membership Inference Attack [36], Model Inversion Attack [11], Model Poisoning Attack [25], Model Extraction Attack [42], Model Evasion Attack [3], Trojaning Attack [22], etc. The range of these attacks typically depends on the level of access an adversary has to the trained model. For example, an adversary may have perfect knowledge about the type of the model used as well as its workings or s/he may have no knowledge about the model at all. Our focus in this research is on demonstrating a Model Evasion Attack for IDS whereby an adversary can evade the ML model for network-based IDS by crafting adversarial samples. If s/he is successful, the attacker may be able to gain access to the network with malicious traffic and cause significant harm.

The following are the main contributions of the paper:

- We construct a Multilayer Preceptron (MLP) model, a popular Neural Network topology, to perform binary classification over benign and attack traffic in a network-based anomaly IDS. In our experiments, we achieve more than 99% accuracy for all experimental datasets used.
- We demonstrate Model Evasion Attack against the built MLP model in a white-box setting, where the accuracy of the attacked model drops significantly and discuss possible countermeasures to prevent this type of attacks.

The rest of the paper is organized as follows: Section 2 provides an overview of the Model Evasion Attack. Section 3 first explains the datasets we use in our experiments, followed by the evaluation of our twofold research by discussing the construction of a Multilayer Perceptron (MLP) network as well as its classification results and then describing the design of our attack and its effectiveness. We conclude this section with a description of some possible countermeasures against this attack. Section 4 provides an overall discussion of our

978-1-7281-4085-8/20/$31.00 ©2020 IEEE

experiments, followed by relevant work in section 5. Section 6 summarizes the paper, its contributions, and future work to further improve upon this research.

II. Model Evasion Attack

The goal of Model Evasion Attack is to cause the machine learning model to misclassify observations during the testing phase (as shown in Fig. 1). Applied to a network-based IDS, an adversary attempts to evade detection by altering the malicious instances in such a way that the IDS misclassifies this behavior as *benign*. To elaborate, there are four different ways this can take place [26]: *Confidence Reduction*, where reducing the confidence score output leads to misclassification; *Misclassification*, where an adversary tries to alter the correct output classification to a class than the original class; *Target Misclassification* (our approach), where the adversary produces a sample that fools the model to classify the behavior as a target class; and *Source / Target Misclassification*, where the attacker makes the output class classification of a specific adversarial sample to be a specific target class.

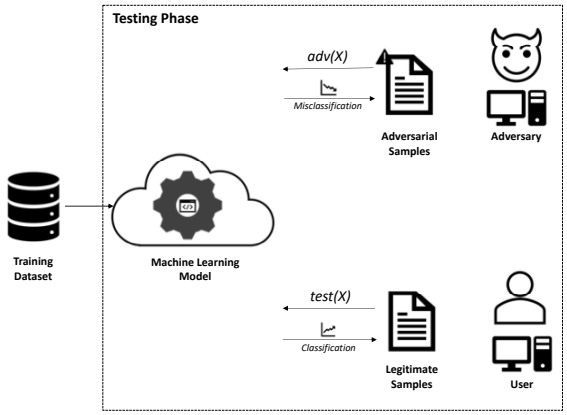


Fig. 1. Diagram of the model evasion attack against a trained machine learning model during the testing phase.

To clarify the capabilities of an adversary, we outline three different knowledge scenarios [29] -

White Box. The adversary has perfect knowledge of the target classification model including the type of the classifier used and its structure. S/he also knows all parameters of the model that are required to perform prediction as well as all or part of the training dataset and its features.

Black Box. The adversary has zero knowledge of the target model. It may be known that the model performs classification, but the adversary does not have access to the training data, model structure or type, or any parameters of the model. S/he is able to offset this lack of information by querying the model for potential information leakage.

Grey Box. The adversary has an incomplete knowledge of the target model and knows the features considered by the model and its type. S/he does not have any part of the training set or the weights within the model. As in the case of the Black Box model, the adversary can query the target model such that any information the model leaks, can potentially be exploited.

In all cases, the adversary is only able to alter the data during the testing phase of the classification. Our attack depends on the knowledge about certain parameters used in the trained model but does not depend on the training dataset. Because of this, we consider our attack scenario to be white box. We also assume that the attacker is able to modify the test instance in such a way as to modify any of the features seen by the IDS. Additionally, the general intuition behind performing a successful model evasion attack is to define a loss function that the adversary aims to maximize or minimize for each sample to results in misclassification [3], [19]. It is important to note that, our experimentation is based on a white-box setting and do not evaluate the attack's effectiveness based on other attack scenarios (*e.g.*, black-box setting).

III. Evaluation

We first describe the datasets that we use for evaluation, followed by the description of the target model and our experimental setup. We then present the results of model evasion attacks against different datasets.

A. Dataset

CICIDS 2017. Released by the Canadian Institute for Cybersecurity in 2017, this dataset closely resembles real-world data [35]. The IDS logs were recorded over five days with a total 51.1 GB of packet capture (PCAP) files[1] built upon the abstract behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols. With 12 different victim machines and 2 attacker machines, this labelled dataset features common attacks, such as, Web based Brute Force, XSS and SQL Injection, DoS, DDoS, Infiltration, Heart-bleed, Bot, and Scan. The dataset is available online for public use[2].

For our experiments, we sampled $950,000$ records in total. Each record consists of 80 continuous features with a binary labelled class of *benign* or *attack*.

TRAbID 2017. Viegas et al. [43] produced a network based intrusion database in a controlled and reproducible environment. To depict a real-world use case, the dataset includes client-server communication. Legitimate traffic was generated by the client requesting services available in the server, such as, HTTP, SMTP, SSH, SNMP, and DNS, while the attacker from a client machine launched attacks to the same server. The type of the attacks primary included different categories of DoS (*e.g.*, SYN flood, ICMP flood, etc.) and Scan (*e.g.*, SYN scan, ACK scan, etc.).

For our experimentation, we collected $18,000$ records in total. Unlike CICIDS 2017, each record in this case consists of 43 continuous features. The dataset is also labelled with a binary class of *benign* or *attack* and is available online

[1]https://fileinfo.com/extension/pcap

[2]https://www.unb.ca/cic/datasets/ids-2017.html

for public use[3]. It is important to note that both datasets are balanced and suitable for binary classification tasks.

B. Building the ML Model for IDS

We now describe the construction of the target machine learning model for network-based intrusion detection system (IDS) and its performance in detecting attack traffic.

Multilayer Perceptron (MLP) Network. We use a Multilayer Perceptrons (MLP) network, which is a widely used Neural Network topology. With w as the real vector of weights, $b \in \mathbb{R}$ as the bias, and h as the transfer function, the decision function of the MLP can be formally defined:

$$f(x) = w \cdot h(v_i \cdot x + d_i) + b$$

where, $(v_i, d_i) \in \mathbb{R}^n \times \mathbb{R}$ is a representation of the weight of the i-th hidden unit [8]. An MLP network is usually constructed with three or more layers, that is, one input layer, one or more hidden layer, and one output layer. We select one hidden layer and build a fully connected network (*i.e.*, each node in the input layer is connected with a certain weight to every node in the hidden layer) as shown in Fig. 2.

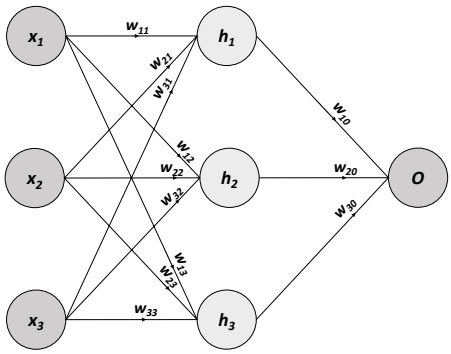


Fig. 2. A simple Multilayer Perceptron (MLP) network with one input layer $X = (x_1, x_2, x_3)$, one hidden layer $H = (h_1, h_2, h_3)$, and one output O.

We use a Rectified Linear Unit (ReLU) activation function [32], where the function and its derivative are monotonic with a range of 0 to ∞, in the input layer as well as hidden layer of the MLP network. Since we derive the probability value of the binary class in the output layer, we use Sigmoid activation function, where the S-shaped function is differentiable with a range of 0 and 1 [40]. To compile the model, we use an Adam optimization algorithm, a first-order gradient-based optimization of stochastic objective functions [18], and Binary Cross Entropy (BCE) loss function [21]. To control the generalization ability of a perception, we incorporate the use of early stopping during training. We then monitor validation loss to trigger this action. We also select 10% of the training records as the validation set to perform this task.

We base our analysis on both the CICIDS 2017 and TRAbID 2017 datasets and tune our model similar to what we discussed before. We first scale the dataset so that all the feature values are in the range of 0 to 1, which is also referred as MinMax Scalar. Then, we split the data in 80% training and 20% testing instances. Thus, we have approximately $760,000$ and $14,400$ training records from CICIDS and TRAbID dataset respectively. Additionally, we perform stratified splits of training and testing instances to preserve the same percentage for each target class as in the complete set provided in the dataset.

Experimental Results. In this section, we describe our experimental results in terms of *Accuracy*, which denotes the extent of correct predictions by the model; *Precision*, which is a measurement of the ratio of the true positive records to all positively labelled instances; *Recall*, which is the ratio of the true positive instances to all instances that should have been labelled positive; and F_1 score, which is the harmonic mean of precision and recall [30].

$$\text{Precision Score} = \frac{\sum \text{True Positive}}{\sum \text{True Positive} + \sum \text{False Positive}}$$

$$\text{Recall Score} = \frac{\sum \text{True Positive}}{\sum \text{True Positive} + \sum \text{False Negative}}$$

$$F_1 \text{ Score} = 2 \cdot \frac{\text{Precision Score} \times \text{Recall Score}}{\text{Precision Score} + \text{Recall Score}}$$

For both datasets, MLP model was succesfull in accurately detecting benign traffic as well as attack traffic. Fig. 4 shows that the model performed at 99.5% accuracy for CICIDS 2017 dataset while at 99.8% accuracy for TRAbID 2017 dataset.

To further illustrate our results with the described parameters (*i.e.*, precision, recall, and F_1) scores, Table I shows that the obtained results are very close in all settings. We demonstrate the performance of benign and attack traffic prediction individually with weighted average.

TABLE I
PERFORMANCE OF THE MLP NETWORK ON THE LEGITIMATE INSTANCES.

Particular	Precision		Recall		F_1		Support	
	CICIDS	*TRAbID*	*CICIDS*	*TRAbID*	*CICIDS*	*TRAbID*	*CICIDS*	*TRAbID*
Benign	0.9957	0.9978	0.9951	0.9973	0.9954	0.9975	110,246	1,832
Attack	0.9932	0.9973	0.9951	0.9978	0.9954	0.9975	79,553	1,8317
Weighted Avg.	**0.9946**	**0.9975**	**0.9946**	**0.9975**	**0.9945**	**0.9975**	**189,799**	**3,663**

We train both model for 100 epochs with a batch size of 64. From Fig. 3, we notice that the MLP network tends to overfit after 13 epochs for CICIDS 2017 and 16 epochs for TRAbID 2017. However, we preserve the generalization of the model by triggering an early stopping task based on the validation loss. Additionally, we define a delay of 2 epochs to verify that there were no signs of improvement after the initial indication.

C. Adversarial Machine Learning to Evade IDS

In this section, we report on how we have used the built MLP model as our target model to launch the model evasion attack using Adversarial Machine Learning.

Attack Design. The purpose of the IDS evasion attack is to generate data samples in such a way that it confuses the trained MLP model to classify malicious data as benign. In other

[3]https://secplab.ppgia.pucpr.br/?q=trabid

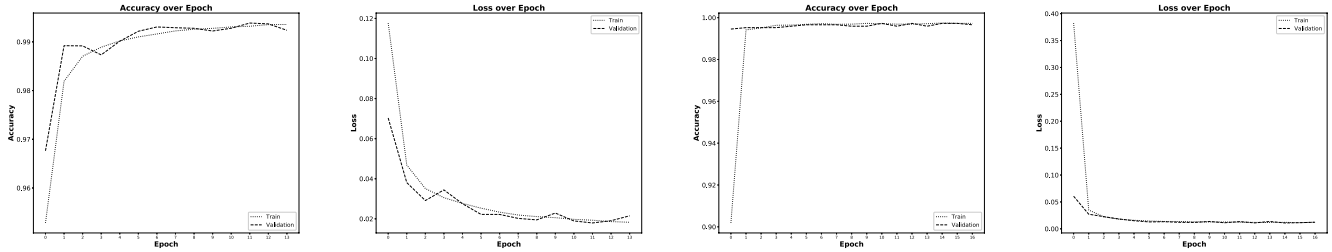
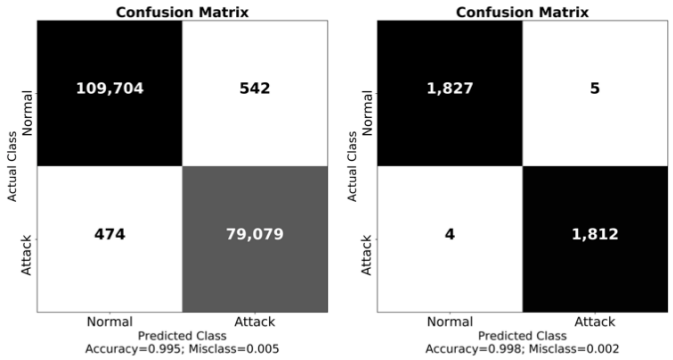


Fig. 3. Accuracy over Epoch and Loss over Epoch curves for the MLP generated on CICIDS 2017 (on left) and TRAbID 2017 dataset (on right).



(a) CICIDS 2017 (b) TRAbID 2017

Fig. 4. Confusion matrix graphs for MLP generated on two different datasets.

words, we aim to inject adversarial instances to the model during testing and obtain benign outputs. More formally, we desire to craft adversarial sample X^* by adding a perturbation to the legitimate sample X (*i.e.*, $X^* \leftarrow X + \delta X$, where δ is denoted as perturbation), such that $F(X^*) = Y^* \neq Y$.

In this study, we base our attack analysis on the assumption that the only knowledge the adversary has is about the parameters the model uses to predict the binary class (benign or attack). This is close to a real-world setting because it is not trivial for the adversary to easily infer knowledge about the model architecture trained for network-based IDS. On a related note, Tramèr et al. [2016] did show that it is possible to extract internal information of the MLP network architecture [42].

Our attack design is primarily focused on creating adversarial test samples based on Jacobian-based Saliency Map Attack (JSMA) [26]. With only a small perturbation in the legitimate data samples, the JSMA generates adversarial samples based on the Saliency Map method, as described in [38]. We leverage the saliency map to search through the legitimate test instances, closely observe the sensitivity information to choose a perturbation δX among the input dimensions that is likely to fool the built MLP model, and make iterative changes to produce an adversarial sample set. Following these steps, we exploit legitimate test instances collected from both datasets.

Attack Effectiveness. We evaluate the effectiveness of our attack by testing the trained MLP model with the adversarial test samples. As shown in Table II, we notice 21.52% and 29.87% performance drop in terms of accuracy for CICIDS 2017 and TRAbID 2017 dataset respectively. This demonstrates that the trained MLP model fails to correctly predict adversarial samples as attack traffic. It also signifies that an adversary can craft a malicious network traffic such that the network-based IDS classifies it as *benign*. Thus, the defense mechanism is successfully evaded by exploiting the legitimated traffic communication between the client and the server.

TABLE II
PERFORMANCE OF MODEL EVASION ATTACK ON THE MLP NETWORK.

Dataset	**# Instances**	**Accuracy** *Legitimate Instances*	**Accuracy** *Adversarial Instances*	**Performance Drop**
CICIDS	$190,291$	99.5%	78.09%	21.52%
TRAbID	$3,694$	99.8%	69.99%	29.87%

It is worthwhile to mention that there are other methodologies to simulate similar types of attacks (*e.g.*, Fast Gradient Sign Method (FGSM) [12]); however, the JSMA method possesses the ability to generate adversarial samples with a lesser degree of distortion. Additionally, the FSGM method has been found to be ill-suited for the IDS setting [33].

Implementation. We implement our MLP model using Keras [7], utilizing the Python machine learning tool Scikit-Learn [28] to execute data processing tasks, and used Matplotlib library [17] to generate all the graphs in this paper. To implement the attack, we used CleverHans, a Python library to benchmark machine learning systems' vulnerability to adversarial examples [24]. Finally, our implementation of the attack was tested with TensorFlow version 1.13.1 [1]. Our implementation has been made open source for the community with MIT License and is available online[4].

Countermeasures. Following is a discussion of a few potential countermeasures to prevent the model evasion attack that we used on our MLP network. Papernot et al. [2016] leveraged the distillation method, introduced by Hinton et al. [15], to reduce the size of the deep neural network and thus the computing resources, as a defense strategy to reduce the network's vulnerability to adversarial sample generation [27]. The paper pointed out that reducing the amplitude of the adversarial gradient would enable the model to generalize better since crafting adversarial samples becomes easier when adversarial

[4]https://github.com/TnTech-CEROC/adversarial_ml_ids

gradients are high. Thus, this increases the resilience of the model to adversarial samples. Other mitigation techniques for model evasion attack include methods to tighten the decision boundary of a classification algorithm so that benign features cannot easily be applied to malicious samples [3], training the classifier over malicious samples generated with adversarial knowledge [34], and removing features from the model that are not immediately necessary [2], [46]. One of our future work will be to validate and compare the effectiveness of such countermeasures applied to our research.

IV. Discussion

We evaluate our MLP network trained over two different datasets and are confident that its accuracy with other network-based IDS dataset will be similarly effective. We then analyze model evasion attack on the presented Multilayer Perceptron (MLP) model for the target class misclassification. In this way, we fool the model into misclassifing attack records as benign and hence evade the network defense. Reducing the output confidence of the predicted class by the classifier would be another avenue to perform this kind of evasion attack, which we leave for future work.

We base our experimentation on white-box setting and do not evaluate the attack's effectiveness based on other attack scenarios (*e.g.*, black-box setting). We validate the effectiveness of our attack in terms of the drop in accuracy when tested with the adversarial samples crafted with small perturbations on the input dimensions and the Jacobian-based Saliency Map Attack (JSMA) method. Another avenue of our future work includes gathering knowledge of sensitive features that cause flipping the prediction of the class.

V. Relevant Work

Adversarial machine learning has become a topic of much interest in the cybersecurity space. This is largely because classification algorithms have been successful in solving the problems of malware detection. There are different kinds of adversarial machine learning techniques that allow attackers to subvert these classification algorithms in malicious ways. One such methodology is known as model evasion attack that allows an adversary to alter an adversarial sample such that it is misclassified as benign. Pitropakis et al. [2019] provided a detailed taxonomy on the model evasion attack to well understand different types of the applications, the architecture of the models, and the used datasets [29].

Model evasion attack is often done via gradient descent over the discrimination function of the classifier [3], [4], [6]. By applying gradient descent over the discrimination function of the classifier, these methodologies are able to identify traits of benign samples, such that these traits may be applied to malicious samples to force misclassification. Much like [33], we also leverage the Jacobian-based Saliency Map Attack (JSMA) method to evade the model's detection using different datasets. Gradient descent methodologies are not without their weaknesses. They specifically target classifiers with differentiable discrimination functions. Such classifiers typically include Support Vector Machine (SVM) with differentiable kernels and neural networks. Other methodologies have been devised to perform attacks on a broader spectrum of classification algorithm, including [45] which uses genetic algorithms, and in [26] which devises a Forward Derivative based on Jacobian Matrices.

We demonstrate the model evasion attack on the network-based IDS; however, there are other applications where this type of attack was employed such as, spam filtering [13], [20], [23], visual recognition [5], [14], [31], and malware detection [3], [9], [44]. This research adds intrusion detection to the list.

VI. Conclusion

In our study, we first build a supervised machine learning model to detect and classify benign and attack traffic using two different network-based intrusion detection system (IDS) datasets: CICIDS 2017 [35] and TRAbID 2017 [43]. We construct a Multilayer Perceptron (MLP) network to perform binary classification task and achieved outstanding detection results, i.e., 99.5% and 99.8% accuracy for CICIDS and TRAbID, respectively. We then apply model evasion attack from the adversarial machine learning suite to demonstrate that it is possible to evade intrusion detection systems effectively. We consider the trained MLP model as our target model. To implement our attack, we select the Jacobian-based Saliency Map Attack (JSMA) method in a white-box setting, where an adversary has perfect knowledge over the parameters required for the model to perform prediction. In other words, the adversary crafts adversarial samples with small perturbation to the legitimate testing samples and attempts to fool the model during the testing phase. We demonstrate success in our attack and validated its effectiveness in terms of 22.52% and 29.87% accuracy drop in performance for CICIDS and TRAbID, respectively. This signifies that evading network defense is possible without much effort unless proper countermeasures are undertaken, as discussed.

Acknowledgement

The work reported in this paper has been entirely supported by Cybersecurity Education, Research & Outreach Center (CEROC) at Tennessee Technological University.

References

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016. URL: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

[2] Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing robustness of machine learning systems via data transformations. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE, 2018.

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[4] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[5] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[6] Lingwei Chen, Yanfang Ye, and Thirimachos Bourlai. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 99–106. IEEE, 2017.

[7] François Chollet. Keras, 2016. URL: https://github.com/fchollet/keras.

[8] Ronan Collobert and Samy Bengio. Links between perceptrons, mlps and svms. In *Proceedings of the twenty-first international conference on Machine learning*, page 23. ACM, 2004.

[9] Luca Demetrio, Battista Biggio, Giovanni Lagorio, Fabio Roli, and Alessandro Armando. Explaining vulnerabilities of deep learning to adversarial malware binaries. *arXiv preprint arXiv:1901.03583*, 2019.

[10] Bo Dong and Xue Wang. Comparison deep learning method to traditional methods using for network intrusion detection. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 581–585. IEEE, 2016.

[11] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015.

[12] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL: http://arxiv.org/abs/1412.6572.

[13] Michael Großhans, Christoph Sawade, Michael Brückner, and Tobias Scheffer. Bayesian games for adversarial regression problems. In *International Conference on Machine Learning*, pages 55–63, 2013.

[14] Jamie Hayes and George Danezis. Machine learning as an adversarial service: Learning black-box adversarial examples. *arXiv preprint arXiv:1708.05207*, 2, 2017.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J Doug Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58. ACM, 2011.

[17] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. http://dx.doi.org/10.1109/MCSE.2007.55 doi:10.1109/MCSE.2007.55.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Pavel Laskov and Marius Kloft. A framework for quantitative security analysis of machine learning. In *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*, pages 1–4. ACM, 2009.

[20] Bo Li and Yevgeniy Vorobeychik. Feature cross-substitution in adversarial classification. In *Advances in neural information processing systems*, pages 2087–2095, 2014.

[21] Jenny Liu. Global optimization techniques using cross-entropy and evolution algorithms. *Master's Thesis, Department of Mathematics, University of Queensland*, 2004.

[22] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

[23] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647. ACM, 2005.

[24] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

[25] 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.

[26] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[27] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. A taxonomy and survey of attacks against machine learning. *Computer Science Review*, 34:100199, 2019.

[30] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[32] Prajit Ramachandran, Barret Zoph, and Quoc Le. Searching for activation functions. 2018. URL: https://arxiv.org/pdf/1710.05941.pdf.

[33] Maria Rigaki. Adversarial deep learning against intrusion detection classifiers, 2017.

[34] Paolo Russu, Ambra Demontis, Battista Biggio, Giorgio Fumera, and Fabio Roli. Secure kernel machines against evasion attacks. In *Proceedings of the 2016 ACM workshop on artificial intelligence and security*, pages 59–69. ACM, 2016.

[35] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.

[36] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[37] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[38] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[39] Jayveer Singh and Manisha J Nene. A survey on machine learning techniques for intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(11):4349–4355, 2013.

[40] Maxwell Stinchcombe and Halbert White. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. In *IJCNN International Joint Conference on Neural Networks*, 1989.

[41] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. Deep learning approach for network intrusion detection in software defined networking. In *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 258–263. IEEE, 2016.

[42] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016.

[43] Eduardo K Viegas, Altair O Santin, and Luiz S Oliveira. Toward a reliable anomaly-based intrusion detection in real-world environments. *Computer Networks*, 127:200–216, 2017.

[44] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pages 1689–1698, 2015.

[45] W Xu, Y Qi, and D Evans. Automatically evading classifiers: A case study on pdf malware classifiers. ndss, 2016.

[46] Fei Zhang, Patrick PK Chan, Battista Biggio, Daniel S Yeung, and Fabio Roli. Adversarial feature selection against evasion attacks. *IEEE transactions on cybernetics*, 46(3):766–777, 2015.