Overview of Adversarial Machine Learning with emphasis on Malware Classification

Dmitrijs Trizna
Department of Computer Science
University of Helsinki
Helsinki, Finland
dmitrijs.trizna@helsinki.fi

ABSTRACT

During the last decade, the Machine Learning community realized that model behavior can be affected by the presence of a potential adversary. Producing slight perturbations on input data, unperceivable to a human analyst, threat actor capable to create an adversarial example. With such input, ML models result in malfunctional operations like misclassification. This proved to be a severe challenge for specific domains, like computer vision or malware classification. We provide insights into why adversarial examples succeed, with a review of known attack types and threat models. Furthermore, we examine the problem in a malware classification setting. An analysis of modern malware evasion techniques based on reinforcement learning, genetic algorithms, and other techniques is provided. Additionally, we cover known defenses, their strengths, and weaknesses, alongside promising future research directions.

CCS CONCEPTS

• **Computer systems organization → Machine Learning**; • **Security and privacy** → *Malware and its mitigation.*

KEYWORDS

adversarial examples, machine learning, security, malware, neural networks

1 INTRODUCTION

Modern Machine Learning (ML) algorithms are a result of more than 50 years of research. Discriminative ML models discover probability distribution $p(y|x)$, which represents a mapping of some specific input data vector x, to an output vector y. Historically ML algorithms were developed with an assumption that environment and input data are benign during training and evaluation. However, recent advancements in adversarial input generation revealed a severe challenge for ML models in various sensitive domains.

Adversarial inputs are data samples generated by potential "adversary" to affect the ML model's behavior. Successful adversarial example (AE) contains carefully chosen information perturbations, that result in a different ML algorithm's output y. Consider scenario with two-dimensional input data x, with ML model doing a binary classification task, providing $y \in \{0; 1\}$. AE for this model can alter x_1 dimension, resulting in crossing the decision boundary. Eventually, this affected the model's prediction in a misclassification. Real-world data is more multivariate, and adversaries need to perturb only a part of input dimensions, like altering a subset of pixel RGB values in the input image.

It is important to emphasize that AE still needs to possess characteristics of the original data class. It could be a modified road sign that fools the classifier, but appears the same for human's eye, or a malware sample that evades anti-virus software, yet executes malicious logic.

Adversarial attacks are a rapidly growing field. Since 2014 there have been about 5000 papers on adversarial attacks and defense[1]. Nonetheless, we notice a severe dis-balance in research focus. Let's consider several sensitive ML deployment domains, where adversaries are especially dangerous:

- Face recognition
- Autonomous vehicles
- Financial and trading algorithms
- Malware classification

While tasks in the first two fields represent an image analysis problem, other fields are highly distinct from image classification. For example, financial and trading data mainly consists of time-series data. Malware classification has even more distinct data, actual executable files. Moreover, the presence of adversaries in malware classification is a real-world fact. The ability to build a robust model, that can defend against adversarial inputs, is crucial for this field. However, only about 2% (approx. 60 publications at the moment of writing) focus on the malware domain, whereas the prevailing part of research discusses AE only in the image classification setting.

The aforementioned situation might be caused by the fact, that cutting-edge adversarial robustness research for malware classification requires complex teams, with specialists

[1]https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

from different domains. Besides machine learning and mathematical background, such research requires a high degree of cybersecurity domain and reverse engineering expertise. AE generation depends on an understanding of executable file structure and operating system functionality since modifications cannot break malware behavior.

Therefore, in this article, we first describe adversarial examples as a general discipline but then focus on methods for malware classification evasion and defenses. In Section 2 we discuss the taxonomy of known adversarial techniques, developed mostly for the image classification domain. In particular, subsection 2.1 focuses on white-box methods, whereas subsection 2.2 describe known on black-box approaches. Further, we look at adversarial malware generation in Section 3. We show some methods prevalent in research, and their limitations in subsection 3.1. More realistic approaches are discussed in subsection 3.2. Finally, Section 4 covers potential defenses and characteristics of robust models. Ideas for future research and methodology advancements are proposed in Section 5.

2 TAXONOMY OF ADVERSARIAL ATTACKS

2.1 White-Box attacks

When an adversary has full access to model parameters and architecture, we call such threat model as *white-box*. The generation of adversarial examples in such a setting is similar to the ML algorithm's optimization problem. However, opposed to cost minimization, an adversary needs to identify the most vulnerable cost regions with respect to input parameters. Finding specific parameter perturbations that shift input to high-cost regions eventually leads to source-target-misclassification.

Therefore, parameter optimization algorithms can be used for adversarial example generation as well. For instance, Kolosnjaj et al. [7] weaponize version of gradient descent.

The general idea behind adversarial example x^* is to solve the problem of the smallest possible perturbation that causes misclassification:

$$x^* = x + \mathtt{argmin}\{||z|| : f(x+z) = t\}, \tag{1}$$

where x represents an input that originally is correctly classified, $||\cdot||$ is a norm that defines similarity constraints between x^* and x. Norm and, consequently, z value defines how the algorithm should change the original sample, for example on image l_0 norm might be used to produce noticeable changes, but for several pixels only, whereas l_∞ norm to modify every pixel by a tiny amount [3]. t defines a target class, it can be any different than the original class $f(x)$ in case of an *untargeted* attack, or have a specific class value for *targeted misclassification.*

Goodfellow et al. [3] describe three canonical examples to achieve this goal. Limited-memory Broyden – Fletcher – Goldfarb – Shanno (L-BFGS) algorithm is a quasi-Newton method originally used for parameter optimization. The other two algorithms are the Fast Gradient Sign Method (FGSM) and Jacobian Saliency Map Approach (JSMA). Characteristics of all three algorithms concerning runtime and stealth are represented in Figure 2.

JSMA and L-BFGS are iterative algorithms and produce stealthier perturbations that are harder to detect. Alas, it is achieved by a price of higher computational cost. FGSM on the contrary is much faster and may work better than the previous two if the gradient is relatively small. Needless to say that any of mentioned methods may fail to fool the classifier. The exception is an L-BFGS, which almost always succeeds given enough computation time since it is essentially a brute-force approach [3].

FGSM. Still, many researchers favor FGSM for AE generation, as it produces functional results with low computational demand. We observe this preference not only for attacks on image classification, but white-box malware evasion techniques also use FGSM if the target model's gradient is known. Hence we review this algorithm more deliberately.

General goal of algorithm is to produce adversarial example x^* that maximizes model's loss $J_f(x)$, given that x^* does not violate constrains of allowed modification ratio ϵ:

$$x^* = \max_{x^*} J_f(x), \ \exists \ ||x^* - x||_\infty \leq \epsilon \tag{2}$$

This problem can be treated as optimization process, but algorithm is typically intractable. However, if true J_f is substituted by first-order Taylor series approximation of J_f, using the gradient at x, closed form solution is as follows [4]:

$$x^* = x + \epsilon \cdot \mathtt{sign}(\nabla_x J_f(x, y)) \tag{3}$$

Therefore, FGSM based AE generation depends on the gradient of the loss function and hyperparameter ϵ. The higher is ϵ value, the more AE differs from the original sample, but the attack generally becomes more successful. Although it is beneficial to access the target model's gradient, since AE generation becomes more deterministic, such privilege is not mandatory for a successful attack as we see in the next subsection.

2.2 Black-Box attacks

While white-box attacks are important for understanding the nature of adversarial examples, the situation when an attacker has access to the inner functionality of the target model is relatively rare. The more realistic threat model is a *black-box* scenario when the adversary does not possess knowledge about inner architecture nor training of target

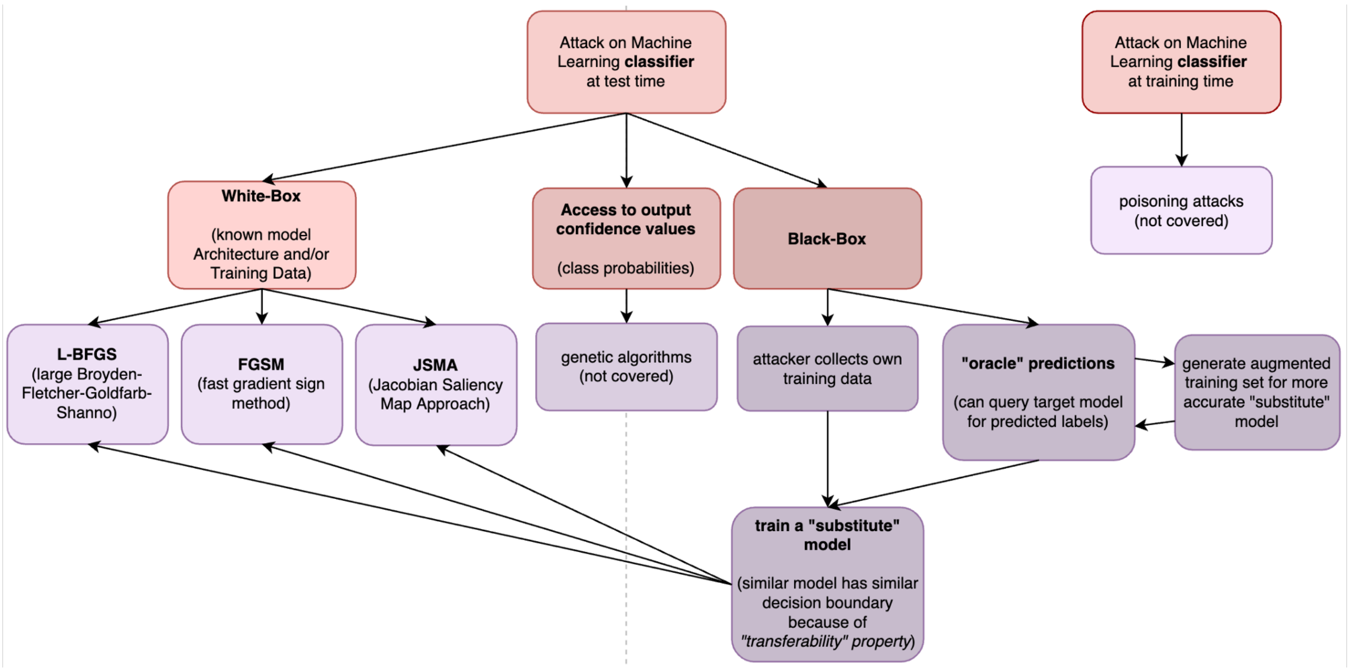
Attack on Machine Learning **classifier** at test time
Attack on Machine Learning **classifier** at training time
poisoning attacks (not covered)
White-Box
(known model Architecture and/or Training Data)
Access to output confidence values
(class probabilities)
Black-Box
L-BFGS
(large Broyden-Fletcher-Goldfarb-Shanno)
FGSM
(fast gradient sign method)
JSMA
(Jacobian Saliency Map Approach)
genetic algorithms (not covered)
attacker collects own training data
"oracle" predictions
(can query target model for predicted labels)
generate augmented training set for more accurate "substitute" model
train a "substitute" model
(similar model has similar decision boundary because of *"transferability" property*)

Figure 1: Temp (https://temp).

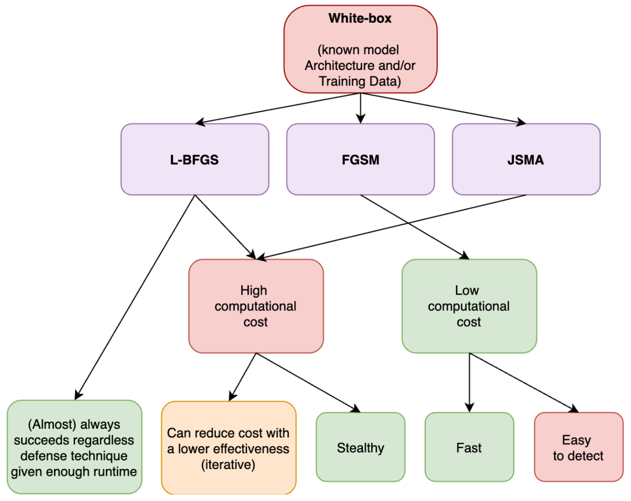
White-box
(known model Architecture and/or Training Data)
L-BFGS
FGSM
JSMA
High computational cost
Low computational cost
(Almost) always succeeds regardless defense technique given enough runtime
Can reduce cost with a lower effectiveness (iterative)
Stealthy
Fast
Easy to detect

Figure 2: Characteristics of L-BFGS, FGSM, and JSMA with respect to runtime and stealth.

model. Usually, an attacker may have only the model's predictions via a limited API interface. In research, such a model is called the *oracle O*. It is important to explicitly distinguish between two potential outputs from the oracle-predicted label $\tilde{O}(x)$ of input x and probability vector $O(x)$. Label, in this case, is most probable of N classes:

$$\tilde{O}(x) = \underset{0\cdots N-1}{\operatorname{argmax}}\, O(x)$$

Papernot et al. [12] provide in-depth analysis of minimalistic threat model with assumptions that (a) adversary have only label $\tilde{O}(x)$, and (b) adversary cannot collect a dataset comparable with target model's training data. They overcome both limitations by (a) training a substitute model and (b) generating a synthetic training set that is acquired using adversary inputs and labels from the oracle.

Adversarial examples are generated using the FGSM algorithm on the substitute model after it is trained on synthetic augmented data for a manually defined number of epochs p. They validate attack strategy against remote Deep Neural Network (DNN) model hosted by Oracle using either MNIST (hand-written digits) and GTSRB (road sign) datasets. Attack reach up to 84.24% success rate on target model if input variation parameter from Equation 3 is at least $\epsilon = 0.3$ [12].

Such technique appears to be efficient due to the "transferability" property first observed by Szegedy et al. [16]. Transferability is not DNN specific and holds across many types of machine learning models. For instance, the substitute logistic regression model can successfully mimic the decision boundary of support vector machine and decision tree models having similar perturbation norms [11].

When class probabilities $O(x)$ are not hidden by the defense, that provides additional epistemic capacity for an adversary and exposes additional attack vectors. For example,

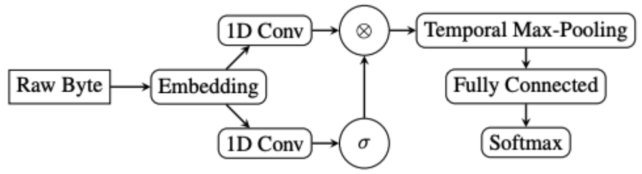

Figure 3: Architecture of MalConv model. [13]

as shown by Xu et al. [17] it is possible to generate AE using a genetic algorithm given class probabilities. They generate adversarial examples from malicious PDF files and use such a unique approach to overcome specific obstacles present in malware evasion we discuss further.

3 MALWARE CLASSIFICATION

Conventional anti-virus products relied on signature-based detections, which explicitly filtered out known malware samples. However, to build efficient defenses against previously unseen malware or even 0-day attacks[2], more sophisticated heuristics should be utilized.

One way to solve this problem is to introduce an ML algorithm, that learns common patterns across a huge dataset of known malware, obtaining a predictive power to classify malicious samples. Since the first paper in 2001 on malware detection using ML techniques [14], the field has grown notoriously, and most commercial anti-virus solutions include ML components. A noticeable challenge in research is the fact, that security vendors use ML-powered malware classifiers in a concealed, proprietary manner, without the ability to evaluate their inner functionality.

One of the successful open implementations was shown by Raff et al. [13] who presented a *MalConv* model - DNN that reads raw bytes of executable and proceeds with embeddings and 1-D convolution as seen in Figure 3. MalConv can successfully learn patterns of both malicious and benign file structures, reaching Area-Under-the-Curve (AUC) values as high as 98.5% on test set [13]. Model's open parameters and similarity with image classification architectures bootstrapped adversarial research for malware evasion.

3.1 Gradient based attacks

Since model architecture is highly similar to computer vision DNN solutions, several research groups showed that it is possible to perform successful white-box evasion attacks on MalConv, transferring adversarial techniques from the image classification domain. Kolosnjaj et al. [7] one of the first to proved that byte-level attack is effective. They confirm that injecting random bytes can yield successful attacks, nonetheless, gradient-based techniques result in a higher evasion rate. Payload type comparison is visualised in Figure 4. With the sophisticated craft of perturbations, injecting less than 1% of total bytes results in a decrease of MalConv accuracy by over 50% [7].

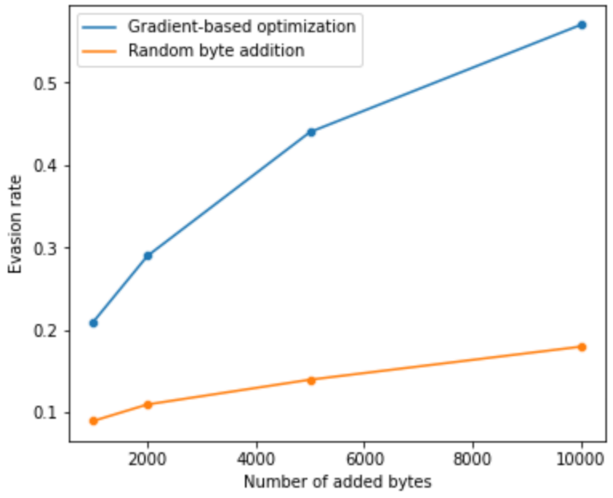

Figure 4: Evasion rate against number of injected bytes in Kolosnjaj et al. [7] experiments against MalConv (reconstructed graph).

Still, they use naive file modification, injecting perturbations as an overlay, appending new bytes at the end of the file. Such a technique is trivial to detect with a simple manual check. As opposed to that, modification of Portable Executable (PE) structure is a complex task. PE files have a specific structure with different headers and sections. The operating system expects to find specific bytes in particular places, otherwise, the operating system will not execute PE.

A more sophisticated PE modification is shown by Kreuk et al. [8], who injects adversarial bytes to unused parts of already existing sections. They utilize FGSM to generate AE and show that it is possible to achieve more than 99% evasion rate on MalConv model trained using EMBER dataset [2].

These attacks show that adversarial techniques borrowed from computer vision apply to malware classification under specific circumstances. Yet neither of methods can be directly used to evade modern anti-virus solutions, because of distinct threat models. Commercial solutions do not expose features or confidence scores used and may use hybrid architectures. However, without evaluation of commercial products and their architectures, we cannot assess the adversarial landscape in the industry.

3.2 Black-box malware evasion

The white-box attack scenario in a real-world setting is unrealistic. Attackers do not possess insider knowledge, can

[2]https://en.wikipedia.org/wiki/Zero-day_(computing)

only assume the model's architecture and features, as well as do not possess datasets used for training. For that reason, valuable research in this direction should emphasize the black-box threat model.

Numerous researchers provided insights on adversary capabilities within such a setting. Xu et al. [17] use genetic algorithms for adversarial PDF malware. Nonetheless, provided Proof-of-Concept (PoC) solution is slow, and not practical. Attack has a high cost even against low dimensional models like random forests and support vector machines (only 500 evading examples found during 6 days of algorithm execution), hence the technique might not be applicable to DNN models at all.

Hu et al. [6] were amongst the first to use Generative-Adversarial-Networks for adversarial example generation. Their solution titled *MalGAN* was tested in a black-box manner against multiple classical ML models and showed almost perfect evasion rates. However, a crucial drawback of their approach is to perform attacks purely in feature vector-space, without reflection back to the actual PE file. MalGAN results cannot be used directly, and further adjustments to this technique are needed.

Anderson et al. [1] use reinforcement learning (RL) for adversarial example generation. Their work is highly influential, since it defined a realistic threat model, with sophisticated PE modification techniques often borrowed by other research groups in future attack scenarios. Unfortunately, evasion metrics are weak, with only a few percent decrease in the target model's performance. Their approach was significantly improved by the MAB reinforcement learning framework from Song et al. [15], who severely reduced the number of broken binaries after adversarial modification, and increased evasion rates against MalConv and EMBER models by up to 50%.

Additionally, Song et al. [15] demonstrate that in malware domain AE evading surrogate model are unlikely to evade commercial AV system. Despite huge success against purely ML-based classifiers, their success against three commercial solutions is minuscule. It is clear that real-world threats cannot borrow techniques directly from the image classification domain here and should use a different approach to evade realistic defenses.

Supposedly, beneficial research should be based on the utilization of AV solution as an oracle, since successful incidents were seen in the real world already, for example, bypass of Cylance classifier in 2019[3].

4 DEFENSES

Defense against adversarial attacks can be (1) *proactive* making the model itself more robust to attacks, and (2) *reactive* validating input data to detect and prevent adversarial examples.

4.1 Increasing model robustness

A subset of defense techniques uses *gradient masking* idea, by constructing a model that does not have directly accessible gradients. For example, using nearest neighbor instead of DNN. This does not expose directions in which the model is sensitive, therefore preventing white-box attacks. Although Papernot et al. [12] prove that attacks based on substitute models overcome these defense techniques.

Szegedy et al. [16] suggested the use of artificially generated adversarial inputs during a training phase of a model thus making the model more robust to AE in general. Originally generation of AE was computationally expensive and this option was hard to realise in practice. But since Goodfellow et al. [4] introduced a low-cost FGSM, this option became realistic.

Additionally, there is evidence that label smoothing [10] is capable of shifting decision boundaries in adversarial directions. This property is utilized in defensive distillation. Distillation was introduced by Hinton et al. [5] to map the function of a more complex and sophisticated model to the simpler one. The second model does not need to be smaller, main idea is to train it not using "hard" labels of vector y, but "soft" labels - probabilities that the first model assigns to input data $\hat{y}$. This second distilled model is more robust to white-box attacks.

In domains other than image recognition, models rely less on the operation of convolution and implement custom pipelines with more specific feature extraction. There is evidence that specific features are easier affected by adversarial perturbations and cause most misclassification, and might be adjusted to make the full pipeline more robust. MAB framework provided by Song et al. [15] allows to identify modifications essential for evasion, hence, provides insights for future work on explainability of obscure models in the black-box setting.

4.2 Input validation

Another technique that might yield noticeable defenses against AE is a validation of input to filter out potentially perturbed examples. This idea is highly similar to conventional signature-based security validations in contemporary systems and is highly domain-specific. For example, we might want to examine input images from the camera sensor on realism, like neighbor pixel statistics and light characteristics.

In Section 3.1 we have seen that malware classification algorithms can be bypassed by appending overlay data to an executable file. As this is not a usual characteristic of any executable file to have overlay bytes, input validation

[3]https://skylightcyber.com/2019/07/18/cylance-i-kill-you/

here could either deny such an example right away or crop the overlay bytes for a model to analyze only actual code instructions.

5 FUTURE WORK

Adversarial Machine Learning has a huge research base, which grows at an increasing pace. However, while AE nature is not fully explored, it is possible to identify promising research directions and drawbacks of current works:

- Almost little to no solutions are provided to identify and segregate adversarial inputs from real. Most white-box research and practical implementations use FGSM for the generation of adversarial payload but are unclear whether it is suitable to use stealthier algorithms such as L-BFGS to avoid detection.
- During broad analysis of the research base, we have seen only attacks on classifiers. The question is what results may yield other target models, like that (a) solves regression problems, (b) performs anomaly detection. Potential adversarial goals might be to generate AE to tampering predictions of the house prices for (a) or bypassing a complex anomaly detection system to mimic usual behavior for (b).
- Focusing on AE for malware classifiers, there is a generalization across malicious software families. Model authors could benefit from a focus on tools, techniques, and procedures (TTP), influenced by for example MITRE ATT&CK matrix[4], rather than an abstract definition of malware. Such, more granular approach could yield higher robustness of malware classification.
- Mostly, malware evasion through AE is shown only for static classifiers, no dynamic heuristics are taken in place, with few exceptions like AE generation by Kucuk et al. [9]. Consequently, research outcomes still cannot be applied to modern offensive operations, to infer the capabilities of powerful adversaries like government-powered Advanced Persistent Threat groups[5], who will be able to weaponize such attacks in a real-world context.

TBD: ideas from seminar paper and filepath paper

6 CONCLUSIONS

In this paper, we have provided a general overview of adversarial machine learning. First of all, the intuition and taxonomy of known attack methods are covered. We discuss the mathematical basis of FGSM, a widely used adversarial algorithm. Furthermore, we focus on a specific class of target models, exploring evasion attacks against machine learning-based malware classifiers. A survey on the most promising techniques in this domain is provided with a brief analysis of results. Ideas with promising future work directions and existing research drawbacks are specified.

[4]https://attack.mitre.org/

[5]https://www.fireeye.com/current-threats/apt-groups.html

REFERENCES

[1] Hyrum S. Anderson, Anant Kharkar, Bobby Filar, David Evans, and Phil Roth. 2018. Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning. arXiv:1801.08917 [cs.CR]

[2] Hyrum S. Anderson and Phil Roth. 2018. EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. *CoRR* abs/1804.04637 (2018). arXiv:1804.04637 http://arxiv.org/abs/1804.04637

[3] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. 2018. Making Machine Learning Robust against Adversarial Inputs. *Commun. ACM* 61, 7 (June 2018), 56–66. https://doi.org/10.1145/3134599

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML]

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML]

[6] Weiwei Hu and Ying Tan. 2017. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. arXiv:1702.05983 [cs.LG]

[7] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. 2018. Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables. *CoRR* abs/1803.04173 (2018). arXiv:1803.04173 http://arxiv.org/abs/1803.04173

[8] Felix Kreuk, Assi Barak, Shir Aviv-Reuven, Moran Baruch, Benny Pinkas, and Joseph Keshet. 2019. Deceiving End-to-End Deep Learning Malware Detectors using Adversarial Examples. arXiv:1802.04528 [cs.LG]

[9] Yunus Kucuk and Guanhua Yan. 2020. Deceiving Portable Executable Malware Classifiers into Targeted Misclassification with Practical Adversarial Examples. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy* (New Orleans, LA, USA) *(CODASPY '20)*. Association for Computing Machinery, New York, NY, USA, 341–352. https://doi.org/10.1145/3374664.3375741

[10] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2020. When Does Label Smoothing Help? arXiv:1906.02629 [cs.LG]

[11] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. arXiv:1605.07277 [cs.CR]

[12] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. arXiv:1602.02697 [cs.CR]

[13] Edward Raff, Jon Barker, Jared Sylvester, Robert Brandon, Bryan Catanzaro, and Charles Nicholas. 2017. Malware Detection by Eating a Whole EXE. arXiv:1710.09435 [stat.ML]

[14] Matthew Schultz, Eleazar Eskin, F. Zadok, and Salvatore Stolfo. 2001. Data Mining Methods for Detection of New Malicious Executables. *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 38–49. https://doi.org/10.1109/SECPRI.2001.924286

[15] Wei Song, Xuezixiang Li, Sadia Afroz, Deepali Garg, Dmitry Kuznetsov, and Heng Yin. 2021. MAB-Malware: A Reinforcement Learning Framework for Attacking Static Malware Classifiers. arXiv:2003.03100 [cs.CR]

[16] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. arXiv:1312.6199 [cs.CV]

[17] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically evading classifiers. In *Proceedings of the 2016 network and distributed systems symposium*, Vol. 10.