Machine Learning Security Survey

Turhan Kimbrough
Department of Computer Science
Towson University
Towson, Maryland
tkimbr1@students.towson.edu

***Abstract*—Machine learning has experienced a significant growth in usage over the past few decades. Due to its data-centric approach in modeling, machine learning has seen use in a variety of subfields in computer science. In particular, researchers have been interested in incorporating machine learning into the domain of cybersecurity, utilizing it from the perspective of an adversary or ally. Researchers have also been concerned with the security state of current machine learning models. This survey paper provides an overview of machine learning, attacks and mitigations to current machine learning systems, machine learning use cases for adversaries, and future research directions.**

***Keywords*—Neural networks, security, classification, machine learning model, training, validation**

I. Introduction

With the increasing widespread adoption of machine learning technology, its usage is being observed in a variety of different fields. Some notable examples include image recognition, voice assistant technologies, email spam filters, and search engines. Much of its recent popularity can be attributed to the availability of frameworks such as TensorFlow, a python-based library which provides easy-to-use bindings for complex operations.

However, one growing concern tied to the ubiquity of machine learning is its accessibility to adversaries. Based on the assessment of current and prior research, there are a number of vulnerabilities in current machine learning models which can be exploited with little knowledge of a system's domain. In addition, attackers have been able to leverage machine learning technology to assist with the deployment of cyberattacks. Enterprise machine learning applications may often contain large datasets of important information, becoming a potential candidate of a targeted attack. This paper will survey the domain of cybersecurity with regard to machine learning. This topic will explore the fundamentals of machine learning, current attacks on machine learning systems, mitigations to machine learning attacks, machine learning technology from the perspective of adversaries, and future research directions.

II. Overview

This section will introduce the fundamentals of machine learning. Additionally, there will be a discussion about the key terms used by members of the community.

A. Machine Learning Basics

Traditionally, when software developers are tasked with solving a problem, they use a combination of rules and logic to find a solution. The basic routine consists of finding appropriate input values, creating the logic and rules to process the input, and producing the appropriate output. The traditional approach to software development allows for fine-tuned control of program behavior to achieve the solution. However, this approach does not scale with the complexity of additional rules and/or possible solutions. An example is image classification, where the logic needed to compare images is complex. This becomes a bigger concern when new classifications need to be derived with new data. Machine learning flips the traditional programming paradigm on its head, by taking a series of solutions as input, and letting the machine develop the rules by detecting patterns in the solutions [1]. The result is a self-propagating mathematical model, capable of making decisions on newly supplied data. This is what is known as a *machine learning model*, a software component which can be saved to a file and integrated into larger software systems [1].

Machine learning models can be implemented using a number of different algorithms, some popular examples will be discussed in this paragraph. *Regression algorithms* use statistics to model the relationships between different variables plotted on a graph [2]. *Decision tree algorithms* represent the problems using trees, where internal nodes represent conditional statements and leaf nodes represent decisions [3]. Lastly, *artificial neural*

networks are used in a subset of machine learning known as *deep learning*, where the algorithms are modeled to mimic the structure of biological neural networks found in animal brains [4].

Briefly mentioned earlier, a common use case for machine learning is *classification*, where a dataset is categorized into different groups based on one or more *features*. A *feature* is defined as some measurable property or characteristic being observed in a dataset [1]. There are several types of classifications, including *binary classification*, *multi-class classification*, and *multi-label classification*. *Binary classification* categorizes data based on whether a feature is present or not, resulting in an outcome of true or false [1]. *Multi-class classification* categorizes data into different groups, where each data instance is assigned according to a single feature [1]. *Multi-label classification* categorizes data into different groups, where each data instance is assigned according to its expression of two or more features [1]. *Learning* is the procedure for training algorithms to detect patterns in data.

There are three main types of training mechanisms, *supervised training*, *unsupervised training*, and *reinforcement learning*. *Supervised training* requires each data instance to have one or more labels, defining which category or feature it expresses [5]. In contrast, *unsupervised training* omits the need for labels, and the machine learning model will categorize datasets on its own [5]. *Reinforcement learning* uses a reward-based system to maximize good decisions when a software agent performs a task [5]. Typically, after the training phase, a machine learning model will go through the process of *validation*. *Validation* typically consists of a separate dataset to guarantee the accuracy of a model and preventing a phenomenon called *over-fitting*, where a model will only 'memorize' characteristics or patterns of training data [5].

III. Attacks to Machine Learning Models

Due to the unique mechanism for constructing machine learning models, there are several ways which they can be exploited by an attacker. As a result, the attack surface which exists on machine learning models will differ from traditional software systems. This section will discuss several attack types which target machine learning systems.

A. Data Poisoning

Data Poisoning is the act of manipulating, removing, or adding data during the training phase of machine learning [6]. This type of attack is known as a *black box attack*, where an attacker does not need to know the implementation of a system to attack it.

One popular instance of a data poisoning attack is the adversarial example. This requires the attacker to have some knowledge of the training data, such as its dimensions and data type. The attacker would then manipulate data instances in a way where the machine learning model would be fooled, but appears normal to a human observer [6]. The reason for manipulating data instances in this way is two-fold. First, machine learning models are often constrained to data fitting a specific dimension, shape, size, or length of characters. This is typically done to prevent incompatible data from entering the model during training. Second, there is often one or more people who are observing the model with testing or validation datasets. An attacker would want to minimize any evidence of the data being tampered with.

Often, the goal of this attack is to make a machine learning model incorrectly classify data. The consequences of this attack can be devastating.

Researchers at [17] use adversarial examples to trick a real-world image classification model. The researchers used variations of image perturbations, where noise and/or pixel values are slightly modified to affect the detected features in an image. As a result, images were misclassified even when the disturbances were applied to images originating from a smartphone camera.

B. Membership Inference

Membership inference is a mechanism of data extraction, where an attacker intends to know whether certain samples were used as training data for a machine learning model [7]. This type of attack is classified as a *gray box attack*, where the attacker has some knowledge of the software system.

Researchers at [18] evaluated publicly available models to detect the presence of data characteristics in hypothetical datasets. The researchers built several 'shadow' models using parameters that closely resembled the target machine learning model. The shadow models would take the output prediction of the target model as inputs, and distinguish the characteristics between how the target evaluated training data and new data. As a result, the shadow models were able to detect when the target exhibited over-fitting, indicating a data member was part of the training set.

Often, this can exploit the confidentiality of information on a system. An attacker has the capability to cor-

relate information between datasets to target individuals for other cyberattacks.

C. Transfer Learning

Transfer learning is a mechanism where an attacker has the ability to study a publicly available machine learning model, and use that insight to sneak past and/or corrupt similar target systems [8]. This type of attack is classified as a *white box attack*, since the attacker would need to have full access to at least one machine learning model.

This particular attack requires knowledge of the input data, learning mechanism, and output behavior of a machine learning model similar to the target. Once this information is obtained, the attacker would test the system and learn its overall behavior to enumerate flaws in the model's logic [8]. The assumption is that the flaws found in the available model's logic would appear in a target system.

Often, the goal of this attack is similar to the other two mentioned above. An attacker would use the information to trick, fool, manipulate, or sneak passed a target machine learning model.

Researchers at [19] use the transfer learning technique to launch backdoor attacks on a target machine learning model. The researchers used insight from the access they had to a neural network structure closely resembling the target. The attack exploited the presence of extra neurons in the target model, which affect the classification of the model when activated. Using carefully constructed input data, a backdoor attack was launched causing a misclassification of images and time series data.

IV. Mitigations to Attacks

Due to the large attack surface found in machine learning models, there has been a significant focus in its security. This section will discuss several mitigations to the attacks discussed in the previous section.

This section will also reference a generic process known as the *machine learning lifecycle*. This lifecycle consists of building the dataset, feeding the input to the model, training the model, verifying the model accuracy, deployment, and maintenance.

A. Sanitizing Input

One mechanism for securing a machine learning system is to filter malicious samples from clean samples during the training phase. There are several different approaches to sanitizing depending on the type of data being fed to the model. A general procedure for sanitizing input data is still an open research topic.

One naive approach for filtering data is the use of *gradient masking*. This simple technique will manipulate each input sample to create a sharper decision boundary for the machine learning model to work on. A common implementation of this technique for images is *binary thresholding*, where each pixel in an image is converted to black or white depending on its color value. This technique will mitigate against perturbations to data [9].

Another approach called ANTIDOTE, proposed by [11] uses an *anomaly-based* detection scheme to characterize data samples. ANTIDOTE uses statistics to differentiate between normal and malicious samples. Once a malicious sample is detected, it is automatically discarded from the model.

B. Machine Learning Retraining

After a machine learning model is finished with its initial training and verification, it is typically ready to be deployed. After deployment, a good practice to maintain a machine learning model is to periodically retrain it. Retraining can either use the original dataset, or a new dataset containing similar information as the original. This is ideal for machine learning models which continuously learn post-deployment.

A specialized approach to retraining using samples of malicious data is a technique called *adversarial retraining*. This process requires a sufficient number of malicious samples with perturbations, along with the original training data. The technique involves labelling malicious data with perturbations as adversarial samples, and training the model to detect them appropriately [10]. This ensures that adversarial samples will not affect the overall accuracy of the model, by learning to filter perturbations added to any new data [10]. The model will learn to differentiate between legitimate and adversarial data when re-deployed. ADD RESEARCH HERE

C. Differential Privacy

During deployment, data will typically be flowing in and out of a machine learning model. Information coming in and out of a machine learning model may be sensitive, so systems will typically incorporate APIs to interface with it. Even if a system is robust, a data leak will completely compromise confidentiality. To circumvent this issue, *differential privacy* is a mechanism to effectively store information while hiding confidential details [12].

The implementation of differential privacy will differ depending on the data being processed. A standard mechanism for differential privacy is to present output

data showing non-sensitive information normally, while sensitive information is encrypted. Another mechanism is presenting the output information of a machine learning model as a reference to another dataset, which will then need to be queried with subsequent processing. ADD RESEARCH HERE

V. Assisting Cyberattacks with Machine Learning

After discussing the attacks/mitigations for machine learning systems from the perspective of the defender, this section will discuss how machine learning technology is used to assist with the deployment of cyberattacks from the perspective of the attacker.

A. Evasive Malware

Modern-day malware detection schemes use several different techniques to classify malicious and safe programs. One method is the use of detection engines, which are often available in two distinct types, *signature-based* detection engines and *anomaly-based* detection engines. A signature-based detection engine keeps a record of malicious behaviors and footprints, comparing software behaviors against the record to identify malware [13]. On the other hand, an anomaly-based engine will compare software behaviors against a record of 'normal' system behavior to detect deviations, classifying them as malicious actions [13]. Both approaches can be extended with machine learning capabilities to improve their detection rates. In particular, supervised learning has been shown to be an effective strategy.

However, even with the advancements in malware detection schemes, research has shown that there can be a significant flaw associated with detection engines using static features and definitive labels. In particular, researchers at [13] created *DQEAF*, a framework which has demonstrated the ability to evade malware-detection by the use of reinforcement learning. DQEAF operates by using a machine learning agent to interact with different malware samples. During its interaction, it will slightly modify the behaviors of detected malware samples without impacting its overall structure and/or functionality. The agent will continuously modify detected malware samples until it is able to be completely undetected by a malware detection engine.

B. CAPTCHA Bypass

CAPTCHA is an acronym which stands for the "Completely Automated Public Turing test to tell Computers and Humans Apart". CAPTCHAs are a mechanism to combat against the growing number bots which are used on the internet, ideally providing a challenge which is relatively easy for humans to solve, but difficult for machines. CAPTCHAs come in a variety of types, including the deciphering of obfuscated text, interpreting audio messages, selecting images based on descriptions, and the tracking of end-user behavior.

Over the past few years, CAPTCHAs have grown to be more difficult due to the techniques used by adversaries. The techniques include the exploitation of weaknesses in the CAPTCHA generating mechanism and using machine learning for CAPTCHA solving. Machine learning CAPTCHA solving is an especially attractive option, given the automation capabilities. In fact, researchers at [14] suggested that reinforcement learning has provided the capability to solve one of Google's reCAPTCHA mechanisms, which records mouse movements to distinguish human and bot behavior. This was achievable by representing the pixels on a screen as a matrix, and using a software agent to move the cursor across the screen. The *Markov Decision Process* is the algorithm which was used to generate random movements, and trains the software agent to behave more human-like over time. This process solved reCAPTCHAs with greater than 90% accuracy.

C. Brute Forcing Passwords

One of the most prevalent cybercrimes in today's age is account hijacking. From banking, social media, streaming, and more, digital accounts play a major role in the lives of everyday people. Adversaries are interested in gaining access to these accounts in an effort to unravel useful information. With the works of prior research and the open source community, there are a significant number of tools available for account hijacking. One of the simplest tools are *brute force password-crackers*, which typically employ a random combination of characters to guess the resulting password.

Due to the increasing awareness in cybersecurity best practices, many sites with accounts will employ mechanisms to enforce stronger passwords. As a result, simple password crackers are no longer able to make guesses within a reasonable amount of time. However, researchers at [15] suggest that a generative machine learning model is capable of creating more accurate guesses based on previously exposed password datasets. The approach uses the *Markov Model* with a neural network to derive characteristics from a password dataset, and generate new passwords with the same characteristics. Comparing the generated passwords against a

separate leaked dataset, the researchers were able to match up to 42% of the passwords using the generative machine learning model.

VI. Future Work

Much of the research which has been conducted in the field of machine learning security has been relatively new. As a result, the research community still has many questions as to what components are necessary and/or feasible for securing future machine learning systems. This section will outline potential directions for future research based on what has been reviewed in prior sections.

A. Trusted Platforms

Currently, the usage of *Trusted Platform Modules* (TPMs) are seeing a rise in popularity for securing systems. TPM technology typically consists of a set of cryptographic algorithms along with a dedicated processor known as a TPM chip. The purpose of a TPM chip is to provide a dedicated hardware device to independently verify that a software system has not been tampered with.

An interesting research direction would be to incorporate TPMs into a machine learning system. In addition, there has been a recent surge in mobile devices which ship with dedicated machine learning processing units, such as Apple MacBooks powered by M1 processors and Samsung Galaxy S-series flagship devices. It would not be surprising if researchers would be interested in using a TPM and machine learning processing unit as building blocks for a trusted platform.

B. GAN-assisted Model Retraining

The concept of Generative Adversarial Networks (GANs) is still a relatively new, being discovered only within the last decade. Generative Adversarial Networks consist of two separate neural networks, a *generator* and *discriminator*. Given an initial training data set, the goal is to produce new data which mimic the characteristics of the training set. This is accomplished by using the *generator* to produce realistic fake data from a random seed, and the *discriminator* which learns to differentiate the fake data and real data [16]. The two networks are in an adversarial relationship, hence the name, and work toward minimizing the differences between the fake and real data [16]. This concept has brought significant insight into the inner-workings of machine learning, along with insights on the relationships between data samples.

An interesting research direction would be to use Generative Adversarial Networks to aid in model retraining. In order to do this, a variety of malicious data samples would need to first be aggregated. Then the *generator* and *discriminator* pair would work to create a model which can realistically create new malicious data samples. This new (malicious) model can then be used to generate labeled malicious data, which will be fed to machine learning model which needs to undergo retraining. The machine learning model which is being retrained will have a more reliable malicious dataset, which can strengthen its detection mechanism.

C. Reinforcement Learning Framework for Cyber Defense

One of the biggest challenges in cybersecurity is finding a general software solution which can defend against new threats. Especially in the context of *zero-day exploits*, which are cyberattacks that occur before a software vulnerability is well-known to the public. Additionally, even known malware samples may pass through a system undetected due to the nature of anti-malware systems.

An interesting research direction would be to incorporate a general framework for securing a software system through the use of reinforcement learning. This type of learning would be ideal in a live software environment due to the near-infinite number of possibilities for which it can be attacked. Specifically, it would act more as an aid to patching software vulnerabilities, suggesting where they could occur based on internet data and assessments.

VII. Conclusion

In this paper, the topic of machine learning security from different perspectives was explored. From the perspective of someone new to machine learning, an overview was given. From the perspective of a defender, the vulnerabilities and mitigations for machine learning systems was given. The perspective of adversarial use-cases for machine learning was also portrayed. Lastly, prospective work for researchers have been included as well. It's clear that machine learning is the way forward for many software solutions, and it is important to consider how its ubiquity will affect the domain of cybersecurity.

References

[1] E. Alpaydin, Introduction to machine learning, MIT press, 2014.

[2] E. Gambhir, R. Jain, A. Gupta and U. Tomer, "Regression Analysis of COVID-19 using Machine Learning Algorithms," 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 65-71, doi: 10.1109/ICOSEC49089.2020.9215356.

[3] R. C. Barros, M. P. Basgalupp, A. A. Freitas and A. C. P. L. F. de Carvalho, "Evolutionary Design of Decision-Tree Algorithms Tailored to Microarray Gene Expression Data Sets," in IEEE Transactions on Evolutionary Computation, vol. 18, no. 6, pp. 873-892, Dec. 2014, doi: 10.1109/TEVC.2013.2291813.

[4] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, Neural network design, Martin Hagan, 2014.

[5] W. Grant and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends; A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," 2018, doi: 10.1109/ACCESS.2018.2830661.

[6] C. Liu, B. Li, Y. Vorobeychik, and A. Oprea, "Robust linear regression against training data poisoning," in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 91-102: ACM.

[7] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 3-18, doi: 10.1109/SP.2017.41.

[8] B. Wu, A. Shuo Wang, A. Xingliang Yuan, C. Wang, C. Rudolph, and A. Xiangwen Yang, "Towards Defeating Misclassification Attacks Against Transfer Learning."

[9] Y. Yanagita, M. Yamamura, "Gradient Masking Is a Type of Overfitting," 2018 International Journal of Machine Learning and Computing. 8. 203-207. 10.18178/ijmlc.2018.8.3.688.

[10] S. H. Silva and P. Najafirad, "Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey."

[11] B. Rubinstein, B. Nelson , L. Huang, J. Anthony, L. Shing-hon, N. Taft, J. Tygar, "ANTIDOTE: understanding and defending against poisoning of anomaly detectors," 2009, Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC. 1-14. 10.1145/1644893.1644895.

[12] M. Abadi et al., "Deep Learning with Differential Privacy," 2016, doi: 10.1145/2976749.2978318.

[13] Z. Fang, J. Wang, B. Li, S. Wu, Y. Zhou and H. Huang, "Evading Anti-Malware Engines With Deep Reinforcement Learning," in IEEE Access, vol. 7, pp. 48867-48879, 2019, doi: 10.1109/ACCESS.2019.2908033.

[14] I. Akrout, A. Feriani, and M. Akrout, "Hacking Google reCAPTCHA v3 using Reinforcement Learning."

[15] B. Hitaj, P. Gasti, G. Ateniese, F. Perez-Cruz, "PassGAN: A Deep Learning Approach for Password Guessing," 2017

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Networks," 2014, Advances in Neural Information Processing Systems. 3. 10.1145/3422622.

[17] A. Kurakin, I. Goodfellow, S. Bengio, "Adversarial examples in the physical world," 2016

[18] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 3-18, doi: 10.1109/SP.2017.41.

[19] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen and T. Chen, "Backdoor Attacks against Transfer Learning with Pre-trained Deep Learning Models," in IEEE Transactions on Services Computing, doi: 10.1109/TSC.2020.3000900.