Analysis of Security of Machine Learning and a Proposition of Assessment Pattern to deal with Adversarial Attacks

Asmaa Ftaimi[1*], and *Tomader* Mazri[1]

[1]Laboratory of Electrical and Telecommunication Engineering, Ibn Tofail science University, Kenitra, Morocco

Abstract. Today, Machine Learning is being rolled out in a variety of areas. It is a promising field that can offer several assets and can revolutionize several aspects of technology. Nevertheless, despite the advantages of machine learning technologies, attackers can exploit learning algorithms to carry out illicit activities. Therefore, the field of security of machine learning derives attention in these times to meet this challenge and develop secure learning models. This paper will overview a taxonomy that will help us understand and analyze machine learning models' security. In the next sections, we conduct a comparative study of the most widespread adversarial attacks. Then, we analyze common methods that were advanced to protect systems built on Machine learning models from adversaries. Finally, we discuss a proposition of a pattern designed to ensure a security assessment of machine learning models.

1 Introduction

Nowadays, we are in the golden era of Machine Learning. In fact, Machine learning has become a promising field of artificial intelligence and one of the most rapidly growing technologies. There are several ways where machine learning has been proving its potential. Today, it is widely applied in many applications and is used to solve many real-world complex problems, which cannot be resolved with traditional methods. For instance, virtual assistants integrate ML models to improve the user experience by refining interactions based on previous involvement. Social media platforms are utilizing machine learning as well as toward personalizing the user's news feed and customizing advertisements targeting. Traffic prediction applications are also powered by ML models that help in estimating the regions where congestion can be found based on daily experiences [1].

However, Machine Learning techniques are held back by the challenges and obstacles that it still encounters, mainly in security. Indeed, it has been shown that Machine learning algorithms are vulnerable. In 2017, Papernot et al. [2] showed that remote adversarial attacks could be performed on ML models without access to the algorithm or the training dataset. Many studies have been conducted on the field of adversarial machine learning; some of them are theoretical and aim to build frameworks and threat models for secure learning, while others are merely experimental and try to design attacks with concrete examples in order to prove that adversaries can maliciously exploit machine learning models during training or inference phases [3]. Therefore, these observations have motivated researchers to explore the security of ML models under the classical security approach of confidentiality, integrity, and availability (CIA). A group of researchers, including Barreno et al. [4], has implemented a taxonomy related to the security in Machine Learning systems that provides a classification of adversarial attacks against models depending on the attacker's goal, knowledge, and capabilities. Recent research has also focused on developing mitigation techniques to address malicious attacks [5]. Defensive mechanisms implemented for attacks against ML models are mainly founded on two approaches that consist of either cleansing the data used in the training and the inference phase or increasing the model's complexity to make it more robust towards adversarial attacks [6]. Nevertheless, before choosing defensive methods that can be employed against adversarial attacks, an assessment of the targeted system's security is required. In this context, we analyzed the existing security assessment patterns, although we noticed a significant shortage of resources in the literature. For this reason, we have designed in this research paper a pattern for evaluating the security of a machine learning built system inspired by the existing computer security standards. This pattern performs a proactive and curative security assessment of the model and uses an impact estimation and risk prioritization to establish a mitigation strategy to deal with the targeted system's vulnerabilities.

This paper will go through the different security aspects in machine learning to establish the security assessment pattern that will help us design the appropriate defensive strategy for the targeted model and the eventual vulnerabilities that they may enfold. Below, we will introduce a threat model that many researchers have detailed in their works. In the next

* Corresponding author: a.ftaimi@gmail.com

section, we will conduct a comparative study of the most common adversarial attacks targeting systems built on machine learning models, and we will present an analysis of the most known defensive mechanisms. Finally, we will construct a pattern for security assessments in both proactive and curative approaches.

2 Threat Model

While analyzing the security of Machine Learning systems, it is essential to identify a threat model that would help us better understand adversarial attacks and, therefore, develop effective defensive mechanisms against them [7]. The threat model considers the attacker's goals and the means at his disposal to perform the attack. It also examines the potentials and aptitudes that the adversary possesses and the strategy he follows to attack Machine Learning models [8].

2.1 Attacker's Goal

Whereas the attackers are not promoted by the same motivations, they all share the same goals, which are all converged around three objectives: espionage, sabotage, or fraud [6] as presented in the following table:

Table 1. Attacker's goal in adversarial learning.

Goals	Methods	Damages
Espionage	Leakage sensitive information	Compromising the confidentiality and the privacy of the system
Sabotage	denying normal operations	Compromising the availability of the system
Fraud	Injecting malicious data samples	Compromising the integrity of the system

2.2 Attacker's goal in adversarial learning

Whatever the adversary's incentives are, the success rate of the attack is more maximized when he has a fuller knowledge of the targeted system [9]. Furthermore, the method adopted by the attacker is highly dependent on his level of comprehension of the targeted system to perform his attack. The more he is informed about the dataset used to train the system in the learning phase, or the better he understands the model features, the more he constitutes a real danger. Papernot et al. [2] have established a classification of the attacker's knowledge, which can be presented in three levels as illustrated in Figure 1.:

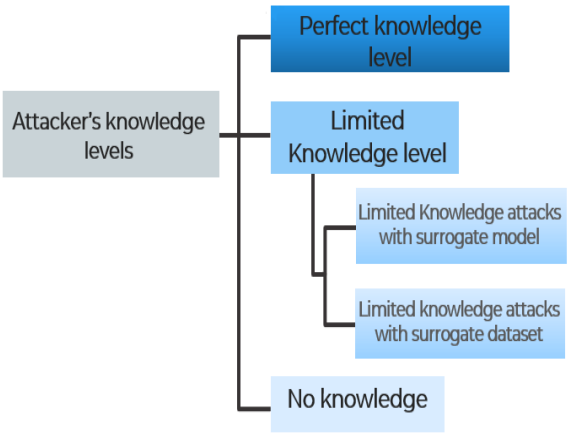

Fig. 1. Levels in the attacker's knowledge

2.3 Attacker's Capabilities

In addition to knowledge of the target system, the adversary's objectives, the attacker's capability is directly instrumental in determining the method, and the approach chosen for executing the attack. In fact, the means and potentials that the adversary possesses significantly affect the attack's success rate [8]. The leverage of the attacker on the targeted system can be classified according to three main axes as in Figure 2.:

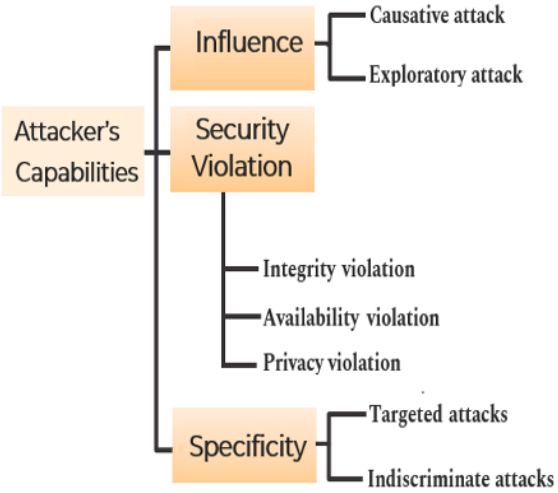

Fig. 2. Classification of adversary's capabilities

2.4 Attacker's Strategy

Based on the elements seen below that define the threat model, we can infer that the attacker's strategy is merely an optimization problem that considers all the aspects examined in the previous section. Indeed, if we assume that the prediction function built by the algorithm during the learning phase disposes of a determined cost for each prediction error, we can conclude that the adversary's strategy is to maximize the prediction function's cost via false positives or false negatives [10].

Considering a hypothesis $h\theta(X)$ representing the prediction function, every output value is calculated by the prediction function based on the features X and the parameters θ. The difference between the true output Y and the one forecasted by the prediction function $h\theta(X)$

is represented by a cost function $\mathscr{L}(\theta,X)$. The attacker aims to maximize the cost function by determining the parameters and features that would achieve that value.

$$Arg\ max\ \mathscr{L}(\theta_i,x_i),\ \theta_i \in \theta_i, x_i \in X \quad (1)$$

This problem can be viewed from another point of view, i.e., the attacker seeks to find infinitesimal perturbations ε that can be added to the inputs to generate a value different from the expected.

$$h\theta\ (X + \varepsilon\ ) \neq h\theta\ (X) \quad (2)$$

Thus, the attacker seeks to find the optimal and adequate method to successfully achieve his malicious goals while considering his knowledge of the target system and the potentials and the means at his disposal to perform the attack [5].

3 Comparative study of Adversarial Attacks

Several studies have focused on analyzing potential effects that may reside in the flexible aspect of machine learning models. These vulnerabilities have given rise to several attack experiments that have been published in many research papers [11]. This section will explore the most widespread attacks that target systems built on machine learning models. Then we will conduct a comparative study based on the impact and timing of each attack. Attacks against machine learning models can mainly be classified into two types according to the time when they occur:

3.1 Attacks at Training Time

The training phase is primordial in the life cycle of a classification or regression model. During this phase, the algorithm trains on labeled input data to detect patterns between the input data and the target outputs. These patterns enable the model to generalize the observed associations between the training data and their labels on new data sets. However, this phase can be misused by an attacker who can introduce well-designed malicious data to influence the model's behavior and thus push it to make inaccurate predictions [12]. This type of attack, also known as poisoning attacks, is a category of attacks that can be considered as one of the most widespread [13]. It occurs in the training phase, and it aims to inject malicious data in the training phase [14]. This attack usually introduces false negatives in the training set and, as a result, generates a huge impact on the model by altering its behavior and inducing it to make inaccurate predictions. Barreno et al. [4] have experimented with this attack against SpamBayes filter by injecting ham emails containing a specific set of words that resemble spam emails. They succeeded in fooling the SpamBayes filter to misclassify 36% of benign messages to malicious ones by poisoning only 1% of the training dataset [15]. The results obtained by Shen et al. [16], according to poisoning attacks, are impressive. In fact, they achieved a success rate of 99% in collaborative deep learning models by taking control over 10% of the training dataset.

This attack was performed by employing a new approach to affect the model's ability to produce accurate predictions. This attack, also known as trojanning, has gained momentum due to the approach that most companies are following to build models adapted to their needs [5]. Today, most companies do not build models from scratch but download existing models from the internet and train them on new datasets to align them with their business goals [17–19]. This gives attackers an important opportunity to replace the uploaded models on the internet with their own modified version, where they can introduce additional hidden behavior for specific instances without altering the normal behavior expected for other instances. Liu et al. [20] developed an experiment of this attack against some models used for image recognition and successfully misled the model with a high confidence level.

Another recent attack was derived from poisoning attacks and known as backdoors attack. It looks similar to trojaning attacks and aims to inject additional hidden and malicious behavior into the model and keep this behavior even after recycling the algorithm with a new dataset. TianyuGu et al. [21] have tested this attack on models implemented in traffic sign detectors commonly used in automated vehicles. They showed that they could induce the backdoored model to detect them as speed limit signs with a 94,7% confidence level by inserting a simple yellow post-it note in a stop sign. This attack also showed impressive results after recycling the backdoored model with Swedish road signs. Indeed, the compromised model kept other malicious behavior even after being recycled with a new dataset.

3.2 Attacks at Inference Time

3.2.1 Evasion Attacks

It is one of the most widespread attacks that has gained the interest of several researchers. Indeed, there are several research studies about exploratory attacks. This attack occurs in the model's inference phase and aims to introduce adversarial inputs or perturbations that derive the model to produce incorrect classifications. This attack has a huge impact on the targeted system. In fact, it can push the model to produce incorrect outputs with a high confidence level. To test this attack, Abadi et al. [22] conducted an experiment of adversarial example by employing DREBIN, a malware detector often used in the android operating system, as a targeted classifier.

In this specific evasion attack, Abaid et al. have changed some features in malicious applications to design false positives and, as a result, evade detection by linear classifiers with a success rate of 100%. Several works have used exploratory attacks to develop attack types capable of achieving a high success rate. Elsayed et al. [23] carried out a new type of exploratory attack against Deep Neural Network classifiers that were designed and trained initially for image recognition and managed to reprogram it into a model that calculates the number of squares in an input image. Therefore, they showed that it is possible to perform successful

adversarial reprogramming attacks with a lower precision even with a small adversarial program. The premise of this attack is based on reprogramming the model to perform new tasks using specific inputs. Indeed, this attack aims at redirecting the model towards the adversary's objectives [24].

3.2.2 Privacy Attacks

This attack aims to compromise the confidentiality of the target system. The adversary tries through particular methods to reveal sensitive information related to the learning process. Generally, there are three main types of attacks in this category, including the membership inference attack when the attacker tries to determine whether a specific instance belongs to the data set used during the algorithm's training phase [25]. The malefactors can extract all the data used to train the algorithm by performing Input inference attacks. This type of attack is the most frequent one currently, as it is easy to perform. The parameter inference is the last type of privacy attack, and it is considered the less common one. In this scenario, the adversary attempts to determine the model used in the targeted system or its features to plan further severe attacks [26,27].

3.2.3 Comparative Analysis of Common Attacks

The study of attacks involves a comparison by criteria of timing, frequency, and impact on the targeted system, as shown in Table 2. This analysis is carried out based on publications on attacks in research papers. It is worth noting that poisoning and evasion attacks are the most popular ones. Their high frequency is mainly due to their efficiency since they can cause a large degradation of performance in various models with minimal perturbations. Besides, these attacks can be executed in a black-box environment. Moreover, Moosavi-Dezfooli et al. [28] have computed nearly imperceptible universal perturbation noise in Deep Neural Networks, leading to image misclassification with a high success rate. These perturbations are doubly universal, both for data and Neural Network architectures. The universal perturbations computed for the DNN framework presented more than a 90% success rate.

The privacy attack is also common in the literature. There are various ways to execute such an attack. Indeed, the attacker can expose the properties of the model in Parameter Inference attacks or may determine the membership of an element to the training dataset and completely extract a part of the training data set. The privacy attack allows extracting sensitive information from the training dataset to plan further evasion and poisoning attacks. This attack can produce very harmful results, as illustrated by the experiment performed by Fredrikson et al. [29] against machine learning models to access a database collected by the International Warfarin Pharmacogenetics Consortium (IWPC), which contains personal information and clinical histories for thousands of patients around the world. Surprisingly, they have reached a 70% rate of accuracy in the information disclosed.

Trojanning and Backdooring attacks are less common since they are very recent. They have essentially the same principle and the same impact, enabling the generation of specific perturbations called triggers that mislead the model yet maintain its normal behavior for others input data. The only difference between Trojanning and Backdooring is that the latter retains the abnormal behavior for the triggers even after recycling the model with a set of data. Tianyu Gu et al. [21] demonstrated that the backdoors built into a traffic sign detector remained active even after recycling the system to identify Swedish traffic signs instead of American traffic signs. The reprogramming attack is also recent and innovative as it allows redirection of the model to new tasks by injecting carefully designed inputs in the inference phase. This attack allows the opponent to exploit the resources of the targeted system to achieve its tasks. Therefore, Research in adversarial learning is in perpetual development, and new attacks will likely appear, hence promoting new efficient defensive methods [30,31].

Table 2. Comparative table of most common adversarial attacks

Attack	Frequency	Time	Taxonomy	Impact
Adversarial examples	Widespread	Inference	Evasion attack	Misleading the classifier with specifically designed inputs
Reprogramming	Recent	Inference	Evasion Targeted attack	Repurposing the model to new tasks
Privacy Attack	Common	Inference	Evasion Indiscriminate attack	Leaking sensitive information about the model
Poisoning attacks	Widespread	Training	Causative attack	Modifying the model's behavior
Trojaning attacks	Recent	Inference	Evasion Targeted attack	Affecting the model's behavior and his ability to make accurate predictions
Backdoors	Recent	Inference	Evasion Targeted attack	Altering the model's behavior for specific triggers even after being recycled.

4 Analysis of defensive mechanisms

The alarming situation of machine learning security has engendered a growing interest in developing a defensive mechanism. This field of study focuses on analyzing attacks and developing the required tools and methods to guarantee the model required tools and methods to guarantee model security [32]. In fact, it aims to ensure secure learning and high performance of the model even under adversarial conditions. The first works were related to poisoning and evasion attacks and have been mainly directed towards two important approaches that can be summarized as either removing the malicious data from the training set [33] or making the model more complex to avoid the influence of malicious data on its behavior [3,34,35] as shown in Figure 3.

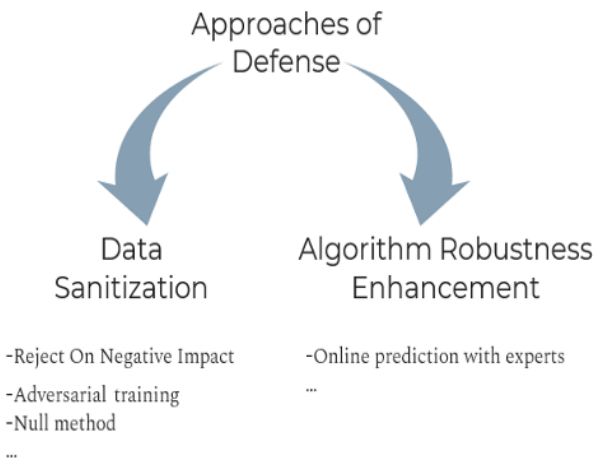

Fig. 3. Approaches followed for developing mitigation techniques

Concerning the first option, the RONI (Reject On Negative Impact) method was designed by researchers specifically to optimize the training set. The RONI method takes advantage of the iterative aspect of the training phase to measure the influence of each iteration on the algorithm's performance and removes the data instances that have a negative impact on the algorithm's ability to make accurate predictions. This technique of data cleansing protects the algorithm from poisoning attacks where the adversary introduces malicious inputs. The effectiveness of this method has been evaluated by Barreno et al. [4] in their research related to Spam filters and has proven its relevance. However, this technique remains unsatisfactory for performance evaluation. Furthermore, these approaches were used to defend against other types of attacks. Indeed, several research works have been conducted around mitigation techniques for evasion attacks that are mainly based on optimizing the training dataset's treatment. For instance, adversarial training is a widespread mitigation technique that consists of incorporating malicious data into the training set to adapt the algorithm to the malicious data that can eventually be injected by an attacker willing to change the model's behavior. Thus, once the process of adversarial training is accomplished, the model acquires the ability to distinguish benign and malicious data and can easily ignore the latter and exploit only the data collected from trusted sources. Although this method's effectiveness in building robust models, its application remains restrained by the high cost required to generate contradictory examples [36,37].

Nevertheless, parallel to this technique, the Null method is another empirical defense that has proven its effectiveness and simplicity of implementation. Indeed, in this technique, the model simply abstains from defining labels for input data that it is incapable of classifying. Thus, instead of forcing the model to predict the output data, it simply defines the NULL class as a label. Hosseini et al. [38] have suggested a three-step approach to applying this technique and tested it on images from MINST where they introduced perturbations and calculated the probabilities of attribution each label. They demonstrated this technique's effectiveness, which considerably improves the model's resistance to adversarial attacks without diminishing its ability to make accurate predictions [39]. Meanwhile, the second approach has been implemented by the Online prediction with experts method. In this method, the focus is on optimizing the model rather than controlling the input data. This optimization is done iteratively for each data instance. The model considers the information and advice provided by a set of models. This technique will allow, on the one hand, to adapt the model to the dataset and, on the other hand, to harden the model since the final model is an optimal combination of several models, as shown in Table 3 [10,16].

Table 3. Summary table of analysis of mitigation techniques

Defensive techniques	Methodology	Advantages	Disadvantages
Reject On Negative Impact	Data cleansing	Efficiency in removing malicious inputs	Unsatisfactory for performance evaluation
Adversarial training	Data cleansing	Adaptation of the algorithm to data input	The high cost of optimal adversarial sample's generation Performance lowering
Null method	Data cleansing	Simplicity of implementation	The high cost of optimal adversarial sample's generation
Online prediction with experts	Enhancing the model's Robustness	Adaptation of the algorithm to data input	Hard to implement

5 Proposition of Security assessment pattern

Recently there is a growing interest in improving the security of machine learning models, and manifold mitigation techniques were developed [40]. However, we believe that to develop defensive mechanisms, it is necessary before to be able to evaluate the security of the targeted model. However, until today, there is no process to evaluate the security of systems based on machine learning models. For this reason, we have developed a process that allows analyzing the model in order to qualify its strengths and weaknesses from a security point of view. For this, we have opted for a pattern based on two approaches: proactive and curative. The proactive approach essentially aims to protect against possible adversary attacks before they occur, while the main objective of the curative approach is to ensure recovery from an adversary attack against the targeted system.

5.1 Proactive approach

We believe that to immunize a targeted system from adversary attacks, it is necessary to start by first of all the identification and prioritization of risks, as shown in Figure 5. This can be done by assessing the vulnerabilities that can generate potential threats, which in turn can be exploited by an adversary seeking to carry out a malicious operation. This step consists of identifying existing vulnerabilities in the model and prioritizing them according to their degree of severity. During this step, it is necessary to list the system functionalities and proceed to risk analysis and identification by choosing one of the methods used in risk management such as the Ishikawa method [41] employed in identifying the causes and effects of possible vulnerabilities and then proceed to a risk assessment by establishing the risk matrix as illustrated in Table 4. The latter classifies risks according to their probability of occurrence.

Table 4. Matrix of the probability of occurrence of risks

Probability of occurrence	Rating	Risk 1	Risk 2	…	Risk N
Almost certain	6				
Very likely	5				
Likely	4				
Possible	3				
Unlikely	2				
Rare	1				

Following the vulnerability assessment, an impact estimation is required. We have proposed a model that can be used to evaluate the impact of the vulnerability discovered in the system. Indeed, if the weakness does not allow the attacker to perform any malicious action, we consider the impact as level 0 (None) as illustrated in Figure 4.:

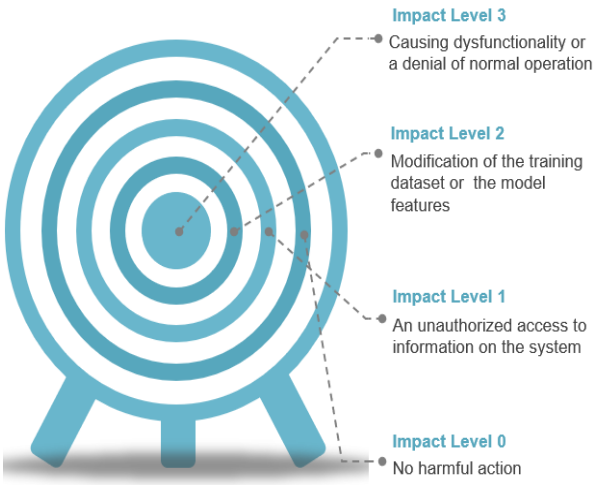

Fig. 4. Impact Evaluation levels

However, if the vulnerability causes unauthorized access of the adversary to information on the system, we assign it as level 1 impact, which is medium. While, if the vulnerability gives the adversary the possibility to modify the training dataset or the model features, the impact is considered as high (level 2). Otherwise, if the vulnerability causes a dysfunctionality or a denial of normal operation, the impact is classified as level 3, which is very high.

Fig. 5. Pattern of the Proactive Approach

After estimating the impact, we considered that studying the associated risks can help determine the next actions to be taken. Indeed, two cases may arise. Either the assessed risk is low and can be accepted as residual risk, or the risk can be corrected. In this case, a risk mitigation strategy must be developed and then applied. The security assessment process ends with a step of testing and reviewing the points examined from which further corrections can be implemented.

5.2 Curative approach

The curative approach is based on sensing performance degradation and reacting to a security incident. The steps of this approach are illustrated in Figure 6.

The first step of the curative approach consists of identifying and evaluating the impact of the attack. This step can be carried out using the impact prioritization according to the model provided in the Proactive Approach paragraph and illustrated in Figure 4. Then, an analysis of the corrupted assets should be performed to determine the attacker's goals, its knowledge of the system, and its capabilities. This information can be useful to infer the strategy adopted in the attack and, therefore, can help to select the appropriate actions to mitigate the damage of the attack. Another essential step is to conduct a profound analysis to determine the vulnerabilities exploited at the target system to develop the required corrective actions, thus ensure the availability and integrity of the target system. Following the implementation of the corrective actions, the testing stage is critical to ensure that the corrupted assets are repaired, that the vulnerabilities are removed, and that the attacker's strategy is no longer reliable against the target system.

Fig. 6. The pattern of the Curative Approach

The measures deployed must resolve the security incident while ensuring the continuity of service offered by the machine learning built system. Finally, the test results must be analyzed for further iteration of the curative process until optimal system protection is achieved while ensuring the continuous improvement of system security.

6 Implementation of security Assessment Pattern

Agents: Defender, Attacker.
Model: Machine learning model with vulnerabilities.
Input: Model, features, input data.

6.1 Proactive approach:

Identifying and prioritizing risks

- Call a method for risk identification $f : V \rightarrow R$, where V is the distribution of vulnerabilities and R the distribution of Risks.
- Compute the probability of occurrence $P_{occurrence}$ for different risks $r \in R$;
- Classify risks depending on $P_{occurrence}(r)$ into six categories: *{Almost certain, Highly likely, Likely, Possible, Unlikely, Rare}*

Estimating security impact

- Estimate Impact using the 4-levels Classification $I(r) = \{L_1, L_2, L_3, L_4\}$

Studying associated risks

- Evaluate the associated risks using the two-dimension vector $\boldsymbol{(P_{occurrence}(r), I(r))}$

Elaborating and applying risk mitigation strategy

- Determine $S_{mitigation}$, which can be either *{Accept the risk r, mitigate the risk r}*

Testing and controlling

- Test whether all risks are being addressed. If not, restart the process to cover all the risks.

6.2 Curative approach:

Identify and evaluate the impact of the attack.

- Estimate the impact of the attack using the 4-levels Classification $I = \{L_1, L_2, L_3, L_4\}$

Analyze the corrupted assets

- Examine the attack and identify the attacker's strategy.

Determine and evaluate the exploited vulnerabilities

- Determine V, where V is the distribution of vulnerabilities

Develop corrective actions

- Determine $S_{mitigation}$, which is a set of actions A that mitigate the vulnerabilities defined in the previous step $\{A(v), v \in V\}$

Testing and controlling

- Test whether all vulnerabilities are being addressed. If not, restart the process to cover all the vulnerabilities exploited by the attacker.

7 Discussion

The approaches we have provided in this paper take advantage of the flexibility of the model to improve the security of machine learning built systems. These approaches offer considerable adaptability since they can be applied for different attacks and are pertinent against different strategies used by the attacker. They have several advantages. Indeed, they allow the integration of security in the lifecycle of the machine learning model, particularly the proactive approach that ensures a continuous improvement of the model's security. Our pattern also contributes to making a judicious choice of the appropriate defensive mechanism for the employed model.

8 Conclusion

Today, there are several ways to perform attacks against machine learning models. Exploiting the vulnerabilities that reside in applications on machine learning systems can cause harmful and even fatal results in some cases, such as smart transportation and automated banking services. To face these challenges, the security of machine learning is an essential field that can potentially guarantee the development of artificial intelligence. Nevertheless, most of the mitigation techniques that have been developed are specific for particular models. Regarding this context, certain studies focus on defining techniques that can be generalized to multiple models by exploiting the transferability and shared vulnerabilities between models. However, this approach remains theoretical and, therefore, difficult to implement, given the diversity of the model's features and the constraints of technical limitations and high implementation costs.

References

1. G. Shobha and S. Rangaswamy, in *Handb. Stat.* (Elsevier, 2018), pp. 197–228
2. N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, ArXiv160202697 Cs (2017)
3. I. J. Goodfellow, J. Shlens, and C. Szegedy, ArXiv14126572 Cs Stat (2015)
4. M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, Mach. Learn. **81**, 121 (2010)
5. S. Qiu, Q. Liu, S. Zhou, and C. Wu, Appl. Sci. **9**, 909 (2019)
6. N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, Comput. Sci. Rev. **34**, 100199 (2019)
7. L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, 15 (n.d.)
8. Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, IEEE Access **6**, 12103 (2018)
9. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, ArXiv170806131 Cs **7908**, 387 (2013)
10. L. Muñoz-González and E. C. Lupu, in *AI Cybersecurity*, edited by L. F. Sikos (Springer International Publishing, Cham, 2019), pp. 47–79
11. D. Lowd and C. Meek, in *Proceeding Elev. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD 05* (ACM Press, Chicago, Illinois, USA, 2005), p. 641
12. M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, ArXiv180400308 Cs (2018)
13. S. Alfeld, X. Zhu, and P. Barford, 7 (n.d.)
14. B. I. P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S. Lau, S. Rao, N. Taft, and J. D. Tygar, in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf. - IMC 09* (ACM Press, Chicago, Illinois, USA, 2009), p. 1
15. B. Nelson, M. Barreno, F. J. Chi, B. I. P. Rubinstein, U. Saini, C. Sutton, A. D. Joseph, J. D. Tygar, and K. Xia, 10 (n.d.)
16. S. Shen, S. Tople, and P. Saxena, in *Proc. 32nd Annu. Conf. Comput. Secur. Appl.* (ACM, Los Angeles California USA, 2016), pp. 508–519
17. S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, ArXiv170202284 Cs Stat (2017)
18. A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, ArXiv190502175 Cs Stat (2019)
19. A. Kurakin, I. Goodfellow, and S. Bengio, ArXiv160702533 Cs Stat (2017)
20. Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, 17 (n.d.)
21. T. Gu, B. Dolan-Gavitt, and S. Garg, ArXiv170806733 Cs (2019)
22. Z. Abaid, M. A. Kaafar, and S. Jha, in *2017 IEEE 16th Int. Symp. Netw. Comput. Appl. NCA* (IEEE, Cambridge, MA, 2017), pp. 1–10
23. G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, ArXiv180611146 Cs Stat (2018)
24. P. Neekhara, S. Hussain, S. Dubnov, and F. Koushanfar, 11 (n.d.)
25. Y. Long, V. Bindschaedler, and C. A. Gunter, ArXiv171209136 Cs (2017)
26. N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, ArXiv161103814 Cs (2016)
27. R. Shokri, M. Stronati, C. Song, and V. Shmatikov, ArXiv161005820 Cs Stat (2017)
28. S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, ArXiv161008401 Cs Stat (2017)
29. M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, 17 (n.d.)
30. B. Nelson, 245 (n.d.)
31. H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, Int. J. Autom. Comput. **17**, 151 (2020)
32. O. Ibitoye, R. Abou-Khamis, A. Matrawy, and M. O. Shafiq, ArXiv191102621 Cs (2019)
33. G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, in *2008 IEEE Symp. Secur. Priv. Sp 2008* (IEEE, Oakland, CA, USA, 2008), pp. 81–95

34. B. Biggio, G. Fumera, and F. Roli, Int. J. Mach. Learn. Cybern. **1**, 27 (2010)
35. B. Biggio, G. Fumera, and F. Roli, in *2011 IEEE Int. Conf. Syst. Man Cybern.* (IEEE, Anchorage, AK, USA, 2011), pp. 977–982
36. A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, ArXiv181000069 Cs Stat (2018)
37. N. Akhtar and A. Mian, IEEE Access **6**, 14410 (2018)
38. H. Hosseini, Y. Chen, S. Kannan, B. Zhang, and R. Poovendran, ArXiv170304318 Cs (2017)
39. A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, ArXiv190412843 Cs Stat (2019)
40. B. Biggio and F. Roli, Pattern Recognit. **84**, 317 (2018)
41. K. Ishikawa, *Guide to Quality Control*, 13. print (Asian Productivity Organization, Tokyo, 1996)