Enhancing Machine Learning Security: Identifying Risks and Assessing Effectiveness

Renuka Bhagavati, Dept. of Computer Science, Research Scholar, SunRise University, Alwar(Rajasthan)
Dr. Pawan Kumar Pareek, Assistant Professor (Dept. of Computer Science), SunRise University, Alwar (Rajasthan)

ABSTRACT

Due to recent technology advancements, machine learning is now widely applied in a variety of contexts. It has shown remarkable success in solving a wide range of complicated issues, and its talents are strikingly similar to, if not superior to, those of humans. Recent research, however, has shown that machine learning models can be attacked in a number of ways, putting both the models and the systems they are used in at risk. Furthermore, the opaque character of deep learning models makes such attacks difficult to detect. In this survey, we take a comprehensive look at the security concerns surrounding machine learning, investigating the nature of the threats, the strategies for mitigating them, and how to evaluate their efficacy. This study addresses all facets of machine learning security, from the training phase through the testing phase, rather than just one or the other. At first, we introduce the adversarial machine learning model and examine the potential points of attack.

Keywords: Machine Learning Models, Adversarial Machine Learning, Security Assessments

INTRODUCTION

Recent years have seen significant advancements in machine learning techniques, leading to their widespread implementation in a variety of sectors, including image classification, autonomous vehicles, natural language processing, speech recognition, and smart healthcare, to name a few. Machine learning has already surpassed human performance in some areas, such as image classification. Spam filtering and dangerous programme detection are two examples of where machine learning has been utilised to improve security and open the door to exciting new possibilities. New research, however, reveals a plethora of security risks inherent in machine learning models themselves: The first is the use of poisoned training data, which can lower accuracy or be used for other error-generic/error-specific attack purposes; the second is the use of a well-designed backdoor in the training data, which can have disastrous effects on the system; the third is the use of a carefully-crafted disturbance in the test input (adversarial examples), which can cause the model to malfunction; and the fourth is the use of a model stealing attack, model inversion attack, or membership In safety- and security-critical applications like autonomous driving, smart security, smart healthcare, etc., machine learning systems are particularly vulnerable to the aforementioned security concerns.

In recent years, there has been a lot of focus on the topic of machine learning security. Since Szegedy et al. brought attention to the danger posed by adversarial examples in deep learning systems, a great deal of work has been done to ensure their safety. The idea of using machine learning for security purposes is not new; in fact, it can be dated back to 2004's Dalvi et al. These earlier publications, for example investigated so-called adversarial machine learning on non-deep machine learning algorithms for use in detecting spam, PDF malware, intrusions, and so on . Early attacks can be categorised as either evasion attacks or poisoning attacks, with the former being more common.

II. MACHINE LEARNING MODEL IN THE PRESENCE OF ADVERSARIES

What is Machine Learning?

In Fig. 1, we see the big picture of a machine learning setup. The following are some of the ways in which we define machine learning systems.

Stages: In most cases, we can divide a machine learning system into two distinct phases: The first step, known as training, involves the use of data for model formation and parameter estimation; the second, known as testing, involves the application of the trained model to a specific goal, such as classification, to provide a predicted label for the input data.

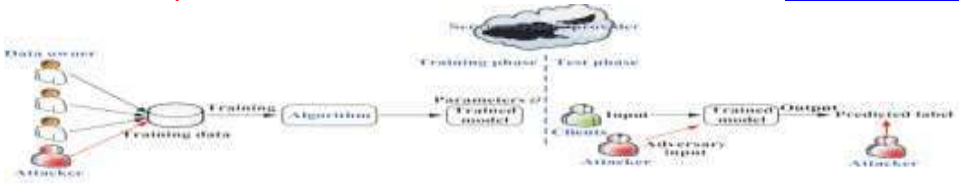

Fig. 1. Overview of Machine learning Systems, which illustrates the two Phases, the learning algorithm, and different Entities.

Learning Algorithm: An Algorithm takes in data from a training set and outputs a parameterized model. We classify machine learning algorithms as either neural network (NN) algorithms or other types of algorithms. Here, we use the term NN algorithms to refer to the many different types of Neural Network (NN) algorithms that have made significant advances in recent years and greatly boosted the efficiency of machine learning systems. In contrast, we refer to the other common machine learning algorithms, like the Support Vector Machine (SVM), k-means, Naive Bayes, etc., as non-NN methods.

Models with Adversarial Entities: As can be seen in Fig. 1, adversarial models of machine learning add attackers to the standard set of data owners, system/service providers, and customers. Large amounts of sensitive training data are typically kept secret and belong to the data owners. The provider of a system or service is the entity responsible for creating the algorithm, training the model, and carrying out the action or providing the service in question. Users who access the service, for instance through the service's prediction APIs, are clients, while an attacker can be either an external opponent or a nosy user within the system.

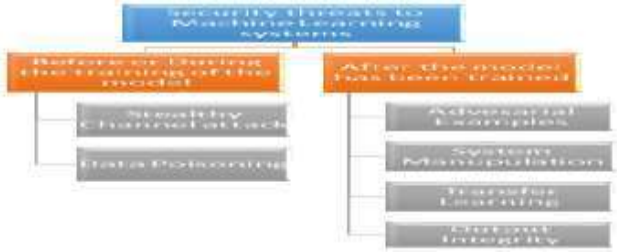

Fig. 2: Attacks on Machine Learning Systems

III. ATTACKS ON MACHINE LEARNING

The dangers and attacks that ML systems have to deal with are discussed here. To far, there are five broad classes into which all security risks encountered by machine learning systems can be classified (see Fig. 2). Recovery of sensitive training data (including model inversion attack and member-ship inference attack); recovery of adversarial examples; recovery of stolen models; and training set poisoning. Two assaults happen during the practise phase, and three more occur during the examination. In the following sections, we will examine these five attacks in turn and provide commentary on them.

. Poisoning of Training Materials

Poisoning attack refers to the deliberate alteration of a model's training data in order to falsely influence the model's prediction. Research shows that even a little amount of intentionally poisoned training data can significantly impact the accuracy of a machine learning model. Fig. 1.3 provides a summary of poisoning incidents. In this research, we categorise poisoning works according to whether or not the NN model is the intended victim.

1. Pollution Attacks on Non-NN Models

Designed for Use in Security and Anomaly Detection : Applications Many security detection applications, including anomaly detection and malware detection, have made extensive use of machine learning. Toxin assaults using these tar- gets are obvious winners. Principal Component Analysis (PCA)-subspace approach based anomaly detection system in backbone networks is proposed to be vulnerable to three poisoning attacks by Rubinstein et al. It is demonstrated that the detector's performance degrades drastically when even a little amount of poisoned data is introduced. Although this approach is straightforward and highly effective, it is specific to binary classification problems and hence cannot be used with any other learning technique. In order to create their poisoning attack, which they call a chronic

poisoning attack, Li et al. employ the Edge Pattern Detection (EPD) method on IDSs that rely on machine learning. The procedure can taint several learning algorithms , such as SVM, LR, and NB. However, the approach in involves slow poisoning over a lengthy period of time and is difficult to practise.

III. ATTACKS ON MACHINE LEARNING

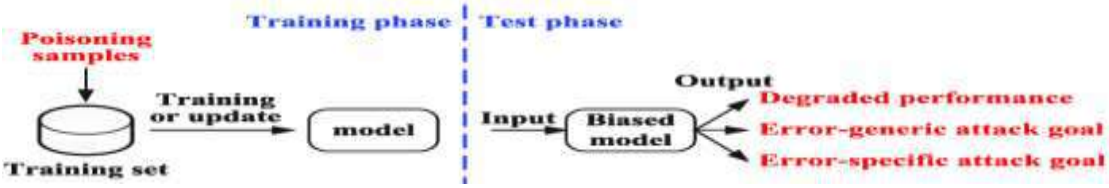

Fig. 3: Overview of Poisoning Attacks

Attacking Biometric Identifiers: Adaptive biometric recognition systems use machine learning methods to account for people' changing biometric characteristics as a result of factors like ageing. However, an attacker can use the updating procedure to bypass security measures . A poisoning attack on a principal component analysis (PCA) face recognition system is proposed by Biggio et al. The adaptive updating mechanism can be exploited to compromise the system template by submitting a series of well crafted phoney faces (i.e., poisoned samples) and posing as the victim. At long last, the assailant has his own face to use in his impersonation of the victim. It is assumed that users only store a single template in the system and that the attacker has full knowledge of the system, including the feature extraction algorithm, the matching algorithm, the template update algorithm, and the victim's template. The approach described above is further developed by Biggio et al. ,who apply it to a more realistic face recognition system in which the system retains numerous templates per user and uses different matching methods, and where the attacker only has an estimate of the victim's face image. It is shown, though, that the success rate of an attack varies from one attacker-victim pair to the next.

Support Vector Machines (SVM) are the focus of poisoning attacks proposed by Biggio et al. , which include injecting designed training data into the SVM classifier in order to raise its test error rates. To create the poisoned data, they employ a gradient ascent technique predicated on the SVM's best answer. This approach uses an optimisation formulation and can be kernelized, but it requires complete familiarity of the algorithm and the training data in order to produce poisoned data.

Clustering techniques have been widely employed in data analysis and security applications , including market segmentation, online page classification, and virus detection. However, a skilled adversary can compromise the clustering procedure itself. The clustering process can be tainted by an attacker, as shown by Biggio et al. , who only need to introduce a few poisoned samples into the training set. It is also possible to conceal these poisoning samples inside the input data. Clustering of malware samples and clustering of handwritten digits are used to assess the effectiveness of this method. Biggio et al. suggest a similar poisoning strategy that aims to eliminate clusters of behavioural malware by augmenting training data with carefully constructed poisoning samples that exhibit poisoning behaviours. These approaches typically work by first determining how far apart two groups are, and then introducing poisoned data to muddy the waters between them. Since this is the case, clustering algorithms will make the mistake of combining three distinct clusters into . These approaches are general and can be used to undermine a wide variety of clustering methods. These techniques, however, call for the attacker to have intimate knowledge of the target clustering algorithm, training data, feature space, etc.

B. Backdoor in the Training Set

Fig. Backdoors Attack in Cybersecurity

New studies reveal that an adversary can plant a backdoor in the pre-trained model or the training data. Figure 4 provides an abstract of backdoor attacks. While the backdoor has no effect on the model's regular operation, it does cause the model to incorrectly assign the target label to the backdoor instance upon the occurrence of a predetermined trigger condition. Due to the opaque character of deep learning models, such a backdoor attack can easily go undetected.

BACKDOOR ATTACKS

BadNet is a maliciously trained network proposed by Gu et al. When a particular input is received, BadNet may lead the model to behave badly. They show how efficient BadNet is by using it to classify handwritten digits and traffic signs. Backdoors in learning systems are the subject of research by Ji et al. Third-party primitive learning modules (PLMs) are to blame for opening the security holes. Once a predetermined trigger condition is met, the malicious PLMs that are part of the machine learning system might cause the system to malfunction. They show how to attack a system used to detect skin cancer without the attacker needing any prior knowledge of the system or the training procedure . In contrast, the attacker inserts backdoors by directly modifying the model's parameters. In practise, this assumption is hard to meet.

Using data poisoning, Chen et al. suggest a backdoor attack on deep learning models. In order to plant a backdoor, poison samples are added to the training dataset. Since their attack may be used with a weak attack model, it doesn't require specific information about the model or the data used in training . While almost 90% of attacks are successful, just 50 poison samples are injected .Backdoor attacks on CNN models are proposed by Liao et al. via the injection of covert perturbations. The attacker can establish a target label based on a specific embedded pattern.

IV. IMPLEMENTING SAFEGUARDS

1)Defenses Against Poisoning Attacks In Non-NN Models

Defences in Anomaly or Security Detection: Rubinstein et al. present a technique to defend against poisoning attacks on an anomaly detector, and they call it ANTIDOTE. By employing strong statistical methods, ANTIDOTE is able to reject the tainted samples and reduce the impact of statistical outliers. Biggio et al. view the prevention of poisoning assaults as an issue in the same vein as outliers detection: the occurrences are rare and their distribution is different from that of the training data. To counteract the effect of these anomalies (poisoning samples), researchers have turned to Bagging Classifiers, an ensemble approach. Specifically, they train multiple classifiers using distinct sets of data and then aggregate their predictions to mitigate the impact of outliers in the training set .They test a spam filter and an online IDS for poisoning attacks using the ensemble approach .

Defending SVM :Zhang and Zhu suggest a game-theory based defence for distributed SVM as a means of protecting against its misuse. They consider the attacker's and the learner's competing goals using game theory. Learner outcomes in hostile circumstances can be predicted using Nash equilibrium . Incorrect updates and performance losses due to poisoned data can be avoided with this strategy for distributed SVMs. A defence based on game theory, however, would have a high computing cost.

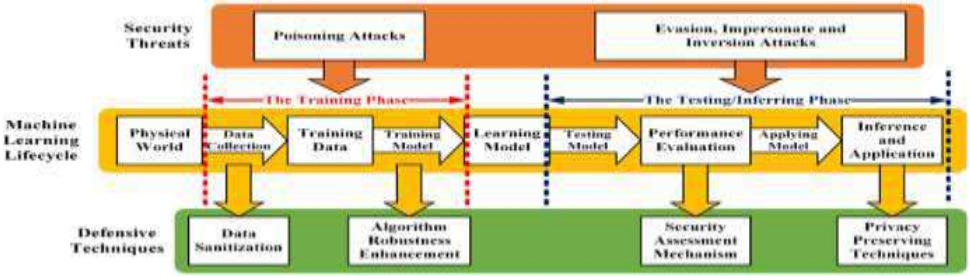

Fig. 5 Defensive Techniques of Machine Learning

2) Defenses Against Poisoning Attacks In NN Models: To protect NN from poisoning attacks, Yang et al. suggest estimating the model's loss as a defence mechanism. Suspicious input data is any that causes a greater loss than the threshold. Despite the simplicity and

generality of their approach , they simply provide a single, uninformative detection result without conducting a thorough analysis of the defence. A system called AUROR is proposed by Shen et al. to protect the collaborative deep learning systems. AUROR checks out suspicious individuals by finding anomalous features because poisoning data strongly influences the distribution of the features learned by the model. While AUROR's defence against poisoning attacks has no impact on the target model's performance, the effectiveness of this method's defences decreases as the number of hostile users increases.

c. Privacy-Preserving Machine Learning Methods to Prevent the Disclosure of Confidential Training Information

There are three main types of protections for ML models against the recovery of private training data: There are three main categories of methods for securing distributed learning: (1) methods based on cryptographic primitives like differential privacy and homomorphic encryption; (2) methods based on secure aggregation and ensembles of distributed learning like Federated Learning and Private Aggregation of Teacher Ensembles (PATE) ; and (3) methods based on trusted platforms and processors. Table 10 provides a summary of privacy-preserving machine learning methods that prevent the recovery of private training data.

1) Cryptographic Approaches Based on Primitive Techniques

A differential privacy based deep learning framework is developed by Abadi et al. To strike a better balance between privacy, efficiency, software complexity, and model quality, they also offer ways to enhance the efficacy of training based on differential privacy. The dif- ferential privacy-based strategy is applicable to a wide variety of ML methods, but it introduces noise into the gradient during model training and hence reduces the quality of the trained model. Jayaraman et al. show that existing privacy-preserving techniques require a trade-off between privacy and model performance. To put it another way, the model's performance will suffer if present privacy-preserving mechanisms are used .Phong et al. demonstrate that the privacy-preserving distributed learning system in may nonetheless divulge sensitive information to the server. So, they add homomorphic encryption and asynchronous stochastic gradient descent to NN to make a better method.

2) Distributed Learning Securely Aggregated/Assembled

To protect personal information, Shokri and Shmatikov present a distributed learning architecture in which numerous entities can learn a NN model together using only a portion of the parameters learnt by each entity individually. The basic principle is that deep learning algorithms based on stochastic gradient descent can be executed in parallel . Mohassel and Zhang present new protocols for logistic regression, linear regression, and neural network models that protect users' privacy. The protocol is implemented in a two-server model, where two servers use secure two-party computation to train their own models on the distributed private data. Bonawitz et al. present a Federated Learning framework that is based on secure Multi-Party Computing (secure aggregation). When using a distributed learning model, the gradient information from each user's model can be kept safe thanks to secure aggregation. Papernot et al. suggest a privacy-protecting training methodology called PATE (Private Aggregation of Teacher Ensembles). Several models, dubbed teacher models, were trained on separate sensitive datasets. Therefore, the student model cannot access the data or parameters of a specific teacher model , as it is learnt based on the output of a noisy aggregation of all the teachers. To accommodate massive workloads and imperfect, uncurated datasets, Papernot et al. expand PATE. In order to assemble the teacher models with minimal noise, they create new noisy aggregation procedures.

V. ASSESSING EFFECTIVENESS

Threat Modeling: Begin by identifying potential threats and vulnerabilities specific to your machine learning system. This could include attacks such as adversarial attacks, model inversion attacks, data poisoning, model stealing, and more. Understand the potential impact of these threats and the likelihood of their occurrence.

Security Controls and Countermeasures: Evaluate the security controls and countermeasures implemented within your machine learning system. This could involve techniques like input validation, robust model training, adversarial training, data sanitization,

privacy-preserving mechanisms, and access controls. Review the effectiveness of these controls in mitigating known threats.

Penetration Testing: Conduct penetration testing or red teaming exercises to simulate real-world attack scenarios. This involves hiring ethical hackers or security experts to identify vulnerabilities and attempt to exploit them. The goal is to assess the system's resilience to different attack vectors and identify areas for improvement.

Adversarial Testing: Specifically test the system's vulnerability to adversarial attacks, where malicious actors intentionally manipulate input data to deceive or mislead the machine learning model. This could involve crafting adversarial examples and assessing the model's response. Measure the model's robustness against different attack strengths and evaluate any defenses in place.

Monitoring and Anomaly Detection: Implement mechanisms for real-time monitoring and anomaly detection within your machine learning system. This can include techniques such as auditing model predictions, monitoring input and output data distributions, and detecting any deviations from expected behavior. Timely identification of anomalies can help prevent or mitigate potential security breaches.

Continuous Improvement: Security is an ongoing process, and it is important to continuously update and improve security measures as new threats emerge. Stay updated with the latest research and developments in machine learning security, participate in the security community, and learn from the experiences of others.

A. Design-for-Security

The designer of a machine learning system typically pays attention to the model selection and performance evaluation phases, but not the security phases. In light of the aforementioned security threats to ML systems, it is crucial to conduct thorough security evaluations of ML systems during the design phase and employ cutting-edge ML security methods. This approach, often known as the design-for-security paradigm, is an essential complement to the more common design-for-performance approach. For instance, Biggio et al. suggest a methodology for assessing the reliability of classifiers in terms of security. By raising the adversary's capability and adversary's knowledge, they replicate attacks at progressively higher levels. Similarly, Biggio et al. recommend gauging a classifier's robustness by measuring how much its performance suffers in the face of a variety of threats. In particular, they produce training and test sets, in addition to simulating assaults for use in penetration tests.

B. Judging on the Strength of one's Attacks

Carlini and Wagnerassess 10 modern detection methods and demonstrate how they can be defeated by employing robust attacks with modified loss functions. To this end, it is recommended that machine learning algorithms undergo a thorough security examination employing sophisticated attacks that take into account the two points below. White-box attacks are the gold standard since the attacker knows everything about the model, the data, and the defence mechanism and has significant control over the outcome of the evaluation. Second, instead of evaluating under minimally perturbed assaults solely, consider high-confidence attacks/maximum-confidence attacks . High-confidence attacks can circumvent the defences offered against minimally-perturbed attacks, as demonstrated by Carlini and Wagner Early efforts on hostile examples examine how deeply deep learning algorithms are affected by even small changes. However, the maximum-confidence adversarial attacks, which can reflect the security of an algorithm against more powerful attacks , are a more logical choice for analysing the safety of a deep learning algorithm.

C. Evaluation Metrics

To begin, it is recommended that more metrics be used to report the performance of the learning algorithm, including not only accuracy but also the confusion matrix (true positive, false positive, true negative, false negative), precision, recall, ROC (receiver operating characteristic) curve, and AUC (the area under the ROC curve), so that the full performance information can be reflected, and works can be compared with ease. Second, you can utilise the curves for judging security . In order to assess the safety of educational systems, Biggio

and Roli recommend utilising security evaluation curves. Comprehensive evaluation of the system's performance under attacks can be obtained using the security evaluation curves, and the curves are also useful for comparing the efficacy of various defensive strategies. These curves characterise the performance of the system under attacks of varying severity and by attackers with varying levels of knowledge.

VI. FUTURE DIRECTIONS

The field of machine learning security research is thriving. Recent years have seen a proliferation of literature on tit-for-tat attacks and defences.

1) Attacks subjected to Actual Physical Conditions: Many security flaws in machine learning models have been discovered, and most of them have been tested in digital simulations. Research is ongoing to determine how effective these attacks are in real-world physical settings and to develop solutions that take these conditions into account. Physical adversarial instances can trick traffic sign recognition systems, but they stand out visually and aren't very realistic. There have been a number of recent efforts devoted to creating physically robust adversarial examples. In addition, several organisations are now using DNN-based intelligent monitoring systems. Is it possible for humans to become invisible to object detectors using only adversarial examples? Unlike digital adversarial example assaults or road sign-oriented adversarial example attacks, this task is made more difficult by the huge intra-class variances of humans as well as their dynamic motions and various postures.

2)Methods of Machine Learning that Protect users' Anonymity: There has been a rise in concern over the privacy implications of machine learning in recent years. security of model parameters from the service provider's perspective and security of user privacy data from the user's perspective are both necessary for the successful deployment of deep learning. To this day, there is need for improvement in the effectiveness of machine learning strategies that rely on cryptographic primitives, as these strategies often add unnecessary complexity during model training, which can have a negative impact on the model's final performance. Inefficiencies and performance issues persist in distributed or integration-based training frameworks. Researching safe and effective machine learning models and frameworks is essential. One interesting method is the use of a collaborative design that combines hardware platform, software, and algorithm to safeguard DNN's privacy.

3) Intellectual Property (IP) Protection of DNN： Deep learning model training necessitates large amounts of training data and substantial computing power. It can take several weeks or months to complete the training procedure. This means that the producers of machine learning models have important commercial intellectual property that must be safeguarded. There are now only a few number of watermarking-based IP protected works for ML models .Improved ways of IP protection for DNN are still outstanding issues.

4) Lightweight or Remote Machine Learning Security Techniques: Platforms will increasingly employ machine learning in decentralised, remote, and Internet of Things settings. Many standard security measures cannot be implemented under these conditions due to the lack of resources. A potential area of study is the development of remote or lightweight machine learning security techniques that are both effective and dependable.

5) A Methodical Approach to Assessing the Safety of Machine learning Systems:

There is a lack of research into evaluating the safety of machine learning systems. To be more specific, there is currently no all-encompassing means of assessing the safety of models, as well as the confidentiality of their inputs and outputs, throughout training. In addition, there is no standard approach and set of criteria to measure the efficacy of existing attacks and defences. There has to be research into and the establishment of measures for evaluating the security, robustness, and privacy of machine learning systems.

CONCLUSION

Although applications built on machine learning are more common, these systems remain vulnerable to several risks at every stage. The safe implementation of machine learning is an area of continuous investigation. This paper provides a thorough overview of machine learning security, discussing the five most common forms of threats and the appropriate responses for each stage of a machine learning system's lifespan. The threats are genuine, and

new security threats are always emerging, as a generalisation. Research has shown, for instance, that adversarial instances are transferable, which means they work well across a variety of machine learning models. It is demonstrated that poisoning examples can generalise well to various learning methods as well. Black-box scenarios are vulnerable to attacks thanks to the transferability. Due to the opaque nature of machine learning models, further research is required to understand the underlying causes of these attacks (e.g., is the adversarial example a flaw or an intrinsic aspect of the model). With any luck, the information presented in this paper can serve as a comprehensive set of rules for developing safe, robust, and confidential machine learning systems.

REFERENCES

1. Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). Towards the Science of Security and Privacy in Machine Learning. arXiv preprint arXiv:1611.03814.
2. Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 39-57). IEEE.
3. Biggio, B., Nelson, B., & Laskov, P. (2014). Poisoning attacks against support vector machines. In Proceedings of the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (pp. 380-389). IEEE.
4. Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 1310-1321). ACM.
5. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In 25th USENIX Security Symposium (USENIX Security 16) (pp. 601-618).
6. Barreno, M., Nelson, B., Sears, R., Joseph, A. D., & Tygar, J. D. (2006). Can machine learning be secure? In Proceedings of the 2006 ACM Symposium on Information, computer and communications security (pp. 16-25). ACM.
7. Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., ... & Erhan, D. (2018). Technical Report on the CleverHans v2.1.0 Adversarial Examples Library.
8. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.
9. Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., & Tygar, J. D. (2011). Adversarial machine learning. In Proceedings of the 4th ACM workshop on Security and artificial intelligence (pp. 43-58). ACM.
10. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.
11. Barreno, M., Nelson, B., Joseph, A. D., & Tygar, J. D. (2010). The security of machine learning. Machine Learning, 81(2), 121-148.
12. Papernot, N., & McDaniel, P. (2018). Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint arXiv:1803.
13. Zhang, X., Zhu, X., Gong, Y., & Xu, J. (2019). Adversarial defense by stratified convolutional sparse coding. IEEE Transactions on Neural Networks and Learning Systems, 30(10), 3140-3152.
14. Barreno, M., Nelson, B., Joseph, A. D., & Tygar, J. D. (2008). The security of machine learning algorithms and their implementations: A survey. Journal of Machine Learning Research, 9(9), 1-50.
15. Wang, T., Zhang, J., Liu, L., Wang, G., & Wang, B. (2018). Characterizing privacy risks in deep learning: Theoretical analysis and empirical evaluation. Future Generation Computer Systems, 86, 1024-1034.
16. Song, Y., Kim, T., Nowozin, S., & Ermon, S. (2017). Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In Advances in Neural Information Processing Systems (pp. 1903-1913).