A Survey on Machine Learning Adversarial Attacks

Flávio Luis de Mello

***Abstract*—It is becoming notorious several types of adversaries based on their threat model leverage vulnerabilities to compromise a machine learning system. Therefore, it is important to provide robustness to machine learning algorithms and systems against these adversaries. However, there are only a few strong countermeasures, which can be used in all types of attack scenarios to design a robust artificial intelligence system. This paper is structured and comprehensive overview of the research on attacks to machine learning systems and it tries to call the attention from developers and software houses to the security issues concerning machine learning.**

***Index Terms*—Adversarial attack, machine learning, poisoning, privacy attack, trojoning, backdooring, evasion, reprogramming**

I. INTRODUCTION

WHILE Artificial Intelligence (AI) ethics tends to get the most public attention, there is an increasingly concern about the issue of adversarial attacks to Machine Learning (ML). Attacks to ML products is an emerging problem that has not been addressed by many companies. This survey is mainly based on Polyakov work [1] and it tries to provide a structured and comprehensive overview of the research on attacks to ML products. It has been grouped existing techniques into different categories according to the taxonomy recently published by NIST [2]. For each category, it is identified key assumptions, which are used by the techniques to describe the attack taxonomy.

Several papers and posts were reviewed with the aim of identifying those terms and themes, which are the most current among authors. There is much overlap between papers and posts, with authors citing the same sources for the topics and terms they discuss. The reader is encouraged to read some works that provided reasonable explanations and compilations reflecting common if not consensus views across a number of authors. Biggio and Roli [3] provides a historical study, correlating the evolution of ML with a broader focus on computer vision and cybersecurity tasks. Akhtar and Mian [4], focused on computer vision applications, the deepest address on attacks and defenses. Charkraborty et al. [5], Liu et al. [6], and Papernot et al. [7] are all concerned with cataloging attacks and defenses with an even broader focus independent of the specific area of application. Pitropakis et al. [8] also present a taxonomy of attacks, but less structured than NIST.

The ML key issues of an AI system include data, model, and processes for training, testing, and validation. Although AI also includes various knowledge-based approaches, known as Reasoning Systems, the statistical-driven approach of ML introduces particular security challenges in training and inference phases. The ML methodologies operate with the assumption that their environment is benign, but this assumption does not always hold. These security challenges include the potential for manipulation of training data, and exploitation of model sensitivities to adversely affect the performance of ML classification and regression.

Therefore, attacks can occur on two different moments: during training or inference. Attacks during training take place more often than it seems. Most of the production ML systems retrain their prediction models periodically with new data. For instance, social network continuously retrain user's behavior model, which means that each user may interfere in this system by modifying the behavior. Polyakov [1] organizes the attacks on ML models depending on the actual goal of an attacker (Espionage, Sabotage, Fraud) and the stages of machine learning pipeline (training and inference), or also can be called attacks on algorithm and attacks on a model respectively (see Table 1). They are Evasion, Poisoning, Trojaning, Backdooring, Reprogramming, and Privacy attacks. Today, evasion, poisoning and inference are the most widespread.

Table 1. Categories of attacks on ML products (Adapt from Polyakov [1]).

Stage	**Goal**		
	Espionage	*Sabotage*	*Fraud*
Training	Poisoning	Poisoning Trojaning Backdooring	Poisoning
Inference	Privacy Attack	Reprogramming Evasion	Evasion

II. EVASION

THE most common attack to ML system occurs during inference stage and is called evasion. It refers to designing an input, which seems normal for a human but is wrongly classified by ML models. A typical example is to change some pixels in a picture before uploading, so that image recognition

This work was supported in part by Machine Intelligence and Computer Models Laboratory of Federal University of Rio de Janeiro (IM^{2}C/Poli/UFRJ) under grant Poli 19.257 of Coppetec.

Flávio Luis de Mello (D.Sc.) is associate professor at the Electronics and Computer Department from Polytechnic School at Federal University of Rio de Janeiro, Brazil (email: fmello@poli.ufrj.br).

system computes a wrong classification. Figure 1 shows an adversarial example taken from Polyakov [1] can even fool humans.

Fig. 1. Adversarial example for humans (Polyakov [1]).

Szegedy et al. [9] provide a good mathematical formalization for the process of deceiving prediction models. Goodfellow et al. [10] followed Szegedy steps and produced interesting results as shown in Figure 2 illustration. An image was correctly classified as a panda, but when some noise is added to such image, the prediction model classifies the panda image as a gibbon with 99.3% of confidence. It is quite trivia to create imperceptible perturbation that completely fools Deep Neural Networks (DNN) as shown in Figure 2. Jo and Bengio [11] suggests that Convolutional Neural Networks (CNN) are vulnerable to adversarial input attack because they tend to learn superficial dataset regularity instead of generalizing well and learning high-level representation that would be less susceptible to noise.

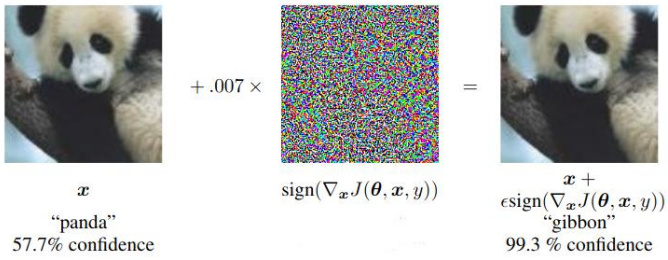

Fig. 2. Evasion attack against a deep neural network prediction model [10].

III. POISONING

IT seems that the first paper on poisoning attack against ML systems is Nelson et al. [12] who tries to fool a spam detector that guards email accounts so that you are able to get your spam emails into someone's inbox. Poisoning attacks are more prevalent in online learning models (models that learn as new data comes in), as opposed to those that learn offline from already collected data. In this type of attack, the attacker provides input samples that shift the decision boundary in his or her favor, that is, he or she attempt to poison your dataset to make your system misbehave.

According to Polyakov [1], there are four strategies for poisoning: (1) Label modification: Those attacks allows adversary to modify solely the labels in supervised learning datasets but for arbitrary data points. Typically subject to a constraint on total modification cost. (2) Data Injection: The adversary does not have any access to the training data as well as to the learning algorithm but has the ability to augment a new data to the training set. It is possible to corrupt the target model by inserting adversarial samples into the training dataset. (3) Data Modification: The adversary does not have access to the learning algorithm but has full access to the training dataset. This dataset can be poisoned directly by modifying the data before it is used for training the target model. (4) Logic Corruption: The adversary has the ability to meddle with the learning algorithm.

IV. TROJANING

POLYAKOV [1] highlights that in poisoning, attackers don't have access to the model and initial dataset, they only can add new data to the existing dataset or modify it. However, in Trojaning an attacker still do not have access to the initial dataset but have access to the model and its parameters and can retrain this model. This may happen in transfer learning. Most companies do not build their own models from scratch but retrain the existing models. For example, if it is necessary to create a model for workers detection at an industrial scenario, a software house may take the latest image recognition model of person and retrain it with dataset containing people dressing industrial coveralls. This means that most AI companies download popular models from the Internet where hackers can replace them with their own modified versions.

Liu et al. [13] describe the method for perform trojaning in ML systems. They inverse the neural network to generate a general trojan trigger, and then retrain the model with reversed engineered training data to inject malicious behaviors to the model. The malicious behaviors are only activated by inputs stamped with the trojan trigger. A trojan trigger is some special input that triggers the trojaned neural network to misbehave. Such input is usually just a small part of the entire input to the neural network (e.g., a logo or a small segment of audio). Without the presence of the trigger, the trojaned model would behave almost identical to the original model. The attacker starts by choosing a trigger mask, which is a subset of the input variables that are used to inject the trigger (see Figure 3a). Then, derive a set of data that can be used to retrain the model in a way that it performs normally when images of the persons in the original training set are provided and emits the masquerade output when the trojan trigger is present (Figure 3b). Specifically, it start with an image generated by averaging all the fact images from an irrelevant public dataset, from which the model generates a very low classification confidence (i.e., 0.1) for the target output. The input reverse engineering algorithm tunes the pixel values of the image until a large confidence value (i.e., 1.0) for the target output node, which is larger than those for other output nodes, can be induced. Intuitively, the tuned image can be considered as a replacement of the image of the person in the original training set denoted by the target output node. Moreover, repeat this process for each output node to acquire a complete training set. Finally, use the trigger and the reverse engineered images to retrain part of the model, namely, the layers in between the residence layer of the selected neurons and the output layer (Figure 3c). The essence of the retraining

is to establish the strong link between the selected neurons (that can be excited by the trigger) and the output node denoting the masquerade target, the weight between the selected neuron and the masquerade target node. It also reduces other weights in the neural network, especially those correlated to the masquerade target node to compensate the inflated weights.

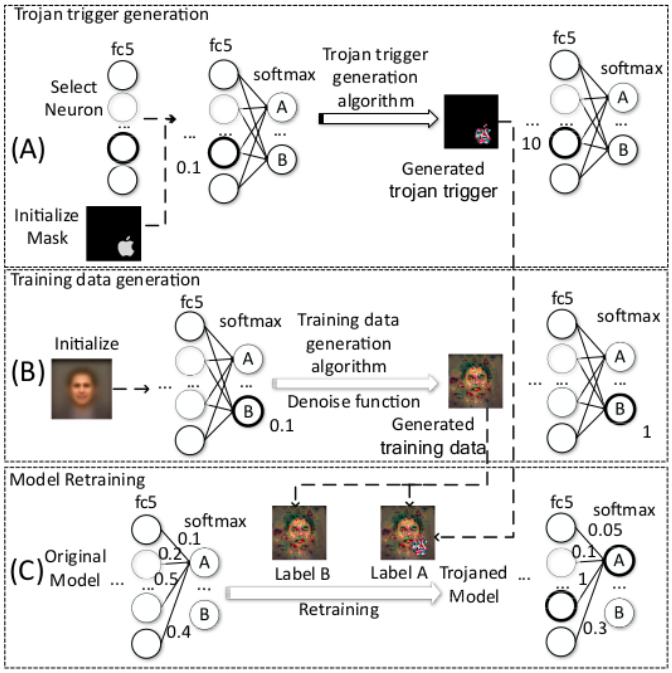

Fig. 3. Trojan attack overview [13].

V. REPROGRAMMING

USUALLY adversarial attacks are untargeted attacks that aim to compromise the performance of a model without necessarily requiring it to produce a specific output. This is quite different from targeted attacks in which the attacker designs an adversarial perturbation to produce a specific output for that input. For example, an attack against a classifier might target a specific desired output class for each input image, or an attack against a reinforcement learning agent might induce that agent to enter a specific state [14].

Elsayed et al. [15] consider a novel and more challenging adversarial goal: reprogramming the model to perform a task chosen by the attacker, without the attacker needing to compute the specific desired output. Consider a model trained to perform some original task: for inputs x it produces outputs $f(x)$. Then, consider an adversary who wishes to perform an adversarial task: for inputs y (not necessarily in the same domain as x) the adversary wishes to compute a function $g(y)$. The authors demonstrate adversarial programs that target several convolutional neural networks designed to classify ImageNet data. These adversarial programs alter the network function from ImageNet classification to: counting squares in an image, classifying MNIST digits, and classifying CIFAR-10 images.

Adversarial attacks allowed them to create images that resembled a specific noise containing several small white squares inside a big black square. They chose the pictures in the way that, for example, the network considered the noise with a white square on a black background to be a tench, and the noise with two white squares to be a goldfish, etc (see Figure 4). The image recognition system became a model that can calculate the number of squares in the picture. In a broader perspective, says Polyakov [1], attackers can use some open ML Application Programming Interface (API) for image recognition to solve other tasks that they need, and use the resources of the target ML model.

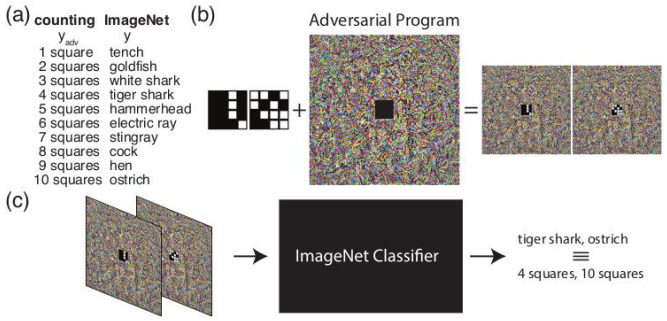

Fig. 4. Evasion attack against a deep neural network prediction model [15].

Figure 5, is also taken from Elsayed et al. [15] illustrate different adversarial programs targeted to repurpose networks pre-trained on ImageNet to count squares in images, to function as MNIST classifiers, and to function as CIFAR-10 classifiers.

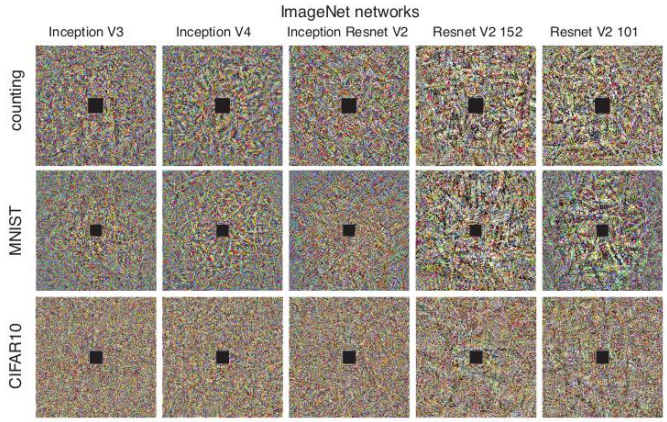

Fig. 5. Adversarial programs according to network architectures and desired tasks [15].

VI. PRIVACY ATTACK

PRIVACY attacks intend to explore the system, such as model, or dataset that can further be useful. In this survey, three types of privacy attacks are presented: membership inference [16], model inversion [17], model extraction [18].

The membership inference is one of the immediate attack against ML systems. It quantitatively investigate how ML models leak information about the individual data records on which they were trained. Given a data record and black-box access to a model, one wants to determine if the record was in the model's training dataset. To perform membership inference against a target model, we make adversarial use of machine learning and train our own inference model to

recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on [16]. Membership inference may be used as an exploratory phase for Evasion attacks.

The model inversion, also called input inference, is a common attack type. Unlike membership inference where someone wants to guess whether an example was in the training dataset, here someone wants to actually extract data from the training dataset. While dealing with images, it's possible to extract a certain image from the dataset, for instance, just knowing the name of a person, you can get his or her photo. In terms of privacy, this presents a big issue for any system, especially today when General Data Protection Regulation (GDPR) compliance is a hotspot.

Fredrikson et al. [17] describe model inversion attack against face recognition ML systems, where the attacker is given only the person's name and access to a facial recognition system that returns a class confidence score. These ML models are quickly becoming the standard by which facial recognition systems are evaluated, so the authors consider three types of neural network models: softmax regression, a multilayer perceptron network (MLP), and a stacked denoising autoencoder network (DAE). These models vary in complexity, with softmax regression being the simplest and the DAE being the most complex. Figure 6 show an image recovered using a the model inversion attack.

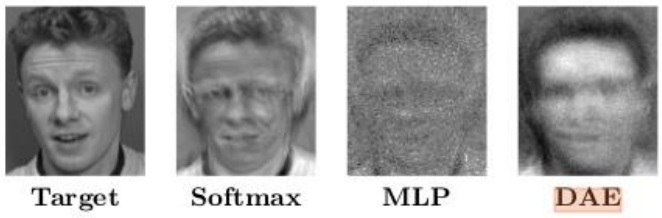

Fig. 6. Reconstruction of the individual on Target by Softmax, MLP, and DAE [17].

The model extraction, also called parameter inference, is the less common attack. The goal of this attack is to know the exact model or even a model's hyperparameters. This information can be useful for attacks like Evasion in the black-box environment. Figure 7 shows a data owner that has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $f^{\wedge} \approx f$ [18].

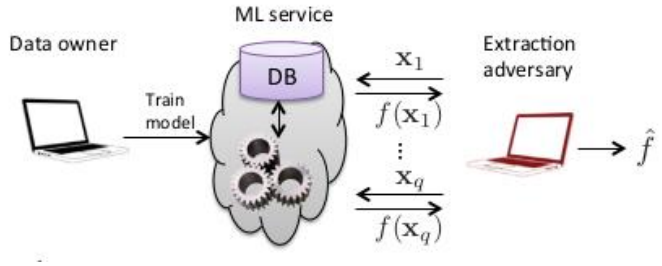

Fig. 7. Diagram of ML model extraction attacks [18].

VII. Attacks in the Physical World

Of course the attacks described so far in this paper are also being conducted in the physical world. Some attacks are ordinary use of techniques described so far in this article. Others are solutions to corrupt the image acquisition systems which will provide bad images to the ML systems.

The increasingly vast suite of surveillance tools available to state authorities has certainly given privacy advocates something to bristle at. In an exhibition, the artist Adam Harvey (see Figure 8) and fashion the designer Johanna Bloomfield demonstrated fashion's potential to thwart surveillance by state actors via accessories like a heat-cloaking anti-drone hoodie and scarf [19], and a series of blocky images that could become the building blocks of anti-surveillance makeup [20].

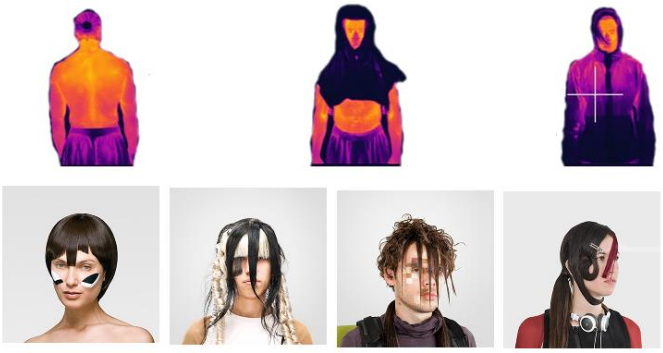

Fig. 8. Heat-cloaking wearbles (adapted from Dillow [19]) and anti-surveillance makeup [20].

Xu et al. [21] propose what they called an "adversarial" T-shirt, one with a printed image that evades person-detectors even when it is deformed by a wearer is changing pose. They claim it manages to achieve up to 74% and 57% success rates in digital and physical worlds, respectively, against the popular YOLOv2 model (see Figure 9).

Fig. 9. Evaluation of the effectiveness of adversarial T-shirts to evade person detection by YOLOv2 [21].

Thys et al. [22] have implemented a similar approach. They show how simple printed patterns can fool an AI system that was designed to recognize people in images (YOLOv2). If you print off one of the students' specially designed patches and hang it around your neck, from an AI's point of view, you may as well have slipped under an invisibility cloak.

Yamada et al. [23] has developed eyeglasses that help users protect their privacy by disabling facial-recognition systems in cameras. They prototype made two types of glasses, one using

near-infrared light and other using reflectors to fool the cameras into not seeing a face (see Figure 10).

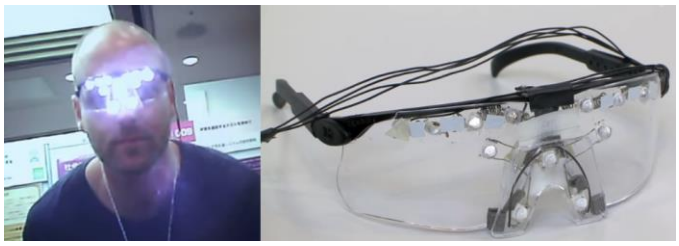

Fig. 10. Eyeglasses fooling facial-recognition systems in cameras.

Moreover, it seems not necessary to create expensive eyeglasses to compromise ML systems. Sharif et al. [24] have shown that specially designed spectacle frames can fool even state-of-the-art facial recognition software. Not only can the glasses make the wearer essentially disappear to such automated systems, it can even trick them into thinking you are someone else. By tweaking the patterns printed on the glasses, the authors were able to assume one another's identities or make the software think they were looking at celebrities (see Figure 11).

Fig. 11. Printed patterns on eyeglass-shaped cut-outs can compromise face recognition.

An expensive solution is the video of a black mirror-esque wearable face projector capable of tricking facial recognition systems, created by art and product designer Jing-Cai Lu [25]. The false faces being projected onto the individual can be seen shifting left and right, despite the wearer's head being still, indicating that the light could be coming from in front of them. Given that the Figure 12 snapshot is being filmed at night, it remains to be seen whether such an item would be usable during the daytime.

Fig. 12. Face projector.

Eykholt et al. [26] proposed a white-box adversarial sample generation method to attack their own trained road sign recognition models, including LISA-CNN models used LISA [27], a U.S. traffic sign dataset containing 47 different road signs, and GTSRB-CNN models, which trained on the German Traffic Sign Recognition Benchmark (GTSRB) [28]. They proposed two effective kinds of disturbance installation methods for road sign recognition scenarios, i.e., posters and stickers, as shown in Figure 13. They followed Sharif et al. [24] in constructing the loss function and took into account the printability and location limitations. Their assessment showed that they had achieved a 100% success rate in the poster installation driving test.

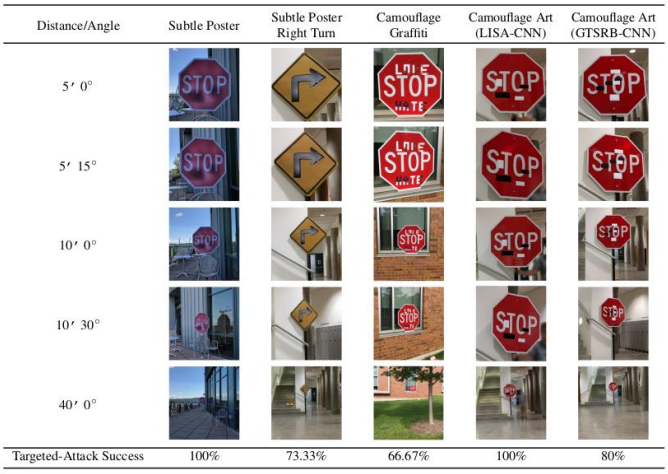

Fig. 13. Sample of physical adversarial examples against LISA-CNN and GTSRB-CNN.

VIII. Countermeasures for Adversarial Attacks

Almost all defenses described in literature are shown to be effective only for part of attacks. They tend to fail to defend from strong (fail to defend) and unseen attacks. It seems that the vulnerability of neural networks to adversarial samples originates from the existence of rarely explored subspaces in each feature map. This phenomenon is particularly caused by the limited access to the labeled data and/or inefficiency of regularization algorithms [29,30].

Metzen et al. [31] created a detector for adversarial examples as an auxiliary network of the original neural network. The detector is a small and straightforward neural network predicting on binary classification, that is, the probability of the input being adversarial. Grosse et al. [33] added an outlier class to the original deep learning model. The model detected the adversarial examples by classifying it as an outlier. They found that the two proposed metrics could distinguish the distribution of adversarial datasets and clean datasets. Feinman et al. [32] claimed that the uncertainty of adversarial examples is higher than the trustful data. Hence, they deployed a Bayesian neural network to estimate the uncertainty of input data and distinguish adversarial examples and clean input data based on uncertainty estimation. Hendrycks and Gimpel [34] showed that after whitening by Principal Component Analysis (PCA), adversarial examples have different coefficients in low-ranked components, and this feature is strong enough to provide a detection.

Adversarial samples may also be introduced into the

training dataset to improve the robustness of the target model by training model with the legalized adversarial samples. Szegedy et al. [9] firstly injected the adversarial samples and modified its labels to make the model more robust in the face of the adversaries. Goodfellow et al. [10] reduced significantly the misidentification rate on the MNIST dataset by using adversarial training. Huang et al. [35] increased the robustness of the model by punishing misclassified adversarial samples. Tramèr et al. [36] proposed ensemble adversarial training, which can increase the diversity of adversarial samples. However, the reader must be aware that it is unrealistic to introduce all unknown attack samples into the adversarial training, which leads to the limitation of an adversarial training such as the ones from this paragraph.

Since the transferability attribute holds even if the classifiers have a different architecture or are trained on the disjoint dataset, the key to preventing the black-box attack is to prevent the transferability of adversarial samples. Hosseini et al. [42] proposed a three-step NULL labeling method, in order to prevent the adversarial samples from one network to another network. Its main idea is adding a new NULL label to the dataset, and classify them to NULL label by training classifier to resist adversarial attacks. The advantage of this method is marking the perturbation input as an empty label rather than classifying it as the original label. At present, this method is one of the most effective defense methods against the adversarial attacks, which accurately resists the adversarial attacks, as well as does not affect the classification accuracy of the original data.

The regularization method aims to improve the generalization ability of the target model by adding regular terms, which are known as penalty terms to the cost function and make the model have good adaptability to resist attacks on an unknown dataset in prediction. Biggio et al. [38] used a regularization method to limit the vulnerability of data when training the SVM model. Lyu et al. [39], Zhao and Griffin [40], Rozsa et al. [41] used regularization method to improve the robustness of the algorithm and achieved good results in resisting adversarial attacks.

Feature squeezing [42] is a model enhancement technique, whose main idea is to reduce the complexity of the data representation, thereby reducing the adversarial interference due to low sensitivity. There are two heuristic methods, one is to reduce the color depth at the pixel level, that is, to encode the color with fewer values; the other is using a smooth filter on the image, that is, multiple inputs are mapped to a single value, thus making the model safer under noise and confrontational attack. Although this technique can effectively prevent adversarial attacks, it also reduces the accuracy of the classification of real samples.

Gu and Rigazio [43] introduced a kind of Deep Compression Network (DCN), which uses noise reduction automatic encoder to reduce the adversarial noise. Based on this phenomenon, DCN adopted a smoothing penalty similar to a convolutional autoencoder [89] in the training process, and was proved to have a certain defensive effect against attacks such as L-BGFS [9].

Samangouei et al. [44] proposed a mechanism applicable to both white-box and black-box attacks to reduce the efficiency of adversarial perturbation. This method utilizes the power of generative adversarial network [45], and the main idea is to "project" input images onto the range of the generator by minimizing the reconstruction error, prior to feeding the image to the classifier. Although defensive-GAN has been proved quite effective in defense against attacks, its success depends on GAN's expressiveness and generative ability, which is hard to achieve.

IX. Conclusion

MACHINE learning algorithms are vulnerable to adversarial attacks, and there are a large number of studies on adversarial attacks and defense methods. In this paper, there is a review the adversarial attacks carried out in the training stage and the inference stage of the target model, respectively. Although some defense methods have been proposed by researchers to deal with adversarial attacks and achieved good results, which can reduce the success rate of adversarial attack, they are generally aimed at a specific type of adversarial attacks, and there is no defense method to deal with multiple or even all types of attacks. Therefore, the key to ensuring the security of AI technology in various applications is to deeply research the adversarial attack technology and propose more efficient defense strategies.

References

[1] Polyakov, Alexander. "How to attack Machine Learning (Evasion, Poisoning, Inference, Trojans, Backdoors)", Towards Data Science, 2019. <https://towardsdatascience.com/how-to-attack-machine-learning-evasion-poisoning-inference-trojans-backdoors-a7cb5832595c> Access: January 12th, 2020

[2] NIST. "A Taxonomy and Terminology of 3 Adversarial Machine Learning", National Institute of Standards and Technology Interagency, Internal Report 8269, Eds: Elham Tabassi, Kevin J. Burns, Michael Hadjimichael, Andres D. Molina-Markham, Julian T. Sexton; October, 2019.

[3] Biggio, B.; Roli, F. "Wild patterns: Ten years after the rise of adversarial machine learning," Pattern Recognition, vol. 84, pp. 317-331, 2018. doi: 10.1016/j.patcog.2018.07.023

[4] Akhtar, N.; Mian, A. "Threat of adversarial attacks on deep learning in computervision: A survey", IEEE Access, vol. 6, pp. 14410-14430, 2018.

[5] Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; Mukhopadhyay, D. "Adversarial Attacks and Defences: A Survey", 2018.

[6] Liu, Q.; Li, P.; Zhao, W.; Cai, W.; Yu, S.; Leung, V. C. M. "A survey on security threats and defensive techniques of machine learning: A data driven view," IEEEaccess, vol. 6, pp. 12103-12117, 2018. doi: 10.1109/ACCESS.2018.2805680

[7] Papernot, N.; McDaniel, P.; Sinha, A.; Wellman, M. P. "SoK: Security and privacy in machine learning", In: 2018 IEEE European Symposium on Security and Privacy(EuroS&P), London, 2018. doi: 10.1109/EuroSP.2018.00035

[8] Pitropakis, Nikolaos; Panaousis, Emmanouil; Giannetsos, Thanassis; Anastasiadis,Eleftherios; Loukas, George. "A taxonomy and survey of attacks against machine learning", Computer Science Review, vol. 34, November 2019. doi: 10.1016/j.cosrev.2019.100199

[9] Szegedy, Christian; Zaremba, Wojciech; Sutskever, Ilya; Bruna, Joan; Erhan, Dumitru; Goodfellow, Ian; Fergus,Rob. "Intriguing properties of neural networks", arXiv, 2014.

[10] Goodfellow, Ian; Shlens, Jonathon; Szegedy, Christian. "Explaining and Harnessing Adversarial Examples", 2014. arXiv 1412.6572.

[11] Jo, Jason; Bengio, Yoshua. "Measuring the tendency of CNNs to Learn Surface Statistical Regularities", 2017. arXiv 1711.11561.

[12] Nelson, Blaine; Barreno, Marco; Chi, Fuching Jack; Joseph, Anthony D.; Rubinstein, Benjamin I. P.; Saini, Udam; Sutton, Charles; Tygar, J. D.; Xia, Kai. "Exploiting Machine Learning to Subvert Your Spam Filter", In: Proceedings of First USENIX Workshop on Large Scale Exploits and Emergent Threats, April 2008.

[13] Liu, Yingqi; Ma, Shiqing; Aafer, Yousra; Lee, Wen-Chuan; Zhai, Juan; Wang, Weihang; Zhang, Xiangyu. "Trojaning Attack on Neural Networks", In: Network and Distributed System Security Symposium, 2018. doi: 10.14722/ndss.2018.23300

[14] Lin, Yen-Chen; Hong, Zhang-Wei; Liao, Yuan-Hong; Shih, Meng-Li; Liu, Ming-Yu; Sun, Min. "Tactics of adversarial attack on deep reinforcement learning agents", 2017. arXiv:1703.06748.

[15] Elsayed, Gamaleldin F.; Goodfellow, Ian; Sohl-Dickstein, Jascha. "Adversarial Reprogramming of Neural Networks", 2018. arXiv:1806.11146

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

[17] Fredrikson, Matthew; Jha, Somesh K; Ristenpart, Thomas. "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures", In: CCS'15: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp.1322–1333, October 2015. doi: 10.1145/2810103.2813677

[18] Tramèr, Florian; Zhang, Fan; Juels, Ari; Reiter, Michael K.; Ristenpart, Thomas. "Stealing Machine Learning Models via Prediction APIs", 25th Security Symposium (USENIX), 2016.

[19] Dillow, Clay. "Anti-Surveillance Hoodie And Scarf Prevent Drones From Tracking You", In: Popular Science, Technoogy, 2013. <https://www.popsci.com/technology/article/2013-01/drone-proof-your-wardrobe-artist-unveils-surveillance-thwarting-hoodie-and-scarf/> Access on December 31th, 2019.

[20] Cvdazzle. "Camouflage from face detection", 2010. <https://cvdazzle.com/>, Access on December 31th, 2019.

[21] Xu, Kaidi; Zhang, Gaoyuan; Liu, Sijia; Fan, Quanfu; Sun, Mengshu; Chen, Hongge; Chen, Pin-Yu; Wang, Yanzhi; Lin, Xue. "Adversarial T-shirt! Evading Person Detectors in A Physical World", 2019. arXiv:1910.11099

[22] Thys, Simen; Van Ranst, Wiebe; Goedem, Toon. "Fooling automated surveillance cameras:adversarial patches to attack person detection", IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019. arXvi: 1904.08653

[23] Yamada, Takayuki; Gohshi, Seiichi; Echizen, Isao. "Privacy Visor: wearable device for privacy protection based on differences in sensory perception between humans and devices", IWSEC - International Workshop on Security, 2012.

[24] Sharif, Mahmood; Bhagavatula, Sruti; Bauer, Lujo; Reiter, Michael K. "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition", ACM Conference on Computer and Communications Security, 2016.

[25] Thalen, Mikael. "Is this wearable face projector being used by Hong Kong protesters?", The Daily Dot, 2019. <https://www.dailydot.com/debug/wearable-face-projector-hong-kong-protesters/>, Accessed on Jan 2nd, 2020.

[26] Eykholt, Kevin; Evtimov, Ivan; Fernandes, Earlence; Li, Bo; Rahmati, Amir; Xiao, Chaowei; Prakash, Atul; Kohno, Tadayoshi; Song, Dawn. "Robust Physical-World Attacks on Deep Learning Visual Classification", IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. doi: 10.1109/CVPR.2018.00175

[27] Mogelmose, A.; Trivedi, M.; Moeslund, T. "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey", Trans. Intell. Transport. Syst. 2012, 3, 1484–1497. doi: 10.1109/TITS.2012.2209421

[28] Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition", Neural Networks. 2012, 32, 323–332. doi: 10.1016/j.neunet.2012.02.016

[29] Denil, Misha; Shakibi, Babak; Dinh, Laurent; "Predicting parameters in deep learning". In: Advances in Neural Information Processing Systems, pp. 2148–2156, 2013.

[30] Wang, Beilun; Gao, Ji; Qi, Yanjun. "A theoretical framework for robustness of (deep) classifiers under adversarial noise, 2016. arXiv:1612.00334

[31] Metzen, J. H.; Genewein, T.; Fischer, V.; Bischoff, B. "On detecting adversarial perturbations", Proceedings of 5th International Conference on Learning Representations (ICLR), 2017.

[32] Feinman, R.; Curtin, R. R.; Shintre, S.; Gardner, A. B.. "Detecting adversarial samples from artifacts", 2017. arXiv:1703.00410

[33] Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; McDaniel, P. "On the (statistical) detection of adversarial examples", 2017. arXiv:1702.06280

[34] Hendrycks, D.; Gimpel, K. "Early methods for detecting adversarial images," ICLR Workshop, 2017.

[35] Huang, R.; Xu, B.; Schuurmans, D.; Szepesvári, C. "Learning with a strong adversary", 2015. arXiv:1511.03034

[36] Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. "Ensemble adversarial training: Attacks and defenses, 2017. arXiv:1705.07204

[37] Hosseini, H.; Chen, Y.; Kannan, S.; Zhang, B.; Poovendran, R. "Blocking transferability of adversarial examples in black-box learning systems", 2017. arXiv:1703.04318

[38] Biggio, B.; Nelson, B.; Laskov, P. !Support vector machines under adversarial label noise", In: Proceedings of the Asian Conference on Machine Learning, Taoyuan, Taiwan, 13–15 November 2011; pp. 97–112.

[39] Lyu, C.; Huang, K.; Liang, H.N. 'A unified gradient regularization family for adversarial examples", In: Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM), Atlantic City, pp. 301-309, 2015.

[40] Zhao, Q.; Griffin, L.D. "Suppressing the unusual: Towards robust cnns using symmetric activation functions", 2016. arXiv:1603.05145

[41] Rozsa, A.; Gunther, M.; Boult, T.E. "Towards robust deep neural networks with BANG", 2016. arXiv:1612.00138.

[42] Xu, W.; Evans, D.; Qi, Y. "Feature squeezing: Detecting adversarial examples in deep neural networks", 2017. arXiv:1704.0115

[43] Gu, S.; Rigazio, L. "Towards deep neural network architectures robust to adversarial examples", 2014. arXiv:1412.5068

[44] Samangouei, P.; Kabkab, M.; Chellappa, R. "Defense-GAN: Protecting classifiers against adversarial attacks using generative models", 2018. arXiv:1805.06605

[45] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. "Generative adversarial nets", In: Advances in Neural Information Processing Systems, Proceedings of the Annual Conference on Neural Information Processing Systems, 2013.

Flávio Luis de Mello received his DSc. in Theory of Computation and Image Processing from the Federal University of Rio de Janeiro - UFRJ (2006), MSc. in Computer Graphics from the Federal University of Rio de Janeiro - UFRJ (2003), Undergraduate degree in Systems Engineering from the Military Institute of Engineering - IME (1998).

He developed command and control systems and implemented military messages interchange applications during twelve years as a Brazilian Army officer. He was responsible for developing software applications based on machine learning and knowledge reasoning from Mentor Group.

Dr Mello currently is Associate Professor at the Electronic and Computer Engineering Department (DEL) of Polytechnic School (Poli) at the Federal University of Rio de Janeiro (UFRJ). He is head of the Machine Intelligence and Computing Models Laboratory (IM^{2}C).