The Future Prospects of Adversarial Nets

Muhammad Sohaib Yousaf[1], Shazia Riaz[1], Saqib Ali[1,2(✉)], Shuhong Chen[2], and Guojun Wang[2(✉)]

[1] Department of Computer Science, University of Agriculture, Faisalabad 38000, Pakistan
{2019ag2623,2018ag4549,saqib}@uaf.edu.pk

[2] School of Computer Science, Guangzhou University, Guangzhou 510006, China
{saqibali,shuhongchen,csgjwang}@gzhu.edu.cn

Abstract. Machine learning has obtained remarkable achievement in longstanding tasks in various domains of artificial intelligence. However, machine learning certainly has some security threats, such as adversarial examples that hamper the machine learning models from correctly classifying the data. The adversarial examples are minor perturbations in the actual inputs to detract the model from its original task. Adversarial Attacks and their defenses are found in parallel when it comes to the literature of machine learning adversaries. In this paper, we have tried to inspect the adversarial attack types and their defenses by comprehensively classifying different techniques.

Keywords: Adversarial examples · Machine learning · Poisoning · Adversarial training · Adversarial defense mechanism

1 Introduction

The availability of large amount of data outsourced in the cloud has increased the popularity of machine learning models. The automation of machines has made them highly vulnerable to external attacks. These attacks can be of any type such as attacks on the privacy leakage of outsourced data [26], adversarial attacks, evasion or inference attack, etc. All these types of attacks can force the machine learning models to misclassify or misbehave to the environment where the model has to perform the classifying task [1]. One of such attacks is the adversarial attack in which the input is perturbed to fool the machine learning model. The model can be deceived to an extent that it will classify a horse to a motorbike. Perturbation in the input is the amalgamation of a minor value that does the trick for the adversary. This minor value is denoted by epsilon ϵ in the machine learning field and the minor change in the input is termed as a perturbation. The term perturbation is used in a negative sense as it is used to dupe the models and drive the machines to flop.

Unlike the previous concept, the new development in this field argues that a little change in the inputs is going to change the output up to a great extent [24].

G. Wang et al. (Eds.): SpaCCS 2020, LNCS 12383, pp. 15–26, 2021.
https://doi.org/10.1007/978-3-030-68884-4_2

Great research has been done in this field in the previous years, but the reality is that the expansion in the machine learning field is inversely proportional to the steadfastness and reliability of the models developed in this respect. The reason behind this issue is the defenses, developed against malicious perturbations did not hold good, rather these defenses showed incorrect and vague evaluations [7]. Several of the findings have been deduced so far. One of them is the realization of the adversarial inputs to get good hold at one model will also have an effective gesture on the other models to malign the outputs of the other models as well for which the adversarial input is not made [15].

The objective of this paper includes a systematic explanation of the adversarial attacks and their defenses in a comprehensive way. The paper discusses the techniques that are used to generate the adversarial attacks and their defenses by discussing the theoretical as well as their implementation in detail. The paper also includes the classification of various techniques that are employed to either generate the adversaries or in the creation of their defenses. Moreover, this work investigates the advantages and shortcomings of these proposed methods and also suggests future directions that can be augmented by the researchers.

The paper is organized as follows. Section 2 describes contemporary adversarial attacks and their classification. The adversarial defenses are discussed in Sect. 3. Finally, the Sect. 4 concludes the paper.

2 Contemporary Adversarial Attacks and Their Classification

The term adversarial example was coined to generate some noise in the actual input and deceive the machine learning model. Adversarial attacks can be categorized as either black-box or white box attacks. The white box attacks can be describe as the attacks where the adversary has to get complete knowledge of the internal structure of machine learning model. Hence the attack example is constructed on the basis of this information. While on the other hand the black box attacks are those in which the adversary has no knowledge about the internal structure rather the attack is generated in target model disguise. In black box attacks a local substitute model is trained by querying the target model. Detail of some famous adversarial attacks is given below.

2.1 Fast Gradient Sign Method (FGSM)

A contemporary approach to the renowned adversarial examples attacks was the Fast Gradient Sign Method (FGSM). Goodfellow et al. [9] proposed a framework on the linearization of the cost function. The method worked on two different types of models one of which was generative adversarial model G that worked on creating the adversarial examples while the other model i.e., the distributive model D worked to guess whether the input came from examples generation model G or clean data. The framework worked by learning the generator distribution p_g from the original data x. There is a noise variable $p_z(z)$ that generates the noise distribution p_g. The generative model G maps to the data space function G (z; θ_g). It is a differentiable function which is represented by a multilayer perceptron having parameters θ_g. The other multilayer perceptron D(x; θ_d) outputs the single scalar. D(x) worked on the probability that x input came from the data than p_g as shown in Fig. 1.

Train D for maximizing the probability for assigning the correct label for both training examples and samples from G. Simultaneously, it trained G for minimizing log(1-D(G(z))).

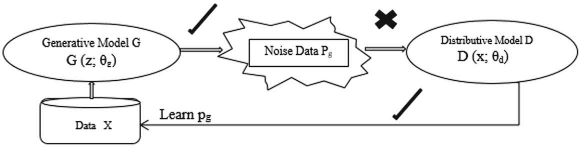


Fig. 1. Fast gradient sign method

J(θ, x, y) was the cost function for training, and J was used for the training of the model f around the training point x. Here in this approach the x corresponded to the actual input while there was an epsilon ϵ value added to the x input and it resulted in the formation of the x* value that was off-course an adversarial input example, the approach could be understood by the following equation.

$$\overrightarrow{x}^* \leftarrow x + \varepsilon . \bigtriangledown_{\overrightarrow{x}} J(f, \theta, \overrightarrow{x}) \tag{1}$$

The minute ϵ value was the parameter that controlled the magnitude of the perturbation. Like Szegedy et al. [24], Goodfellow et al. [9] focused on the more effectiveness of minute change in the actual input so that the attack remained un-detective for most of the defensive models. In this equation ϵ refers to the parameter which controlled the magnitude of the infiltration, which was decided to be included in the input.

2.2 Carlini and Wagner (C&W)

An approach was given by Carlini and Wagner [7] to overcome the defensive distillation developed against the adversarial perturbations. The methodology behind this adversarial attack was to form three types of attacks against the defense mechanisms i.e., L2, L0, and L∞ attacks as shown in Fig. 2. In the first step, the neural network was properly trained. After that, it computed the softmax and soft training labels by applying the network to each of them in the training data set. From these attacks, it could be deduced that the adversarial attack L2 used a minimum Delta value while the L0 type of attack was considered as non-differentiable and did not suit to the gradient descent. In the same way, the L∞ behaved almost the same in case of the gradient descent to the L2 attack but unlike the L2 attack, it was a differentiable attack. The C&W attack could be categorized in the white box configuration because the adversary should know the internal structure of the model before the example generation similarly should also know the parameters required to generate the adversarial examples.

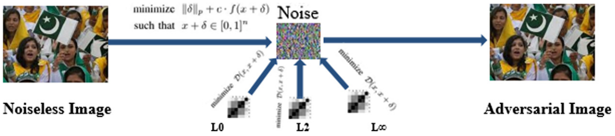
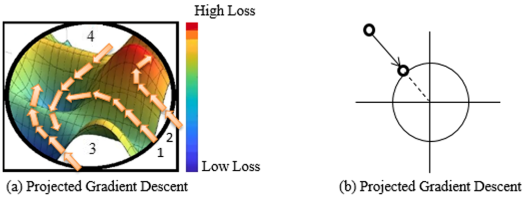

minimize $\|\delta\|_p + c \cdot f(x+\delta)$
such that $x+\delta \in [0,1]^n$
Noise
minimize $\mathcal{D}(x, x+\delta)$
minimize $\mathcal{D}(x, x+\delta)$
minimize $\mathcal{D}(x, x+\delta)$
Noiseless Image
Adversarial Image
L0
L2
L∞

Fig. 2. Carlini and Wagner attack

2.3 Projected Gradient Descent (PGD)

PGD was introduced to overcome the shortcomings of FGSM method by introducing the negative loss function. The idea was floated by Madry et al. [18]. The PGD method was a slightly sophisticated method as compared to the FGSM as the former was just a single step process to generate the adversaries while the latter is a multi-step process. The experiments were developed on two different data sets i.e., MNIST and CIFAR-10. The target value was to compute the L∞ gradient descent in the X + S space from where the initial values were randomly taken. The epsilon ϵ value was kept smaller than a certain value and repeats to find out the maximum loss of the machine learning model. As shown in Fig. 3(a) where four projections were shown that iterate randomly and start from a random position and pass through different gradients to get the gradient which incurred the maximum loss. Figure 3(b) show a single gradient projection that tries to target the desired value.



(a) Projected Gradient Descent

(b) Projected Gradient Descent

Fig. 3. The PGD method

2.4 Low Memory-BFGS

The LM-BFGS technique on adversarial examples was one of the pioneer works which was done by Szegedy et al. [24]. In this technique, the author named the small perturbations as adversarial examples. The basic philosophy behind this technique was the utilization of maximum stimulation on a random basis rather than on a natural basis in inspecting the properties of φ(x). It is shown in the figure below where Fig. 4(a) used the image projection on a random basis while in the Fig. 4(b) used natural projection to make the analysis. The author assumed that the random direction shows very similar semantically interpretable properties. However, the technique utilized $\mathrm{I}_c\ \varepsilon\ \mathrm{R}^m$ was a clean image. To compute the perturbation $\rho\ \varepsilon\ \mathrm{R}^m$, that was a slight change in the actual input.

$$\min_{\rho} \|\rho\|_2 \quad s.t. C(I_c + \rho = \ell; I_c + \rho\ \varepsilon\ [0,1]^m \tag{2}$$

Here ℓ denote the image label while C(.) was a deep neural network classifier. A critical limitation of the L-BFGS is its implementation with small datasets as it consisted of the limited memory. The L-BFGS/LM-BFGS method was also considered as the white box attack because in this method the adversary has some internal knowledge of the internal structure as well as its parameters.

(a) Maximum stimulation on random basis direction

(b) Maximum stimulation on the natural basis direction

Fig. 4. The basic philosophy of L-BFGS

2.5 Iterative Least Likely Class Method

Kurakin et al. [15] worked on this idea of getting the input from physical world where the perturbations could not be seen directly. The technique worked in the idea of Goodfellow et al. [9] in 2014 which was considered as fast adversaries' generated method. The methodology worked by adding the noise to the input iteratively until an adversarial example would be generated as depicted in Fig. 5. Hence there was a drawback to this approach that it only worked on small datasets like MNIST and CIFAR-10. The basic iterative method included a clip function to alter the pixel values but up to a limited extent.

$$X_0^{adv} = X, \quad X_{N+1}^{adv} = Clipx_{,e} X_N^{adv} - \alpha sign(\bigtriangledown_X J(X_N^{adv}, yLL)) \tag{3}$$

In each iteration a minute change was made to limit the step size small. The new method was called the iterative least-likely class method. In this method the previous approach was revised as it iteratively used to adjust the value of epsilon ϵ. As discussed earlier the BIM was a variation of the FGSM, so its black box implementation can be made.

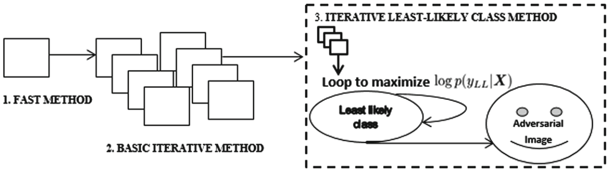

1. FAST METHOD
2. BASIC ITERATIVE METHOD
3. ITERATIVE LEAST-LIKELY CLASS METHOD
Loop to maximize $\log p(y_{LL}|\boldsymbol{X})$
Least likely class
Adversarial Image

Fig. 5. Adversarial examples in the physical world

3 Adversarial Defenses

To make the machine learning models robust in the true sense and resistant against adversarial attacks, researchers are taking a keen interest in designing models containing proper defense mechanisms to detect and reduce adversarial examples. But these defense mechanisms are in the developing stage and are not much robust against these attacks. Every attack is followed by a defense mechanism and in the same way, for every defense strategy, there is an attack following it. However, some certified robust techniques have also been proposed in the literature to fight against adversarial attacks.

3.1 Taxonomy of Adversarial Defense Mechanisms

Defense against adversarial attacks can be characterized in different ways. Researchers adopted several approaches to categorize them. We divide these into two broad categories, i.e., reactive defense and proactive defense.

State of the art adversarial defense mechanisms are classified into these main categories as shown in Table 1.

Table 1. Taxonomy of adversarial defense mechanism in machine learning

Adversarial defense strategies	
Reactive Defense	Proactive defense
Input preprocessing	Universal defense techniques
Data compression techniques	Defensive distillation
Dimensionality reduction	Adversarial training
Defense for data poisoning attacks	GAN based techniques
Gradient obfuscation techniques	Stochastic activation pruning
	Differential privacy as defense mechanism

Reactive Defense: In this approach, the attack designer works with the machine learning model designer to investigate the vulnerabilities of the model.

After the model is developed, the attacker scrutinizes its defense mechanism and formulates an adversarial response to alleviate this defense. Based on information gathered from different iterations of the above attack-defense process, the machine learning model designer augments the model with the required functionality to cope with these types of attacks as presented in Fig. 6(a).

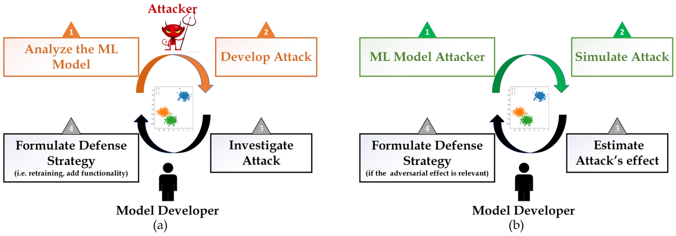


Fig. 6. Theoretical framework of reactive and proactive defenses [5]

Input Preprocessing: Preprocessing of input data is a popular defense mechanism. In [11], the researchers augmented the training data of the DNN model F with non-differentiable preprocessor G(.). Model F(G(.)) trained on transformed input does not undergo differentiable behavior in terms of x, making the adversarial examples failed to harm the DNN model predictions. Buckman et al. [6] introduced the concept of thermometer encoding. It is based on the discretization of input pixels to map them with multidimensional vector, the vector is used as a thermometer. The authors used a discretization preprocessor for this purpose and later on, the DNN model is trained on it.

Data Compression Techniques: Generally, most of machine learning model datasets consist of JPG images. Ghahramani et al. [14] investigates the effect of JPG compression on the performance of machine learning models in terms of their prediction accuracy. They found that in FGSM perturbations, compression does not much affect the classification accuracy of the model. Likewise, some researchers employed an ensemble-based method to study the consequences of compression on JPEG images as a countermeasure against FGSM attacks. Data compression defense techniques can be helpful to some extent as more compression effects the performance of the classifier in terms of prediction accuracies.

Dimensionality Reduction: Another method that is widely used as a defense against adversarial examples is dimensionality reduction. Among the techniques used for dimensionality reduction, "Principal Component Analysis" is the more famous one. For example, Bhagoji et al. [4] enhance the performance and flexibility of machine learning models by exercising the Principal Component Analysis and data 'anti-whitening' techniques to reduce the high dimensionality of input data these models.

Defense for Data Poisoning Attacks: Data poisoning attacks tries to change the statistical properties of training samples [23]. The attacker tries to extract a percentage of α of training samples of original dataset χ to create a new dataset χ_p

$$|\chi_p| = \alpha|\chi| \tag{4}$$

The technique used to defend these attacks is called the data sanitization technique, which separated the adversarial examples from the training set. The model designer trains the model using original samples as well as poisoned samples. The loss function calculated for the machine learning model decides whether attack or defense is successful.

Gradient Obfuscation Techniques: Gradient information of the model is mostly exploited to generate attacks. Shaham et al. [21] develop a framework by applying efficient optimization techniques to enhance the steadiness of neural network models. They tried to limit the loss function of adversarial example during parameter updating in the backpropagation process. The minimization-maximization method is used to implement the approach, which makes it difficult to generate new adversarial examples Gradient obfuscation techniques are still exposed to different attacks crafted in literature [3]. The problem with these techniques is that they cannot guarantee the removal of adversarial examples but, simply fool the adversary.

Proactive Defense: In proactive defense the developer of the machine learning model proposes the defense techniques in advance of the occurrence of attack, by inspecting the susceptibilities and loopholes of the model, from where the adversary can get access to damage the model's output predictions Fig. 6(b).

Defensive Distillation: Defensive distillation is proposed in [12] where the training method aims to decrease the size of the DNN model by transferring the knowledge of a lager DNN to a smaller one by the distillation process. Inspired by this technique, Papernot et al. [19] reformulates a defensive distillation technique that is robust against adversarial perturbation, i.e., Szegedy's L-BFGS attack, FGSM, or DeepFool. It trains the base model as well as the distilled model by using a similar DNN model architecture. They train the original model f on a given training set (x,y) with softmax layer temperature adjusted at T and calculate the probabilities produced by f(x). Then they trained another DNN model f^d on the training set (x, f(x)) sampled from f with the same softmax temperature T. This new model f^d is named as a distilled model. The working of defensive distillation is shown in Fig. 7. It is observed that, as compared to the original distillation model, the defensive distillation is more resistant to adversarial attacks by extracting knowledge from its own structures.

Adversarial Training: Adversarial training was the first strategy to guard against adversarial examples, devised by Goodfellow et al. [10]. In this technique, machine learning models as trained on a hybrid dataset consisting of original as well as of adversarial samples to enhance their robustness. The inclusion of adversarial examples with a true label (X', Y) in the dataset will instruct the model to classify X' as Y. In this way, the classifier will truly classify labels of

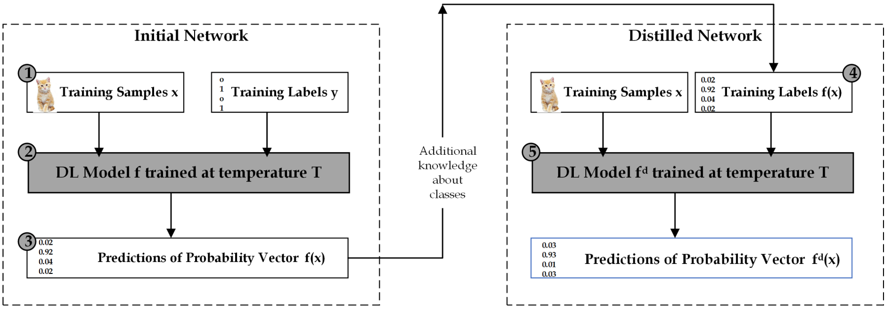

Initial Network
1
Training Samples x
0 1 0 1
Training Labels y
2
DL Model f trained at temperature T
3
0.02 0.92 0.04 0.02
Predictions of Probability Vector f(x)
Additional knowledge about classes
Distilled Network
Training Samples x
0.02 0.92 0.04 0.02
Training Labels f(x)
4
5
DL Model f^d trained at temperature T
0.03 0.93 0.01 0.03
Predictions of Probability Vector f^d(x)

Fig. 7. Defensive distillation mechanism

unseen adversarial examples. Adversarial examples X' for training dataset are produced by non-targeted FGSM as shown in the equation below.

$$X' = \epsilon sign(\bigtriangledown_X L(\theta, X, Y)) \tag{5}$$

To scale the adversarial training model to larger datasets, its training procedure is modified in scaled adversarial training by using batch normalization [13]. They demonstrated that batch normalization will help to enhance the performance of the training process of adversarial training techniques. Later on, ensemble adversarial training was proposed by Tramer et al. [25] which supplements the training process of adversarial training by using the perturbed training data transferred from other models.

GAN-based Techniques: The pioneers of Generative Adversarial Networks are Goodfellow et al. [9]. They introduced this approach to semantically enhance the performance of machine learning models. Later on, Lee et al. [17] presented GAN based defense model as a robust defense mechanism to mitigate adversarial attacks i.e., FGSM attacks. The proposed model accurately classifies both original images and adversarial images. The same GAN based approach is used in [22] to repair the contaminated images. In this approach, the generator component of GAN is used to restore the infected images.

Stochastic Activation Pruning: Dhillon et al. [8] proposed a defense technique, “Stochastic activation pruning”, in which nodes of each layer during forwarding propagation pass of the DNN model are dropped out stochastically. The hidden layers activation adaption effects the classification probabilities of the output, at the same time it enhances the robustness of the technique.

Universal Defense Techniques: Sometimes the universal defense approach was utilized to rectify the perturbed input. They employed pre input layer, perturbation rectifying network (PRN) in the model to deal with contaminated input. The training datasets for both PRN and targeted models are characterized by the same distribution. PRN is trained using the same parameters of base model [2]. The input samples are first passed through PRN before the input layer of the targeted model to identify the contaminations depending on the output of the rectifying unit.

Differential Privacy Defense Techniques: Recently, adversarial attacks are addressed by Differential Privacy (DP). Machine learning models are trained using large scale datasets containing sensitive user information. Thus, the privacy of this sensitive data is much more important against privacy attacks [20]. DP techniques are characterized by adding noise during the training of the model at a certain stage to maintain the privacy of data. Lecuyer et al. [16] proposed an innovative approach "Pixel DP" as certified robustness against adversarial attacks. PixelDP autoencoder can be appended at the beginning of nonmodifiable networks as a robust defense mechanism against any norm-based attacks. It adds a noise layer in the architecture of the machine learning model to generate the random outputs, which leverages the DP on prediction probabilities without altering the model accuracy on prediction results. PixelDP training is analogous to usual deep learning model training using similar loss and optimizer of the original model. It differs in the calculation of pre noise layers bounding it to the sensitivity of p-norm input deviations. The output Q(.) is given by the following equation.

$$Q(.) = h(g(.)) \tag{6}$$

Where g is prior to noise layer calculation and h denotes calculations in the next layers producing Q(.) as shown in Fig. 8.

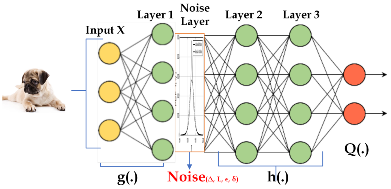


Fig. 8. Architecture of PixelDP approach

4 Conclusion

Our work describes the major adversarial attacks and defenses in a novel form as these techniques are not only classified but also represented graphically. Here we suggested, the generation of such models that not only keep the optimization but also tune the performance of models. The creation of more generalized models helps out to a great deal when it comes to taking care of the optimization problem. Orthodox models cannot help out in defending against various types of hybrid attacks. Similarly, the local smoothness and clean toy problems do not put a smart impact when it comes to the generation of adversarial examples and their defenses. It is better to align the adversarial examples with their defenses but try to develop such examples and defenses that should have their independent goals so that the defense against some developed adversarial example should be harder to find. Some other guidelines are there as well, on the basis of which effective models can be developed that exhibit a chronic impact on defenses. Adversarial examples become treacherous whose reverse engineering is not possible or hard to engineer. Similarly, those adversarial attacks prove to be lethal which are nontransferable from model to model.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant 61632009, in part by the Guangdong Provincial Natural Science Foundation under Grant 2017A030308006, in part by the High- Level Talents Program of Higher Education in Guangdong Province under Grant 2016ZJ01.

References

1. Ahmad, U., Song, H., Bilal, A., Alazab, M., Jolfaei, A.: Secure passive keyless entry and start system using machine learning. In: Wang, G., Chen, J., Yang, L.T. (eds.) SpaCCS 2018. LNCS, vol. 11342, pp. 304–313. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05345-1_26
2. Akhtar, N., Liu, J., Mian, A.: Defense against universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
3. Athalye, A., Carlini, N., Wagner, D.A.: Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. CoRR abs/1802.00420 (2018). http://arxiv.org/abs/1802.00420
4. Bhagoji, A.N., Cullina, D., Mittal, P.: Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. arXiv preprint arXiv:1704.02654, vol. 2 (2017)
5. Biggio, B., Roli, F.: Wild patterns: len years after the rise of adversarial machine learning. Patt. Recogn. **84**, 317–331 (2018)
6. Buckman, J., Roy, A., Raffel, C., Goodfellow, I.: Thermometer encoding: one hot way to resist adversarial examples. In: International Conference on Learning Representations (2018)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57 (2017)
8. Dhillon, G.S., et al.: Stochastic activation pruning for robust adversarial defense. arXiv e-prints arXiv:1803.01442, March 2018
9. Goodfellow, I., et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv e-prints arXiv:1412.6572, December 2014
11. Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations. arXiv e-prints arXiv:1711.00117 (Oct 2017)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv e-prints arXiv:1503.02531, March 2015
13. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv e-prints arXiv:1502.03167, February 2015
14. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of JPG compression on adversarial images. arXiv e-prints arXiv:1608.00853, August 2016

15. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv e-prints arXiv:1611.01236, November 2016
16. Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S.: Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 656–672 (2019)
17. Lee, H., Han, S., Lee, J.: Generative adversarial trainer: defense to adversarial perturbations with GAN. arXiv e-prints arXiv:1705.03387, May 2017
18. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv e-prints arXiv:1706.06083, June 2017
19. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597 (2016)
20. Quinlan, M., Zhao, J., Simpson, A.: Connected vehicles: a privacy analysis. In: Wang, G., Feng, J., Bhuiyan, M.Z.A., Lu, R. (eds.) SpaCCS 2019. LNCS, vol. 11637, pp. 35–44. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24900-7_3
21. Shaham, U., Yamada, Y., Negahban, S.: Understanding adversarial training: Increasing local stability of supervised models through robust optimization. Neurocomputing **307**, 195–204 (2018)
22. Shen, S., Jin, G., Gao, K., Zhang, Y.: APE-GAN: adversarial perturbation elimination with GAN. arXiv e-prints arXiv:1707.05474, July 2017
23. Steinhardt, J., Koh, P.W.W., Liang, P.S.: Certified defenses for data poisoning attacks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 3517–3529. Curran Associates, Inc. (2017)
24. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv e-prints arXiv:1312.6199, December 2013
25. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: attacks and defenses. arXiv e-prints arXiv:1705.07204, May 2017
26. Zhang, Q., Liu, Q., Wang, G.: A privacy-preserving hybrid cooperative searching scheme over outsourced cloud data. In: Wang, G., Ray, I., Alcaraz Calero, J.M., Thampi, S.M. (eds.) SpaCCS 2016. LNCS, vol. 10066, pp. 265–278. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49148-6_23