TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 11/13 pp894–905
DOI: 10.26599/TST.2020.9010050
Volume 26, Number 6, December 2021

Security Issues and Defensive Approaches in Deep Learning Frameworks

Hongsong Chen*, Yongpeng Zhang, Yongrui Cao, and Jing Xie

Abstract: Deep learning frameworks promote the development of artificial intelligence and demonstrate considerable potential in numerous applications. However, the security issues of deep learning frameworks are among the main risks preventing the wide application of it. Attacks on deep learning frameworks by malicious internal or external attackers would exert substantial effects on society and life. We start with a description of the framework of deep learning algorithms and a detailed analysis of attacks and vulnerabilities in them. We propose a highly comprehensive classification approach for security issues and defensive approaches in deep learning frameworks and connect different attacks to corresponding defensive approaches. Moreover, we analyze a case of the physical-world use of deep learning security issues. In addition, we discuss future directions and open issues in deep learning frameworks. We hope that our research will inspire future developments and draw attention from academic and industrial domains to the security of deep learning frameworks.

Key words: adversarial examples; deep learning frameworks; defensive approaches; security issues

1 Introduction

Given the successful application of deep learning in many fields[1–3], Artificial Intelligence (AI) has attracted increasing attention. Owing to the development of Graphics Processing Unit (GPU), deep learning algorithms and large-scale datasets can solve problems in various fields. Moreover, many practical applications and systems are driven by deep learning algorithms.

Companies, ranging from Information Technology (IT) firms to automobile makers (e.g., Google, Tesla, Baidu, Mercedes, and Uber), are testing driverless cars, which require deep learning techniques. In addition, major phone manufacturers offer facial authentication features for unlocking phones, and a number of behavior-based malware and anomaly detection solutions are based on deep learning[4, 5]. Although deep learning can bring certain conveniences, it is prone to numerous vulnerabilities. Recent research has found that deep learning is vulnerable to well-designed adversarial samples, which can easily fool a well-behaved deep learning model.

Szegedy et al.[6] first generated small perturbations in an image classification problem and deceived the most advanced Deep Neural Network (DNN) with high probability. As a result, samples misclassified by a DNN are called adversarial samples.

The generation of adversarial samples is based on understanding model structures and parameters to destroy deep learning model processes or make wrong predictions. This type of attack, including those based on obfuscated gradient[7] and root mean square gradient[8], is called the white-box attack. Meanwhile, the black-box attack is limited by knowledge on the model structure

• Hongsong Chen is with the Department of Computer Science, University of Science and Technology Beijing (USTB), Beijing 100083, China, and also with Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China. E-mail: chenhs@ustb.edu.cn.
• Yongpeng Zhang and Yongrui Cao are with the Department of Computer Science and Technology, University of Science and Technology Beijing (USTB), Beijing 100083, China. E-mail: zypmicro@outlook.com; 1445118489@qq.com.
• Jing Xie is with Defense Electronics Institute, China Industrial Control System Cyber Emergency Response Team, Beijing 100040, China.
* To whom correspondence should be addressed.
Manuscript received: 2020-09-06; accepted: 2020-10-09

and parameters.

Goodfellow et al.[9] claimed that the neural network is affected easily by small disturbances from inputs. The authors proposed the Fastest Gradient Sign Method (FGSM) to generate adversarial samples. Su et al.[10] proposed a black-box DNN attack that makes only differential perturbations to one pixel, which performs well at different image sizes.

Defense measures were proposed to defend against such attacks. For example, the gradient masking method was proposed by Goodfellow et al.[9]. He et al.[11] argued that a single-defense method performs poorly, and a defense system composed of multiple defense measures could better deal with adversarial examples.

A large number of deep learning-based applications are used in the physical world, especially in security-critical environments. Adversarial examples can be applied to the physical world. For instance, an adversary can construct physical adversarial examples and confuse autonomous vehicles by manipulating a traffic sign recognition system[11].

Other surveys on deep learning framework security issues were conducted. For example, Xu et al.[12] classified security issues from the perspective of black-box/white-box attacks, poisoning attacks, and escape attacks. Tariq et al.[13] divided attacks into four categories, namely, causative attacks, exploratory attacks, targeted attacks, and indiscriminate attacks. However, the above surveys lacked a comprehensive and systematic perspective on security and defense approaches in deep learning frameworks. Compared with other surveys, our classification is based on attack phase, adversarial knowledge, attack frequency, attack target, and attack scope. Bae et al.[14] discussed deep learning security and privacy issues. However, the authors mainly analyzed security issues using mathematical formulas and principles. Similarly, we also use figures to illustrate attack principles and mechanisms. At the same time, we list major software vulnerabilities in deep learning frameworks. Qiu et al.[15] discussed AI attack methods in the training and testing phases but did not connect them in the relationship between attacks and defense approaches. We made a one-to-one connection between attacks and corresponding defense technologies. In addition, we discussed future directions and open issues in deep learning frameworks. Thus, we conducted a highly comprehensive and methodical research on the security of deep learning frameworks and performed an in-depth analysis of related studies.

The structure of this paper is organized as follows: Section 2 introduces general deep learning models and processes, and Section 3 presents the deep learning principles as well as vulnerabilities and types of attacks caused by third-party libraries. Section 4 classifies attacks based on different viewpoints, and Section 5 details defense measures against various attacks. Section 6 describes a specific deep learning automatic driving application scenario to identify traffic signs and analyzes security problems. Finally, Section 7 concludes the study and identifies future research directions.

2 Deep Learning Framework Architecture

DNN processing is divided into two phases, that is, the training phase and the prediction phase. The training phase involves using existing data to learn the parameters in the network, and the inference phase employs the learned parameters to predict the unknown data[14]. The general DNN training process is shown in Fig. 1.

The general training process of a neural network involves obtaining parameters to minimize the cost function through known samples. The cost function measures the error between the predicted value of the model and the actual value of the sample. Completion of the DNN training phase requires forward and backward propagations. In the feed forward phase, the input propagates along the layer to calculate the output. Next, to minimize the error between the output and the actual label, a gradient descent algorithm is used. The prediction results are used in the inference phase, in which the model only propagates the input forward and treats the output as a prediction.

Convolutional Neural Networks (CNN) are widely used in the field of image recognition and classification. CNNs have four main operations, namely, convolution, nonlinear transformation, pooling or subsampling, and classification (fully connected layers). The CNN structure is illustrated in Fig. 2.

An example of a Recurrent Neural Network (RNN) is presented in Fig. 3. Unlike traditional forward feedback neural networks, RNNs introduce directional loops that can handle contextual correlations among inputs. The purpose of RNNs is to process time sequence data.

A Generating Adversarial Network (GAN) framework consists of a discriminator (D) and a generator (G). The G generates false data, and the D determines whether the generated data are true. GANs are actively researched in the field of image/speech synthesis and domain adaptation. Figure 4 displays the GAN structure.

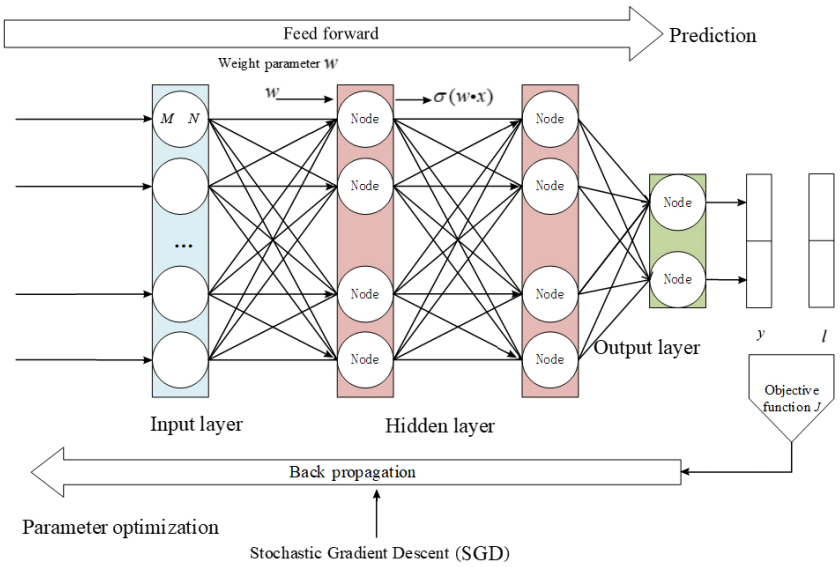

Fig. 1 General DNN training process. Here x represents the neural network input data, y represents labels for input data, w represents the weight of neural networks, M and N represent dimensions of weights, J represents the loss function, σ represents the activation function, and l represents prediction results of neural network.

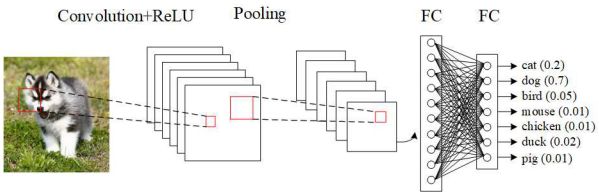

Fig. 2 CNN structure. Here FC stands for fully connected.

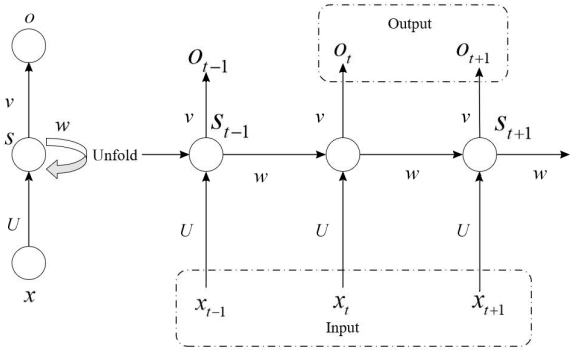

Fig. 3 RNN structure. Here o represents the output of RNN structure, U, and v are parameters of RNN, and O_t and S_t represent the output of the layer t. The difference between O_t and S_t is that O_t is output directly as a result, and S_t needs to be input to the next layer for calculation.

Table 1 shows the differences between neural network models.

3 Security Issue in Deep Learning Frameworks

Deep learning security problems can occur when an attacker either uses the DNN implementation principle to reversely generate adversarial samples or exploits the vulnerabilities of third-party libraries that are dependent on the underlying DNN.

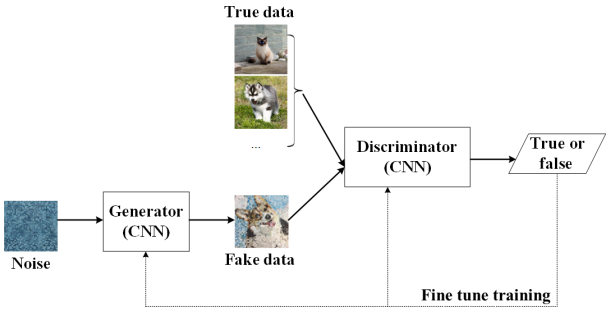

Fig. 4 GAN structure.

Table 1 Differences between neural network models.

Model	Advantage	Limitation
DNN	Simple architecture	Too many layers will lead to overfitting
CNN	Extract local features, such as image recognition	Cannot process time series data
RNN	Deal with time series features	Gradient disappearance
GAN	Generate new training data	Experience difficulties reaching Nash equilibrium

3.1 Adversarial example generation

Goodfellow et al.[9] proposed the FGSM algorithm implementation process, which analyzes the reasons for the existence of adversarial samples, and presented a method for generating the samples based on such analyses. The method involves adding a small

disturbance not easily perceived by the human eye to a picture that exerts maximal influence on the classifier through the action of the activation function. Figure 5 describes the addition of the disturbance and the influence process.

As shown in Fig. 5, the input sample is assumed to be x, and the adversarial sample is $\tilde{x}$, where θ and ω are the parameters of the deep learning algorithm model.

In a linear model, given that the feature of the sample input is limited, the classifier will not be able to distinguish between the sample x and the adversarial sample $\tilde{x}$ when the perturbation value η added to each element value in the sample is less than the input feature accuracy of the sample. Formally, for problems with well-separated classes, we expect the classifier to assign the same class to x and $\tilde{x}$ as long as $\|\eta\|_\infty < \varepsilon$, where ε is adequately small to be discarded by the sensor or data storage apparatus associated with the problem. Consider the product of the weight vector ω and the adversarial sample $\tilde{x}$: $\omega^{\mathrm{T}}\tilde{x} = \omega^{\mathrm{T}}x + \omega^{\mathrm{T}}\eta$. The adversarial perturbation increases the output of the neuron with $\omega^{\mathrm{T}}\eta$. If the dimension of the weight vector is n and the mean of the weight vector is m, then the maximum value is $\varepsilon \times n \times m$, and at this time, $\eta = \mathrm{sign}(\omega)$. Thus, in a high-dimensional space, even small disturbances can have a large impact on the output of the final neural network. Therefore, linear models can also produce adversarial samples.

In a nonlinear model, the linear perturbation is a process of a nonlinear differential equation. The parameters of the model are assumed to be θ, x is the input of the model, y is the target associated with x (the result of the classification), and $J(\theta, x, y)$ is the loss function. We can linearize the loss function near θ, thereby obtaining the best max-norm constrained perturbation of $\eta = \varepsilon \cdot \mathrm{sign}(\nabla_x J(\theta, x, y))$. Next, when we directly add the linear perturbation to the original sample, $\tilde{x} = x + \eta$, and the misclassification rate of the neural network is high. This method is the adversarial example generation process of the white-box attack FGSM algorithm. The calculation of η in the FGSM algorithm is shown in Fig. 6, where the green points represent the original sample and corresponding loss function values, and the red points represent the adversarial sample and corresponding loss function values.

The FGSM and DeepFool[16] are methods for generating adversarial samples, and both are white-box attacks. In a neural network, back propagation is used to minimize the loss function. An FGSM attack goes the opposite direction, adding the disturbance along the

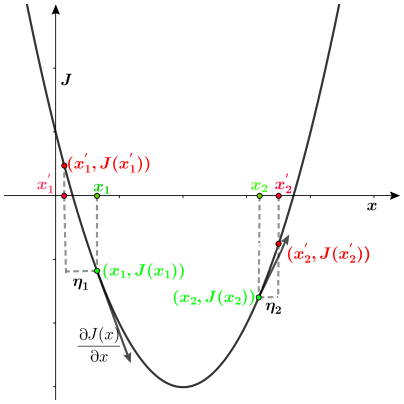

Fig. 6 Calculation of η in the FGSM algorithm.

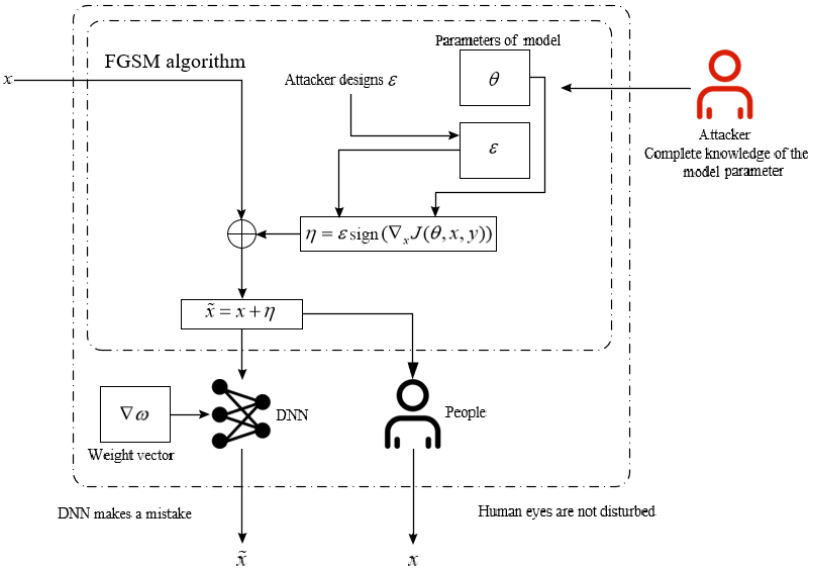

Fig. 5 FGSM algorithm process diagram.

gradient direction to generate the adversarial sample to maximize the loss function, which will fool the neural network model.

Figure 7 presents the results of different ε selections in the FGSM algorithm.

The FGSM can determine the direction of the disturbance addition but not the size of the disturbance. The size of the disturbance is generally artificially determined. Figure 7 shows that the disturbance direction is opposite the direction of x axis, and ε_1 and ε_2 are two disturbance sizes. The classification function can be misclassified by adversarial sample x_1 that is generated by disturbance ε_1, but the purpose of the misclassification cannot be achieved by adversarial sample x_2 that is generated by disturbance ε_2.

The DeepFool method improves the shortcomings of the FGSM, which can determine not only the direction of the disturbance addition but also its distance.

Figure 8 shows the use of the DeepFool algorithm to generate an adversarial sample in a linear binary classification.

In Fig. 8, $f(x) = \omega^{\mathrm{T}} x + b$ is a classifier. The formula is used for parameter optimization, where ω is the direction of the gradient for the decision function, and the previous scalar corresponds to the optimal perturbation coefficient ε. The optimal solution

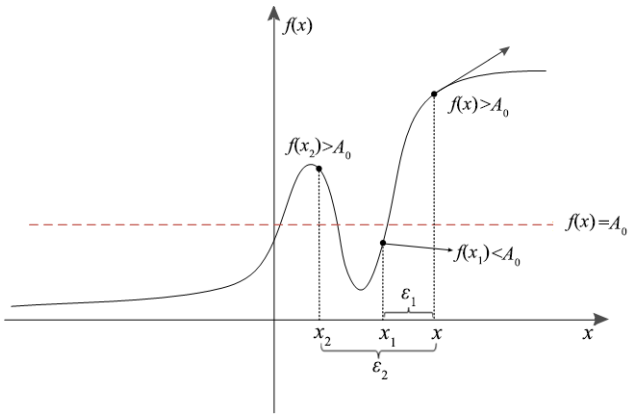

Fig. 7 Results of different ε selections in the FGSM algorithm.

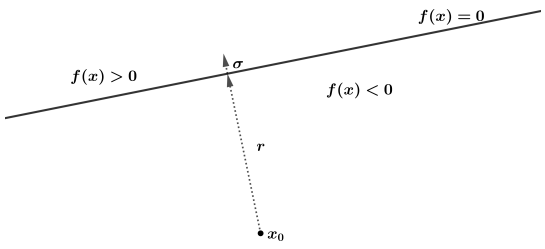

Fig. 8 Adversarial examples for a linear binary classifier[16].

r satisfies $f(x_0 + r) = 0$. Thus, the adversarial sample is $\tilde{x} = x_0 + r + \sigma$, where σ is a small deviation that gives $f(x_0 + r + \sigma) > 0$. The calculation formula of r is as follows:

$$r = \arg\min \|r\|_2 = -\frac{f(x_0)}{\|\omega\|_2^2}\omega.$$

3.2 Vulnerabilities of deep learning frameworks

Common Vulnerabilities and Exposures (CVE) in deep learning frameworks are shown in Table 2[17].

Deep learning faces adversarial sample attacks, and its frameworks have several security issues. The use of deep learning frameworks, such as TensorFlow, Caffe, and Torch, allows application developers to not pay attention to underlying implementation details, thereby substantially improving the development efficiency of AI applications. However, the efficiency of these deep learning frameworks is doomed by the complexity of the framework, and the more complex the system, the more likely the security risks. Specifically, these three frameworks are built on numerous third-party open-source basic libraries. After analyzing a large number of such libraries used by the three deep learning frameworks (i.e., TensorFlow, Caffe, and Torch), researchers found that they have many network security vulnerabilities that are prone to denial of service, escape, and system damage attacks[17]. Vulnerabilities, like memory access cross-border vulnerabilities, can be used by hackers to execute the three types of network attacks mentioned above and tamper with data streams to deceive AI applications.

Table 2 shows that security vulnerabilities from the general security framework used in a series of deep learning systems involve nearly all mainstream deep learning platforms.

4 Attack Classification in Deep Learning Frameworks

We classify deep learning attacks by attack type,

Table 2 CVE in deep learning frameworks.

Deep learning framework	CVE-ID	Type
Caffe	CVE-2017-9782	Heap overflow
Caffe/Torch	CVE-2017-12600	Denial of service
Caffe/Torch	CVE-2017-12604	Software crash
TensorFlow	CVE-2017-12852	Out of bounds
TensorFlow	CVE-2018-7577	Memcpy param overlap
TensorFlow	CVE-2018-10055	Heap buffer overflow
TensorFlow	CVE-2019-9635	Denial of service
TensorFlow	CVE-2020-5215	Denial of service

adversarial knowledge, attack phase, attack frequency, adversarial specificity, and attack method, as shown in Fig. 9.

According to its phase, attacks can be divided into poisoning and evasion attacks. Poisoning attacks add adversarial data to the training sample to influence the training of the classifier and obtain the wrong classifier. Evasion attacks use adversarial examples in the inference stage to make the classifier produce an error output. In terms of adversarial knowledge, attacks can be classified into white-box attacks, black-box attacks, and semi-white-box attacks. If an attacker fully masters the content of the deep learning system, such as the dataset and algorithm used, the structure of each layer of the network, and so on, then an attack based on this realization is called a white-box attack. An attack that knows only a part of this knowledge is called a semi-white-box attack. A completely ignorant attack is called a black-box attack.

Generally, white-box attacks cannot be implemented in real life. Black-box attacks can be classified as transfer-based, score-based, and decision-based attacks. Transfer-based attacks train a local model, then use the adversarial samples generated by the model to attack the target model. Score-based attacks obtain information inside a model by obtaining the classification confidence of the target model to the input. Decision-based attacks can only obtain the classification result of a model on the input. This type of attack is practical but the most difficult to complete. With regard to attack frequency, attacks can be classified as one-time and iterative attacks. One-time attacks only need one time to generate adversarial samples, whereas iterative attacks need several times to update the adversarial samples. Compared with iterative attacks, one-time attacks require less time, but the added disturbance is relatively larger. Moreover, compared with one-time attacks, iterative attacks can produce better results but require vast computing resources. According to attack target, attacks can be classified as targeted and non-targeted attacks. An attack is a targeted attack if the opponent's goal is to change the output of the classifier to a prospective target label. In the case of a non-targeted attack, the opponent's goal is for the classifier to select any incorrect label. Generally, non-targeted attacks demonstrate higher success rates than targeted attacks. Non-targeted attacks can be divided into only misclassification attacks and least likely attacks. Only misclassification attacks require a model to classify adversarial samples differently from the original class, whereas least likely attacks require a model to classify adversarial samples differently from the original class and with the least confidence. Regarding attack scope, they can be classified as individual and universal attacks. Individual attacks only need to modify a few features, whereas universal attacks need to modify every feature. Therefore, adversarial disturbances generated by individual attacks are more imperceptible than those generated by universal ones.

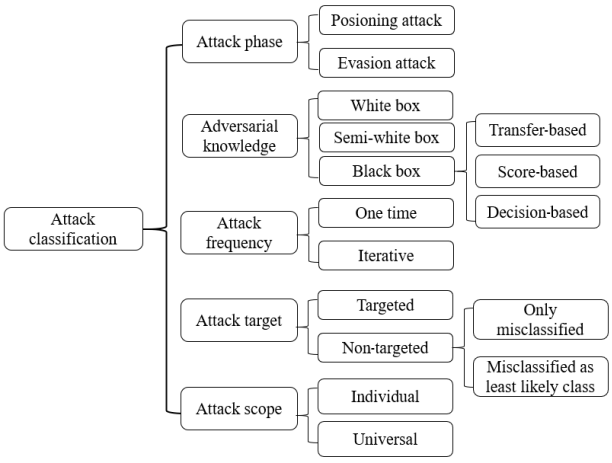

Fig. 9 Attack classification in deep learning frameworks.

The timeline of white- and black-box attacks is shown in Tables 3 and 4. In the two tables, we list some algorithms and research schedules for black- and white-box attacks in deep learning.

In a white-box attack, an attacker can destroy the learning process by injecting designed samples and adjust it with the gradient method. An attack approach and security model for a wireless sensor network and cloud computing are proposed[29, 30], which can serve as references for the deep learning security model. A white-box attack is easy to realize, as attackers have considerable knowledge. By contrast, implementing a black-box attack is difficult owing to model knowledge limitations.

5 Defensive Approach in Deep Learning Frameworks

Many defensive measures have been taken against deep learning security problems to better apply deep learning. The relationship between attacks and defensive approaches in deep learning is shown in Fig. 10.

In Fig. 10, we enumerate measures corresponding to common attack methods. Poisoning attacks are used to generate adversarial samples, and the defense

Table 3 Historical timeline of white-box attacks in deep learning frameworks.

Timeline	Year	Algorithm	Main contribution
Szegedy et al.[6]	2013	L-BFGS	First proposed the concept of adversarial sample and designed an optimized-based method to generate adversarial samples deliberately.
Goodfellow et al.[9]	2014	FGSM	Designed a method using the gradient of loss function, which can generate adversarial samples quickly.
Papernot et al.[18]	2016	JSMA	Designed a novel method that only needs to modify a few pixels.
Kurakin et al.[19]	2016	iFGSM	Designed the iterative FGSM, which can generate smaller disturbances than the FGSM, and showed that machine learning systems are vulnerable to adversarial examples in physical-world scenarios.
Huang et al.[20]	2017	Attacks on RL	Showed that adversarial attacks are also effective when targeting neural network policies in RL.
Athalye et al.[7]	2018	BPDA	Described the characteristic behaviors of defenses exhibiting effects, discovered three types of obfuscated gradients, and developed attack techniques to overcome them.
Xiao et al.[8]	2019	RMSG	Proposed an adversarial method generating perturbations based on root mean square gradient, which formulates the adversarial perturbation size in the root mean square level and updates gradient direction.
Zhang et al.[21]	2019	Boundary projection	Studied manifold optimization for the classification boundary of an adversarial attack and proposed the boundary projection method to generate adversarial examples that reduce the number of iterations for iterative attacks.

Table 4 Historical timeline of black-box attacks in deep learning frameworks.

Timeline	Year	Algorithm	Main contribution
Nelson et al.[22]	2012	Evading convex-inducing classifiers	First proposed existing black-box attacks that do not use a local model for convex-inducing two-class classifiers.
Ateniese et al.[23]	2013	Hacking smart machines	(1) Proposed that releasing trained classifiers is unsafe; (2) defined a model for a metaclassifier; (3) described several attacks against existing ML classifiers.
Narodytska et al.[24]	2016	Greedy local search	Proposed the Greedy Local Search algorithm to generate adversarial samples by perturbing randomly selected pixels with considerable influence on output probabilities.
Chen et al.[25]	2017	ZOO	(1) Showed that a zero-order oracle (without gradient information) can attack black-box DNNs; (2) proposed several techniques, including attack-space dimension reduction, hierarchical attacks, and importance sampling.
Ye et al.[26]	2018	Hessian-aware zeroth-order optimization	(1) Integrated Hessian information into gradient estimation while keeping the algorithmic form similar to the zeroth-order-based gradient descent method; (2) proposed several novel structured Hessian approximation methods; (3) proposed a descent-checking trick for black-box adversarial attacks.
Li et al.[27]	2019	Attack on cloud-based detectors	Designed four types of methods by incorporating semantic segmentation to achieve a high bypass rate with a very limited number of queries to fool cloud-based detectors.
Saxena[28]	2020	TextDecepter	Proposed a novel approach for formulating natural adversarial examples against Natural Language Processing (NLP) classifiers in the hard-label black-box setting.

measure for such attack is to eliminate outliers with large samples[31–34]. Evasion attacks can be prevented by enhancing the robustness of classifiers[7, 9, 35]. Meanwhile, encryption algorithms[36, 37] are used to guard against privacy leakage. Attacks against software vulnerabilities can be defended by writing high-quality codes and selecting highly secure third-party libraries.

6 Case Study—Deep Learning Security Scenario Research

We analyzed a deep learning software that identifies traffic signs to describe the types of attacks and threats

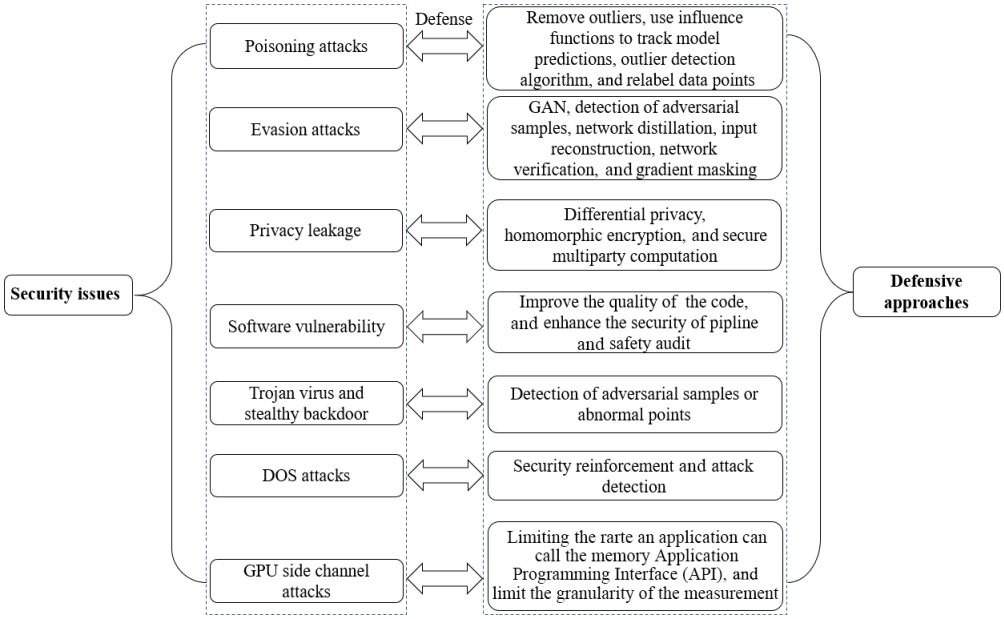

Fig. 10 Relationship between attacks and defensive approaches in deep learning.

that deep learning frameworks may be exposed to in practical applications. By simulating real physical scenarios, we analyzed potential problems in the implementation of algorithms.

The case of an attack in deep learning is shown in Fig. 11. We choose road signs as our research sample, because road signs are relatively simple, thereby making hidden disturbances challenging. In addition, road signs exist in noisy and changeable environments, such as the distance and angle of the observation camera used as well as lighting conditions. Moreover, this case has high research value. Traffic signs, as important elements affecting vehicle safety, should be accurately recognized by algorithms despite the presence of adversarial physical disturbances.

For a deep learning algorithm to realize the correct identification of road traffic signs, various forms of attacks against it may exist. Figure 11 shows an adversarial example that applies algorithms to construct robust perturbations against the deep learning implementation.

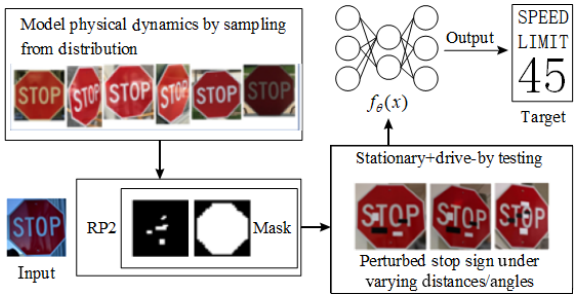

Fig. 11 Case of attack in deep learning[38].

In the article on robust physical-world disturbances[38], designers adopted standard physical science techniques and proposed a two-stage experimental design to verify the robustness of the above physical-world attack algorithm. The first stage was a lab test in which the viewing camera was set to various distance/angle configurations. The second stage was a field test in which a car was driven toward an intersection in uncontrolled conditions to simulate an autonomous vehicle. The test used two datasets, that is, Laboratory for Intelligent & Safe Automobiles (LISA), which is a US traffic sign dataset containing 47 different road signs, and the German Traffic Sign Recognition Benchmark (GTSRB). Two classifiers were built. The LISA-CNN used LISA, and the second classifier, namely, the GTSRB-CNN, was trained on the GTSRB. Both classifiers demonstrated high recognition accuracy. Using two types of introduced attacks (i.e., object-constrained poster and sticker attacks), the developers showed that this method produced robust perturbations for real road signs. The poster attacks were successful in 100% of the stationary and drive-by tests against the LISA-CNN, and the sticker attacks were successful in 80% of the stationary testing conditions and 87.5% of the extracted video frames against the GTSRB-CNN.

Several examples of adversarial AI competitions are shown in Table 5. Numerous machine learning competitions have emerged, and confrontation learning projects have become very important. Competitions discuss the security of real-world AI models and

Table 5 Adversarial AI competitions.

Competition title	Sponsor	Competition content	Dataset	Champion team
NIPS 2017 Adversarial Attacks and Defenses	Kaggle and NIPS	Targeted attacks, untargeted attacks, and defense against attacks	A new dataset compatible with ImageNet	TSAIL team won all three competitions
ASVspoof 2019	EURECOM, NEC, and so on	Automatic speaker verification spoofing and countermeasures	ASVspoof 2019 dataset contributed by an institution or school, such as Google, USTC, and so on	Tsinghua University
NIPS Adversarial Vision Challenge 2018	NIPS and AWS	To facilitate measurable progress toward robust machine vision models and generally applicable adversarial attacks	NIPS Adversarial Vision Challenge 2018 dataset	Robust Model Track and Targeted Attack Track: Petuum-CMU; Untargeted Attack Track: LIVIA
IJCAI-19	Alibaba Security	To explore the security of AI models; participants can either generate adversarial samples or construct a robust model	Product pictures from the Alibaba e-commerce platform	University of Science and Technology of China (USTC) and so on

demonstrate substantial progress in attack and defense methods, which advance theoretical research on practical applications.

7 Conclusion and Future Research Direction

7.1 Conclusion

Starting from the basic composition structure and principles of deep learning, this study describes security problems behind the application of deep learning and summarizes classic attack algorithms for deep learning technologies and development processes. Moreover, it also confirms that adversarial samples against deep learning are widespread. Studying confrontational algorithms can help us better understand and learn deep learning principles and its training and prediction processes.

In this study, algorithm cases of deep learning attacks in recent years are summarized and analyzed; and defense techniques against countermeasure technologies are listed. Furthermore, examples of software flaws in specific implementations are provided. Deep learning prediction is susceptible to slight disturbances, thereby indicating that the deep learning structure has large defects, which is one of the factors hindering its further development. Deep learning can achieve very high prediction accuracy in fixed problems, such as image classification. However, in dynamic real-time scenes with complex interactions with the environment, making mistakes and misjudging emerging scenes are easy. This situation is also an AI technology bottleneck. Therefore, studying the security issues behind deep learning architecture algorithms has far-reaching significance.

7.2 Future research direction

(1) Attacks and defensive approaches in deep learning are continuously being developed. The two elements involve a long-term development process, from the discovery of the poor robustness of deep learning to the emergence of various defensive approaches. Along with this process, both are constantly being developed and improved.

(2) The widespread existence of adversarial samples is helpful for improving the robustness of deep learning algorithms. Deep learning prediction results have considerable deviations in slight disturbances, thereby indicating that deep learning algorithms require a long period of time. Progress is ongoing, and current developments remain incomplete and immature.

(3) In deep learning technologies, computers require relatively high parallel computing power owing to large amounts of training data. Therefore, the industry moved from CPUs to GPUs with numerous nodes. However, underlying software architecture support may also demonstrate various problems, such as data privacy and security.

(4) Although some neural networks perform well in the experimental stage, deep learning systems constructed by neural networks do not perform well in the application stage. The physical world is complex and dynamic, with hidden unknown influencing factors. Therefore, we must fully consider various influencing factors and add them to the training process to construct deep learning systems.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2018YFB0803403); Fundamental Research Funds for the Central Universities (Nos. FRF-AT-19-009Z and FRF-BD-19-012A), and National Social Science Fund of China (No. 18BGJ071).

References

[1] W. W. Jiang and L. Zhang, Geospatial data to images: A deep-learning framework for traffic forecasting, *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 52–64, 2019.

[2] L. Zhang, C. B. Xu, Y. H. Gao, Y. Han, X. J. Du, and Z. H. Tian, Improved Dota2 lineup recommendation model based on a bidirectional LSTM, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 712–720, 2020.

[3] H. M. Huang, J. H. Lin, L. Y. Wu, B. Fang, Z. K. Wen, and F. C. Sun, Machine learning-based multi-modal information perception for soft robotic hands, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 255–269, 2020.

[4] X. Y. Yuan, P. He, Q. L. Zhu, and X. L. Li, Adversarial examples: Attacks and defenses for deep learning, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, 2019.

[5] J. C. Hu, J. F. Chen, L. Zhang, Y. S. Liu, Q. H. Bao, H. Ackah-Arthur, and C. Zhang, A memory-related vulnerability detection approach based on vulnerability features, *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 604–613, 2020.

[6] C. Szegedy, W. Zaremba, I. Sutskever I, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv: 1312.6199, 2013.

[7] A. Athalye, N. Carlini, and D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, arXiv preprint arXiv: 1802.00420, 2018.

[8] Y. T. Xiao, C. M. Pun, and J. Z. Zhou, Generating adversarial perturbation with root mean square gradient, arXiv preprint arXiv: 1901.03706, 2019.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv: 1412.6572, 2014.

[10] J. W. Su, D. V. Vargas, and K. Sakurai, One pixel attack for fooling deep neural networks, *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, 2019.

[11] W. He, J. Wei, X. Y. Chen, N. Carlini, and D. Song, Adversarial example defense: Ensembles of weak defenses are not strong, in *Proc 11th USENIX Workshop on Offensive Technologies*, Vancouver, Canada, 2017.

[12] G. W. Xu, H. W. Li, H. Ren, K. Yang, and R. H. Deng, Data security issues in deep learning: Attacks, countermeasures, and opportunities, *IEEE Comm. Mag.*, vol. 57, no. 11, pp. 116–122, 2019.

[13] M. I. Tariq, N. A. Memon, S. Ahmed, S. Tayyaba, M. T. Mushtaq, N. A. Mian, M. Imran, and M. W. Ashraf, A review of deep learning security and privacy defensive techniques, *Mobile Inf. Syst.*, vol. 2020, p. 6535834, 2020.

[14] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, and S. Yoon, Security and privacy issues in deep learning, arXiv preprint arXiv: 1807.11655, 2018.

[15] S. L. Qiu, Q. H. Liu, S. J. Zhou, and C. J. Wu, Review of artificial intelligence adversarial attack and defense technologies. *Appl. Sci.*, vol. 9, no. 5, p. 909.

[16] S. M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, DeepFool: A simple and accurate method to fool deep neural networks, presented at 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2574–2582.

[17] Q. X. Xiao, K. Li, D. Y. Zhang, and W. L. Xu, Security risks in deep learning implementations, presented at 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2018, pp. 123–128.

[18] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, The limitations of deep learning in adversarial settings, presented at 2016 IEEE European Symp. Security and Privacy (EuroS&P), Saarbrucken, Germany, 2016, pp. 372–387.

[19] A. Kurakin, I. Goodfellow, and S. Bengio, Adversarial examples in the physical world, arXiv preprint arXiv: 1607.02533, 2016.

[20] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, Adversarial attacks on neural network policies, arXiv preprint arXiv: 1702.02284, 2017.

[21] H. W. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, Walking on the edge: Fast, low-distortion adversarial examples, arXiv preprint arXiv: 1912.02153, 2019.

[22] B. Nelson, B. I. P. Rubinstein, L. Huang, A. D. Joseph, S. J. Lee, S. Rao, and J. D. Tygar, Query strategies for evading convex-inducing classifiers, *J. Mach. Learn. Res.*, vol. 13, pp. 1293–1332, 2012.

[23] G. Ateniese, G. Felici, L. V. Mancini, A. Spognardi, A. Villani, and D. Vitali, Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers, arXiv preprint arXiv: 1306.4447, 2013.

[24] N. Narodytska and S. P. Kasiviswanathan, Simple black-box adversarial perturbations for deep networks, arXiv preprint arXiv: 1612.06299, 2016.

[25] P. Y. Chen, H. Zhang, Y. Sharma, J. F. Yi, and C. J. Hsieh, ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in *Proc. 10th ACM Workshop on Artificial Intelligence and Security*, Dallas, TX, USA, 2017, pp. 15–26.

[26] H. S. Ye, Z. C. Huang, C. Fang, C. J. Li, and T. Zhang, Hessian-aware zeroth-order optimization for black-box adversarial attack, arXiv preprint arXiv: 1812.11377, 2018.

[27] X. R. Li, S. L. Ji, M. Han, J. T. Ji, Z. Y. Ren, Y. S. Liu, and C. M. Wu, Adversarial examples versus cloud-based detectors: A black-box empirical study, arXiv preprint arXiv: 1901.01223, 2019.

[28] S. Saxena, TextDecepter: Hard label black box attack on text classifiers, arXiv preprint arXiv: 2008.06860, 2020.

[29] A. Zimba, H. S. Chen, and Z. S. Wang, Bayesian network based weighted APT attack paths modeling in cloud computing, *Future Generation Comput. Syst.*, vol. 96, pp. 525–537, 2019.

[30] H. S. Chen, C. X. Meng, Z. G. Shan, Z. C. Fu, and B. K. Bhargava, A novel low-rate denial of service attack detection approach in zigbee wireless sensor network by combining Hilbert-Huang transformation and trust evaluation, *IEEE Access*, vol. 7, pp. 32 853–32 866, 2019.

[31] J. Steinhardt, P. W. Koh, and P. Liang, Certified defenses for data poisoning attacks, presented at 31st Conf. Neural Information Proc. Systems, Long Beach, CA, USA, 2017, pp. 3517–3529.

[32] P. W. Koh and P. Liang, Understanding black-box predictions via influence functions, arXiv preprint arXiv: 1703.04730, 2017.

[33] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, Detection of adversarial training examples in poisoning attacks through anomaly detection, arXiv preprint arXiv: 1802.03041, 2018.

[34] A. Paudice, L. Muñoz-González, and E. C. Lupu, Label sanitization against label flipping poisoning attacks, in *Joint European Conf. Machine Learning and Knowledge Discovery in Databases*, A. Paudice and L. Muñoz-González, eds. Cham, Germany: Springer, 2018, pp. 5–15.

[35] N. Carlini and D. Wagner, Towards evaluating the robustness of neural networks, presented at 2017 IEEE Symp. Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 39–57.

[36] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy, presented at Proc. 33rd Int. Conf. Machine Learning, New York, NY, USA, 2016, pp. 201–210.

[37] S. Lee, H. Kim, J. Park, J. Jang, C. S. Jeong, and S. Yoon, TensorLightning: A traffic-efficient distributed deep learning on commodity spark clusters, *IEEE Access*, vol. 6, pp. 27 671–27 680, 2018.

[38] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. W. Xiao, A. Prakash, T. Kohno, and D. Song, Robust physical-world attacks on deep learning visual classification, presented at 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 1625–1634.

Hongsong Chen received the PhD degree in computer science from Harbin Institute of Technology in 2006. He is a member of the IEEE, he is a professor at University of Science and Technology Beijing (USTB), China since 2008. He was a visiting scholar at the Department of Computer Science, Purdue University from 2013 to 2014. He is a high-level member of China Computer Federation. His research interests include cloud computing and cloud security, wireless network and pervasive computing, and trust computing. He got the excellent young academic paper award in USTB in 2009. He has published more than 50 academic papers and 5 books.

Yongpeng Zhang is a master student at University of Science and Technology Beijing (USTB), China since 2018. His research areas include information security, machine learning, and big data.

Yongrui Cao is a master student at University of Science and Technology Beijing, China since 2018. His research areas include information security, machine learning, and pervasive computing.

Jing Xie received the master degree in computer science at Beijing University of Post Telecommunication in 2011. He is a senior engineer in China Industrial Control System Cyber Emergency Response Team. His research areas include cyber-space security and artificial intelligence.