A survey of Privacy and Security Issues in Machine Learning

Liwan Zhou
Electrical Engineering
Stevens Institute of Technology
Hoboken, USA
lzhou23@stevens.edu

***Abstract*—Artificial intelligence has penetrated into every corner of life, bringing great convenience to human beings. Especially in recent years, with the vigorous development of deep learning in machine learning, there are more and more applications in daily life. Unfortunately, machine learning systems also face many security risks, which are further amplified by the popularity of machine learning systems. In order to reveal these security risks and realize a powerful machine learning system, the mainstream deep learning systems were investigated. Firstly, an analysis model is designed to analyze the deep learning system, and the survey scope is defined. The deep learning system covered four areas – image classification, speech recognition, malware detection and natural language processing. It extracted four types of security risks and characterized and measured them from multiple dimensions, including complexity, attack success rate and damage. Then, the defense technology and its characteristics of deep learning system are investigated.Finally, through the observation of these systems, some suggestions are put forward to build a robust deep learning system.**

***Index Terms*—machice learning security, deep learning security, adversarial attack, membership inreference attack**

I. Introduction

Widely application of deep learning does not guarantee its safety, it brought the new threats and attacks every day, they compromise deep learning model, which endangers people's privacy, financial assets and safety. As a new technology, the security problems of deep learning often neglected, therefore, learning the security problems of deep learning systematically and further put forward effective measures are urgent and critical.

Deep learning has been widely used in image classification, speech recognition, natural language processing, malware detection and other fields. Due to the tremendous advances in the computing power and the sharp increasing of data volume, deep learning in these scenarios showed superior potential. Deep learning is especially good at characteristics of unsupervised learning, to deepen understanding of an object, with strong ability of prediction. However, deep learning systems are under the threat from a series of planned attacks, such as deep learning systems are easily fooled by adversarial example, which can lead to the wrong classification. On the other hand, users who use online deep learning systems for classification have to disclose their data to the server, which leads to privacy leakage.

In order to fully understand the privacy and security issues in deep learning, we investigated relevant literature and systems, these inverstigations span four areas: image classification, speech recognition, natural language processing, and malware detection. Based on the survey work, these attacks were classified into four categories: model extraction attack, model inversion attack, poisoning attack and adversarial attack. Model extraction and inversion attack are aimed at privacy, the former mainly steals the information of the model, while the latter mainly obtains the information of the training dataset. Poisoning attack and adversarail attack are aimed at security, the former mainly puts malicious data in the training stage to reduce the classification accuracy of the model, while the latter mainly creates adversarial examples in the prediction stage to deceive the model.

This paper mainly studies the scope of machine learning security, the basic components of the whole learning system, attack methods, defense measures, practical evaluation and valuable phenomena and conclusions, including four contributions:

1) Systematic analysis of attack and defense technologies. Four types of attack and three types of defense are summarized, and the privacy and security issues of machine learning system are investigated and summarized comprehensively.

2) Modularization of machine learning system.The machine learning system was analyzed, and the safety knowledge of machine learning was systematically summarized according to the time line of preparation process, training process and prediction process, and the space line of training dataset, training algorithm, model structure, model parameters, prediction data and results.

3) The division of specific technologies within each attack and defense type.This paper analyzes the attack and defense techniques in each type of attack and defense, classifies the numerous technical articles, and analyzes the differences and advantages and disadvantages of different techniques.

4) Based on the observation and summary of the security problems of machine learning systems, as well as the analysis and research of these attack and defense technologies, the suggestions of building secure machine learning systems and protecting the privacy and security of all machine learning participants are proposed.

II. Related work

At present, some literatures have investigated and evaluated the attack and defense of machine learning.In the early work, Barreno et al. [1] investigated the security of machine learning and classified the attacks against machine learning systems. They ran experiments on a filter that counted spam, dissecting attacks from three dimensions: how they operate, how they affect input, and how common they are. Amodei et al. [2] introduced five possible research problems related to accident risk in machine learning and discussed possible solutions by taking cleaning robots as an example according to their working principles.

Papernot et al. [3] reviewed previous work on machine learning system attacks and corresponding defenses.Unlike previous surveys and reviews, they focused on a comprehensive literature review of security threats. Bae et al. [4] summarized the attack and defense methods of artificial intelligence(AI) under the concept of security and privacy, they check for adversarial attack and poisoning attacks in black and white boxes. Subsequently, Papernot et al. [5] systematically studied the security and privacy of machine learning and proposed a threat model of machine learning, they introduce attack methods according to the classification of training process and prediction process, black box model and white box model. However, they don't deal much with the widely used deep learning models. Liu et al. [6] focused on the two stages of machine learning, namely the training stage and the prediction stage, and provided a comprehensive literature review. They divided the corresponding defense measures into four categories. In addition, their research focuses more on issues such as combating the data distribution drift caused by samples and the leakage of sensitive information caused by machine learning algorithms.

Akhtar et al. [7] comprehensively studied the adversarial attacks on deep learning in the field of computer vision, and summarized 12 different types of attack methods. In addition to the usual Convolution Neural Networks (CNN), they also looked at attacks on other models (such as autoencoders, generated models, Recurrent Neural Networks (RNN)) and attacks in the physical world, and they also summarized various defense methods. However, the research content of this work is limited to adversarial attacks in the field of computer vision. DeepSec [8], developed by Ling et al., is a unified evaluation platform. DeepSec integrates 16 attack methods and 13 defense methods against learning to measure the vulnerability of deep learning models and to evaluate the effectiveness of various attacks and defenses.

This paper investigates and summarizes the privacy and security issues of machine learning systems, and classifies the attack and defense methods, analyzes the attack and defense technologies under different categories, and introduces their applications in different fields of image classification, speech recognition, natural language processing and malware detection.

III. overview

A. Machine learning system

Supervised machine learning is mainly divided into two stages: model training and model prediction (inference). In the model training process, the training dataset is used as input, and the model is finally generated. The model prediction process accepts input from the user or attacker and provides the prediction results. In order to complete these two processes, the model designer must specify the training data and training algorithm to be used. The optimized training model and relevant parameters are generated in the model training process. Before running the training algorithm, traditional machine learning requires manual extraction and selection of features, while deep learning entrusts the training algorithm to automatically identify reliable and effective features. Typically, a trained model can be deployed for commercial use.In business applications, the model calculates the most likely outcome based on the input received. Take malware detection as an example, security analysts first collect data (possibly raw data) from malware, extract representative features and build a classification model to detect malware.

To formalize the process of machine learning systems, some symbols are given in Tab. I. Given a machine learning task, the data collected can be expressed as $x=(x^{(1)},x^{(2)},\cdots,x^{(n)})$. Dataset D is a set of x. F is a machine learning system, it can calculate the corresponding result y according to the given input x, that is, $y=F(x)$. In the process of model training, the loss function is used to measure the prediction error of the real results. In the training process, the minimum error value is expected to be obtained by fine-tuning the parameters. The loss function can be calculated as $L=\sum_{1\leq i\leq n}\left\|y_p^{(i)}-F\left(x^{(i)}\right)\right\|^2$, where y_p represents the real result.Therefore, the process of model training can be expressed as

$$\arg\min_F L \tag{1}$$

TABLE I
Formalization in Machine Learning System

Symbol	Definition
D	Dataset
$x^{(1)},x^{(2)},\cdots,x^{(n)}$	Input Data
$y^{(1)},y^{(2)},\cdots,y^{(n)}$	Output Result
F	Model
w_{ij}^k	Weights Parameters
b_j^k	Bias Parameters
λ	Hyperparameters
x_t	Prediction Input
y_t	Prediction Output
δ	Perturbation

B. Security threats

Fig. 1 shows the vulnerability of a classic deep learning model in the process of prediction and training. Recent research has shown that machine learning systems are fragile and vulnerable to specific attacks.

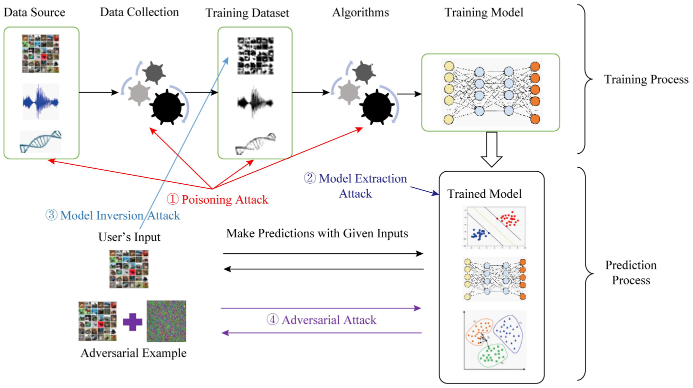

Fig. 1. Overview of attacks in machine learning system.

According to the target, these attacks can be divided into four categories: model extraction attack, model inversion attack, poisoning attack and adversarial attack. In this section, we will detail these attacks with examples and their formal definitions.

1) poisoning attack: poisoning attack mainly refers to the manipulation of machine learning model prediction by attacking training datasets or algorithms during training or retraining process. Since data is usually non-stationary in the field of secure machine learning, and its distribution may change with time, some models are generated not only during training process, but also during periodic retraining process. The methods of attacking the training dataset mainly include polluting source data, adding malicious samples to the training dataset, modifying some labels in the training dataset, and deleting some original samples in the training dataset. Poisoning attack will increase the difficulty of training the appropriate model. It can also add a back door to the generated model for the attacker, who can make the prediction of the model biased in the direction the attacker wants [9].

2) model extraction attack: model extraction attack occurs on a well-trained model and is mainly used to steal model parameters and illegally obtain the model, which violates the confidentiality of the training model. in the setting of machine learning as a service (MLaaS), the model itself in a safe custody of cloud services, it allows users through the prediction application programming interface (API) query model based on cloud. The model owner implements the business value of the model by making the user pay for the predictive API, so the machine learning model is a trade secret. In addition, the training process of a model requires the collection of a large number of datasets, as well as a large amount of time and computing power, so once the model is extracted and abused, it will bring huge economic losses to the model owner.

3) model inversion attack: in the early understanding, there was only one information flow between the training dataset and the training model, that is, from the dataset to the model. In fact, many studies have shown that there is also a reverse flow of information, that is, recovering dataset information from model information, which is called model inversion attack. Model inversion attack refers to extracting the training dataset information from the model. It mainly includes membership inference attack (MIA) and property inference attack (PIA). MIA mainly deduces whether there is a specific record in the dataset, that is, it judges the degree of membership, which is the focus of current research. PIA mainly obtains the information of attributes such as gender distribution, age distribution, income distribution and prevalence rate in the dataset. Model inversion attacks steal the private information of members of the training dataset and damage the business value of the dataset owner. This happens for two reasons: inadequate privacy protection, such as information disclosure [10];Unsafe algorithms [11]. In order to strengthen the protection of personal privacy, the EU promulgated GDPR in 2018, which clearly defines the privacy of personal data and strictly protects it [12].

4) adversarial attack: adversarial attack refers to submitting the adversarial example to the trained model so as to make the model predict wrong, which is also known as evasion attack. .The adversarial example is adding a slight disturbance to the original normal sample, which can lead to the misclassification of the model. Similarly, in the field of speech and text recognition, the countersample did not make any noticeable modification to the original text. In the field of malware detection, malware authors add some special statements to their software to avoid detection by anti-virus software.

IV. Peivacy

Privacy is a common problem in information security domain. Broadly speaking, privacy includes valuable assets and the right to protect data from theft, inference and interference. Deep learning is built on massive amounts of data, the trained model is actually a data model, and the trained model requires a lot of interaction with the test data from individuals, therefore, privacy becomes even more important. In this section, we will introduce the privacy problems in deep learning systems and present the current research status from two aspects: attack and defense.

A. Introduction to privacy issues

Based on the whole deep learning process, we classified the objects of privacy protection as :1) training datasets; 2) model structure algorithm and model parameters; 3) forecast data and results.

The training model in deep learning is a data model and also an abstract representation of training data. In the modern deep learning system, the training stage needs to process a large amount of data and multi-layer training, which has strict requirements on high-performance computing and massive storage. In other words, the trained model is considered as the core competitiveness of the deep learning system. In general, the training model contains three types of data assets: :1) models, such as tradition machine learning and deep neural networks; 2) hyperparameters, designed the structure of the training algorithm, such as the number of network layers and the number of neurons; 3) parameters, is calculation coefficient from one layer to another layer of multi-layer neural network.

In this case, the trained model has critical business and innovative value. Once the model is copied, leaked, or extracted, the interests of the model owner are severely compromised. Privacy comes from users and providers of deep learning systems in terms of predicting inputs and outcomes. A malicious service provider may retain the user's pre-measured data and results in order to extract sensitive information from them or for other purposes. On the other hand, the inputs and results may be attacked by criminals who can use the data to make profits for themselves.

B. Research on privacy issues

At present, the mainstream privacy destruction methods are model extraction attack and model inversion attack. The main difference between these two is that the former focuses on the privacy information of the model and the latter on the privacy information of the dataset.

In the model extraction attack, the attacker sends a large amount of prediction data to the model through the API provided by the deep learning system, then receives the class label and confidence coefficient returned by the model, calculates the parameters of the model, and finally restores the original model. This kind of attack can destroy the privacy of the model itself, harm the interests of the model owner, create business value for the attacker, and also help to realize the model reversion attack and adversarial attack.

In the model reverse attack, the attacker obtains the confidence coefficient of the model by providing the pre-measured data to the model, thus damaging the privacy of users or datasets. As described in section 2, reversion attacks include membership inference attacks (MIA) and property inference attacks (PIA). In MIA, an attacker can infer whether a specific record is included in the training dataset or not. In PIA, the attacker can speculate whether there is a certain statistical feature in the training dataset. Recent studies have found that in the population training dataset, the underrepresentation of samples of people at certain levels (e.g., women and ethnic minorities) can affect the performance of the final model [13].

In terms of implementation, privacy protection can be divided into four technologies :1) differential privacy(DP) [14], [15]; 2) homomorphic encryption(HE) [16], [17]; 3) secure multi-party computation (SMC) [18], [19]; 4) suboptimal choice(SC) [20], [21].

C. Attack methods

This section describes in detail the technical methods of three privacy attacks.

1) model extraction attack: Model extraction attacks destroy the privacy of the model itself, and the attackers try to steal the parameters and hyperparameters of the model. At present, the mainstream method realizes model extraction by constructing accurate model or similar model. Accurate model refers to the attacker's attempt to reconstruct the original model, or to calculate parameters or hyperparameters from the original model;The similarity model is an alternative model constructed by the attacker that is similar in terms of predictive performance.

The research of model extraction attack is mostly carried out under the black box model, and the algorithm of training model can only be obtained under the black box model.

- accurate model: Ramer et al. [22] introduced a method of extracting models through predictive API. They set up the model equation by sending a large number of queries and got the corresponding prediction results. However, this method is only applicable to certain machine learning modes such as decision tree, logistic regression and simple neural network, and is not applicable to DNN. Wang et al. [21] attempted to steal the hyperparameters on the premise of knowing model algorithm and training data. The superparameter, called λ in this article, balances the loss function and regularization term in the objective function. Since the training process requires the minimum objective function, the gradient of objective function at the model parameters is 0. According to this property, the attacker can obtain many linear equations through the query of the model, that is, the relationship between the parameters ,the hyperparameter and the input data. Finally, the linear least square method is used to estimate the hyperparameters. Baluja et al. [23] trained a classifier named meta model to predict model properties. The attacker submits the query input to the target model and takes the output provided by the target model as input to the meta model,then the meta model tries to output the attributes of the target model.
- similar model: similar model only requires that the performance of the model be similar to the original model, and it is mainly used to generate adversarial examples. Papernot et al. [**?**] attempted to generate portable, untargeted adversarial examples. The attacker used Jacobian-based dataset augmentation (JbDA) technology to generate composite samples to query the target model, and established an attack model that approximated the decision boundary of the target model. The attacker then uses the attack model to generate adversarial examples, which will be misclassified by the target model due to its portability. Juuti et al. [24] proposed a new synthetic data generation method through regularization of DNN training and generalization of JbDA. Taking into account the differences between different models, Papernot they [25] also found that knowledge of the target model architecture was unnecessary, since any machine learning model could be replaced by a more complex model, such as DNN.

2) membership inference attack: Truex et al. [26] proposed a universal system scheme for MIA in MLaaS platform. Given an example x and black box access to the classification model F_t trained on dataset D, when training F_t, whether the adversary can confidently infer whether x is included in D. In MIA, theadversary cares more about whether x is in D than the content of x. at present, the MIA can be realized through three methods:

- Training the attack model.
  The attack model is a binary classifier used to infer the information recorded by the target. It transforms the membership inference problem into a classification problem, which can be used for both white and black box attacks. In many studies, shadow model is also introduced to train the attack model, which is mainly used to simulate the target model and generate the dataset needed for the attack model.
  Shokri et al. [27] used API calls in machine learning to implement and evaluate the MIA method of the black box model. They generated a dataset similar to the target training dataset and used the same MLaaS to train the shadow model. Salem et al. [28] relaxed some of the constraints in the literature [27] (shadow model should be trained on the same MLaaS, and the datasets of the shadow model and the target model have the same distribution), and only one shadow model was used in the absence of the knowledge structure of the target model and the distribution of the training dataset.
- Probability information calculation.
  This method uses probability information to infer the scribe degree. However, this method requires certain assumptions and auxiliary information to obtain reliable probability vector or binary results, which is also a limitation of this method.
  Fredrikson et al. [29] attempted to construct the probability of whether a certain data appeared in the target training dataset based on the probability information. Then the input data with the highest probability is found and the data obtained is similar to the data in the target training dataset. The third attack method in Salem et al. [28] only needs to record the probability vector output by the target model, and use the statistical measurement method to compare whether the maximum classification probability exceeds a threshold. If so, the record is considered to belong to the dataset. Long et al. [30] put forward the method of generalized MIA, unlike literature [27], it's easier to attack the fitting data. They trained a large number of reference model similar to the target model (similar to shadow model), choose according to the reference model output probability information vulnerable data, then the target model and comparing the output of the reference model, calculate the probability of data belongs to target training dataset.
- Similar sample generation.
  The method generates training records through the model generated by training, and the generated samples are similar to those of the target training dataset.
  Liu et al. [31] and Hayes et al. [32] both explored the method of attacking the generated model. Different from the discriminant model, the generated model is usually used to learn the distribution of data and generate similar data. Literature [31] proposed a white box attack for single member attack and joint member attack. Considering that the method in literature [27] is difficult to attack CNN, Hitaj et al. [10] proposed a more general MIA method to execute a white box attack in the scenario of cooperative deep learning model. They constructed a target classification model generator and formed a generative adversarial network (GAN) with the generator. The limitation of this method is that all samples belonging to the same category need to be visually similar, therefore, they can not be distinguished under the same category.

3) property inference attack: PIA refers to reasoning the statistical properties of the training dataset.

Ateniese et al. [33] first proposed a white box attack method for training meta-classifier. The classifier takes the feature information of the model as input and the training whether the dataset of the model contains specific attributes as output. They also train shadow models to provide training data for the meta-classifier. Ganju et al. [34] constructed a meta-classifier model, which studied how to extract the eigenvalues of DNN and use them as input to the meta-classifier and other parts are very similar to those in literature [33].

In addition, Melis et al. [35] proposed a collaborative learning white box attack method to address the shortcomings in literature [10]. The theoretical basis is that deep learning models remember too many data features [11]. An attacker can download the latest model multiple times, get the updated model for each phase, subtract out the aggregated updates for different phases, and analyze the updated information to infer members and attributes. They trained a binary classifier to estimate the attributes of the dataset, using the updated gradient values as input.

D. Defense methods

1) Differential privacy: Differential privacy is a cryptographic tool designed to maximize the accuracy of data queries and minimize the chance to identify records when querying statistical databases. Based on the protection goal, the differential privacy method can be extended from the aspects of output disturbance, target disturbance and gradient disturbance [15].

Chaudhuri et al. [36] first proposed the output and target perturbations, which strictly proved that privacy was maintained in the convex loss function machine learning model and was realized as regular logistic regression. The output disturbance includes a noise training model based on boundary sensitivity and increased sensitivity. However, Wang et al. [15] showed that the output disturbance could not be generalized under non-smooth conditions. Zhang et al. [37] proposed that in the case of strong convexity, an appropriate learning rate could be used to improve the operation speed and practicability. Target perturbation is a model for training the minimization of objective function with random items, which is superior to output perturbation in both theory and experience [38]. In order to obtain better performance or support other scenarios, Kifer et al. [39] improved the accuracy by selecting gaussian distribution instead of gamma distribution, and introduced the first differential privacy algorithm for high-dimensional sparse regression.The algorithm and proof of Lipschitz loss function are given in literature [40] and [39].

Song et al. [41] proposed gradient disturbance, its main idea is adding noise during each iteration updating parameters. This method is not subject to strong convex function or strong perturbation restricted optimization problem, has certain superiority in practical application. However, the stochastic gradient descent (SGD) or gradient descent (GD) calculation process is very time consuming, if the dataset is very large, calculation may cost a lot of time. For strongly convex , Bassily et al. [42] and Talwar et al. [40] relaxed the restrictions on Lipschitz convex functions and strict error boundaries. Then, Abadi et al. [14] dealt with non-convex objective functions and trained DNN at a moderate cost. They modified and extended DP-SGD to allow different layers to have different limiting thresholds and noise scales. Subsequently, Zhang et al. [37] first presented the theoretical results of non-convex optimization problems. Literature [15] implements a non-convex case that satisfies the Polyak-Lojasiewicz condition and produces a tighter upper bound. Combining with other algorithms, Zhang et al. [43] also applied gradient perturbation in distributed ERM.

Hamm et al. [44] proposed a method for constructing a global differential private classifier using a local classifier, which does not require access to the private data of either party. Hynes et al. [45] proposed a deep learning framework Myelin, which is used to implement an efficient and private data-independent actual deep learning model in the field of trusted hardware.

Several metrics have been used to estimate privacy risks, the simplest measure is to calculate the sum of privacy consumption [46] based on the composability of differential privacy.

2) Homomorphic encryption: General encryption schemes focus on the security of data storage, while homomorphic encryption (HE) focuses on the security of data processing. HE is usually used in the risk of data leakage. Due to the high complexity of decryption, HE can effectively Protect sensitive data from being decrypted and stolen. In deep learning, it is mainly used to protect predicted inputs and results, train neural network models, etc. The main negative effect of applying HE is reduced efficiency, that is, error transmission problems, ciphertext Long operation time, sharp increase in data volume after encryption, etc.

Liu et al. [47] proposed MiniONN, which is a neural network that supports privacy protection and ensures that the server does not understand the input and the client does not understand the model. The main idea is to allow the server and the client to add additional layers for each layer of the neural network to share input and output values. Jiang et al. [17] gave a practical algorithm for matrix and cipher matrix arithmetic operations. Phong et al. [48] proposed a privacy-protected DL system that uses asynchronous stochastic gradient descent to apply neural Network connection deep learning and cryptography, combined with additive HE. In other respects, Hesamifard et al. [49] developed CryptoDL for running DNN on encrypted data with an accuracy rate of 91.5% on CIFAR-10. In CNN, a low-order polynomial is used to design an approximate function, then the approximate polynomial is used to replace the original activation function to train the CNN, and finally the CNN is implemented on the encrypted data.

3) Multi-party computation: In reality, it is often encountered that multiple data parties want to learn a model on a server together. However, each data party is reluctant to share its own data with other parties. In the case where there is only one server for multiparty data, Shokri et al. [50] implemented a system that allows multiple parties to jointly learn the model without sharing the input dataset. All parties can use the final model independently. In the training process, each data party conducts model training on its local dataset, and then uploads the key gradient of the selected parameter to the global parameter library, and then downloads the latest value of the required parameter. Hong et al. [48] made improvements based on the literature [50]. Each data side uploads the gradient after adding HE encryption, and applies asynchronous SGD to the model. In addition, Phong et al. [51] also proposed a server-assisted network topology and a fully connected network topology system. The parties share the weight of the neural network instead of the gradient. They can not only prevent malicious servers, but also prevent data collusion even if there is only one honest party.

Another scenario in multi-party computing is that the data party does not want to give all the training data to a server to train the model. He hopes to distribute the dataset to multiple servers and train the model together, each server will not understand the training data of other servers. SecureML [18] is a dual-server model protocol to protect privacy. The data owner assigns private data to two non-collusive servers, and uses secure two-party computing techniques to train the joint data to support secure arithmetic operations. Inadvertent transmission and encryption circuits are used, and a multi-party computing-friendly activation function is used. Liu et al. [47] proposed a neural network MiniONN that supports privacy protection. It ensures that the server knows nothing about input and the client knows nothing about the model. The main idea is to allow the server and client to additionally share the input and output values of each layer of the neural network.

The more general scenario of multi-party computing is that M data parties want to use N servers to train their joint data. Any data party or server is required to have no knowledge of the training data of any other data party. In SecureNN [19], $N = 3$ or 4, M can be any value. In addition, the trained model is shared as a secret and hidden from any single server or data terminal. These secret sharing can be combined by the server or any other party to reconstruct the model.

4) Suboptimal choice: Tramèr et al. [22] proposed the first defensive method with a quantitative model to extract the probability of attack prediction. They only allow an attacker to extract a given class label without providing a confidence score or rounding confidence. This method reduces the amount of information provided to the attacker, but also reduces the legitimate services. Later literature [24] showed that the model extraction attack is effective even if the predicted probability is not used. But Lee et al. [20] found that injecting noise into the class probability can still prolong the attack time. The attacker is forced to give up the probability information and only use the label information, which greatly increases the number of queries and attack time. Wang et al. [21] found that rounding the model parameters increases the attacker's estimation error for hyperparameter attacks. Unfortunately, the impact of this error on test performance is negligible.

Another method is to find anomalies from query requests submitted by users. Kesarwani et al. [52] relied on recording all requests from the client and calculating the feature space composed of normal requests. When it is detected that the new request space exceeds the predetermined threshold, the model extraction attack is considered to occur. Therefore, they need to linearly separate the prediction classes in the input to evaluate the feature space. In addition, PRADA [24] is based on the detection of sudden changes in the distribution of samples submitted by a given customer. It is assumed that the distribution of features in samples submitted by attackers is more unstable than in benign queries. Once PRADA detects an attack, according to the prediction of the target model, it returns to category 2 or category 3 with the greatest probability. PRADA requires hundreds of queries to detect attacks on documents [**?**], and thousands of queries to attacks on documents [22].

V. Security

A. Introduction to security issues

There are already privacy research objects in the artificial intelligence system, such as the collected training dataset, the parameters of the training model, the prediction data that the user is about to submit, and the results returned by the model. To protect the legal data (model parameters, datasets, etc.) that originally existed in the system is a privacy issue. However, in artificial intelligence systems, malicious samples that cause security problems are often unknown. For example, poisoning attacks add malicious data to the training dataset, which will negatively affect the prediction of deep learning. How to resist such unknown samples is a security issue. In addition, the attacked classification model will not be exposed to these adversarial samples during the training process. These malicious data were not originally in the learning model. To prevent malicious data that may not exist in the system and may cause model errors is a security issue.

B. Research on security issues

In deep learning systems, the training dataset and prediction data need to interact with the user, and the training process and training model are generally closed. Therefore, training datasets and prediction data are more vulnerable to attacks by unknown malicious samples. More specifically, if a malicious sample appears in the training dataset, we call it poisoning attack; if a malicious sample appears in the prediction data, we call it adversarial attack.

Poisoning attacks add malicious samples during training, which affects the generated model. Most malicious sample search methods are implemented by discovering loopholes in algorithms or training processes. Early machine learning algorithms were also vulnerable to poisoning attacks [53]–[55]. The poisoning attack mainly affects the normal model in two aspects. 1) Directly change the decision boundary of the classifier, destroy the normal use of the classifier, make it unable to classify normal samples correctly, and destroy the usability of the model. Attackers use wrong tags to submit data records, or maliciously modify the tags of existing data in the training dataset. 2) Create a backdoor in the classifier. It can correctly classify normal samples, but it will lead to classification errors of specific data. Attackers can carry out targeted attacks through the backdoor, destroying the integrity of the model. In addition, they can also directly attack feature selection algorithms [56]. Correspondingly, the defense method is mainly achieved by enhancing the robustness of the training algorithm [57] and protecting the security of the dataset [58].

In the prediction process, adversarial attacks will add malicious interference to normal samples. The adversarial sample must deceive the classifier and make it imperceptible. The attack is widely used in the field of image recognition, but also in speech processing [59], speech-to-text conversion [60], text recognition [61], malware detection [62], etc. At present, mainstream methods to find disturbances include FGSM [63], JSMA [64], C & W [65], DeepFool [66], UAP [67], ATN [23] and some variants. Some studies have also attacked other deep learning models other than CNN and DNN, and even produced instances of confrontation in the real world. The defensive strategy is mainly considered from the process of confrontation sample generation and attack, including adversarial training [68], region-based classification [69], input change [70], gradient regularization [71], distillation [72], data processing [73] and training defense networks [74].

C. Poisoning attacks

Poisoning attacks attempt to reduce the prediction of deep learning systems by polluting training data. In the early days of machine learning, poisoning attacks were considered to be an important threat to mainstream algorithms. For example, support vector machines [53], [75], [76], Bayesian classifier [54], hierarchical clustering [77], and logistic regression [78] have all been compromised by poisoning attacks. With the widespread use of deep learning, attackers have also turned their attention to deep learning [79]–[81].

Muñoz GGonzález et al. [82] first conducted a poisoning attack on multiple types of problems based on reverse gradient optimization. The algorithm automatically calculates the gradient step by step and reverses the learning process to reduce the complexity of the attack. Most research on poisoning attacks focuses on offline environments, where the classifier is trained on a fixed input. However, in many training processes, data arrives in sequence in the form of streams, that is, online learning. Wang et al. [83] investigated the data poisoning attack of online learning. They formalized the problem into two settings: semi-online and full-online, and used incremental, interval, and teaching-enhanced three attack algorithms. Their online attacks are better than those that ignore the online characteristics of the input data.

In summary, poisoning attacks are essentially seeking global or local disturbances on the training data. The performance of machine learning and deep learning depends largely on the quality of training data. High-quality data should generally be comprehensive, unbiased and representative. In the process of data poisoning, wrong labels or biased data are intentionally processed and added to the training data, reducing the overall quality. It is observed that there are two reasons for poisoning:

1) Error marking data. In classification tasks, deep learning models are usually trained in advance under labeled data. In other words,$L : \{x_1, x_2, \cdots, x_n\} \rightarrow Y$, where Y is a specific label for a given input. By operating the label as $L : \{x_1, x_2, \cdots, x_n\} \rightarrow Y'$ to generate the wrongly labeled data, where Y' is an incorrect label. The acceptance of erroneously labeled data may lead to two kinds of results: deep learning cannot effectively learn decision boundaries; it significantly pushes decision boundaries to incorrect areas. The results show that the algorithm cannot converge under fault-tolerant conditions. The latter can be terminated with a relatively small loss, but the distance between the decision boundary and the correct boundary is large.

Xiao et al. [76] attacked the support vector machine by flipping the label to adjust the training set. They proposed an optimized framework to find the label flip to maximize the classification error and thus reduce the accuracy of the classifier. Biggio et al. [77] implemented a poisoning attack against single link hierarchy clustering. It relies on heuristic algorithms to find the optimal attack strategy. They use fuzzy attacks to minimize the clustering results. Alfeld et al. [9] proposed a framework for coding attacker desires and constraints under a linear autoregressive model. The attacker can push the prediction in a certain direction by adding the optimal special record to the training data. Jagielski et al. [79] discussed the poisoning attack of the linear regression model. Attackers can manipulate datasets and algorithms to influence machine learning models. They introduced a fast statistical attack, which requires only limited training process knowledge.

2) Specific obfuscated data. Machine learning extracts representative features from a large amount of information for learning and training. The weights of these features are determined by training and have important significance for prediction. However, if some well-designed data have unbiased characteristic distribution, it will destroy the training and get a set of misleading feature weights.

This method is also common in feature selection algorithms such as LASSO, RidgeRegression, and Elasticnet. Xiao et al. [56] directly studied the robustness of common feature selection algorithms under poisoning attacks. The results show that in malware detection applications, feature selection algorithms are destructively affected by poisoning attacks. By inserting less than 5% of toxic training samples, the results obtained by the LASSO feature selection process are almost indistinguishable from random selection. Shafahi et al. [80] tried to find a specific test instance to control the behavior of the classifier without controlling the label of the training data. They proposed a watermarking strategy and trained multiple instances of poisoning. Add the low transparency watermark of the target instance to the poisoned instance to allow some indivisible characteristics to overlap. This method opens the backdoor of a classifier for the attacker, and the attacker does not need to access any data collection or marking process.

D. Adversarial attacks

Adversarial attacks use adversarial examples (AEs) to make model prediction errors, also known as evasion attacks. Adversarial attack is an exploratory attack, which destroys the usability of the model. AEs are generated by adding disturbances to the original sample. They confuse well-trained models, but they are normal to humans, which guarantees the effectiveness of attacks. In image processing, small disturbances are usually used to ensure the similarity between the original samples and AEs. In speech and text, it ensures that AEs are also meaningful and context-sensitive. Malware detection ensures that AEs still have the original malicious function after adding disturbances.

The misclassification of the model has two major categories: targeted and untargeted. The former requires AEs to be incorrectly classified as specific tags to achieve special malicious purposes. The latter only requires AEs to be misclassified (which can be any wrong label) and used to resist detection or other scenarios. The generation process of AEs usually needs to minimize the disturbance, because the smaller the disturbance, the smaller the impact on people. The minimum distance is usually measured by L_p distance (or Minkowski distance), commonly used are L_0, L_1, L_2 and L_∞:

$$\begin{gathered} L_p(x, y) = \left(\sum_{i=1}^{n} \left| x^{(i)} - y^{(i)} \right|^p \right)^{\frac{1}{p}} \\ x = \left(x^{(1)}, x^{(2)}, \cdots, x^{(n)} \right) \\ y = \left(y^{(1)}, y^{(2)}, \cdots, y^{(n)} \right) \end{gathered} \tag{2}$$

Adversarial attacks can be applied in many fields, among which the most widely used is image classification. By adding small disturbances, we can generate confrontational images, which are difficult for humans to distinguish, but can cause classification errors of the model. Adversarial attacks are also used in other fields, such as audio [59], [84], text [61], malware detection [85]–[87], etc. Carlini et al. [60] proposed a text search attack system DeepSearch based on speech-to-text neural network. It can transform any given waveform into any desired target phrase by adding small disturbances. They use sequence-to-sequence neural networks to generate more than 99.9% similar waveforms and achieve an attack rate of

100%. Gao et al. [61] proposed the framework DeepWordBug to generate adversarial text sequences in black box settings. They use different scoring functions to handle better mutant words. They almost minimize the editing distance and reduce the text classification accuracy from 90% to 30% 60%. Rigaki et al. [88] used GANs to modify the network behavior to simulate the flow of legitimate applications to avoid malware detection. They can modify the source code of the malware to adjust the command and control (C2) channel to simulate Facebook chat network traffic. The best GAN model generates more than one C2 flow per minute after 300 to 400 training stages. Literature [89], [90] proposed a method of generating malware instances in a black box to carry out an attack detection model. In addition, the literature [91] proposed an anti-attack algorithm for binary-coded malware detection, achieving 91.9% accuracy.

In the field of images, adversarial attacks are mainly implemented by searching for adversarial samples through gradient descent, optimization, and neural network automation. Some studies have also begun to consider the problem of confronting samples in the real world. Here, we define$F : \mathbb{R}^n \rightarrow \{1, 2, \cdots, k\}$ as a model classifier that maps image value vectors to class labels. $Z(\cdot)$ is the output of the penultimate layer 2 and usually represents the class probability. δ is perturbation, $\|\boldsymbol{\delta}\|_i$ means seeking L_i distance. There are 12 methods of generating anti-disturbance in the image field:

1) L-BFGS attack. Szegedy et al. [92] proposed box-constrained L-BFGS for generating AEs. They also discovered two features that were counterintuitive. First, the semantic information contained in this space is located at the upper level of the neural network, rather than a single unit. Second, perturbations or AEs are relatively robust and can be shared between different neural networks or training datasets. These properties laid the foundation for future research.

2) FGSM attack. FGSM (fast gradient sign method) was proposed by Goodfellow et al. [63]. The article explains that the cause of AEs is the linear behavior of neural networks in high-dimensional space, not nonlinear. Let l_x be the actual classification of x. The loss function describes the loss of input x. The direction of the disturbance δ is determined using the gradient calculated by back propagation. The size of each pixel in the gradient direction is ε. As ε increases, the size of the disturbance and the success rate of the attack increase, and the likelihood of being discovered increases.

$$\boldsymbol{\delta} = \varepsilon \times \operatorname{sign}\left(\nabla_x L\operatorname{oss}_F\left(\boldsymbol{x}, l_x\right)\right) \tag{3}$$

3) BIM attack. BIM (basic iteration method) [93] is an iterative version of FGSM, also known as I-FGSM. The $Clip_{x,\varepsilon}(x)$ function cuts the image of each pixel, and makes the generated AE meet the boundary of L_∞ at each iteration. I-FGSM is stronger than FGSM in white box attacks, but its portability is poor [94], [95].

4) MI-FGSM attack. MI-FGSM (momentum iterative FGSM) [96] is based on gradient introduction. Momentum is used to get rid of local extremum, and iteration is used to stabilize optimization. On the white box or black box model, this method is more portable than the gradient-based single-step method.

5) JSMA attack. JSMA (Jacobian-based saliency map attack) [64] only changes a small number of pixels without affecting the entire image. It limits the L_0 distance instead of L_2 and L_∞. They modify individual pixels of the image each time, record their impact on the classification results, and then proceed iteratively.

6) C & W attack. C & W [65] realized the attack on distillation defense method [97] in L_0, L_2 and L_∞. They try to find the smallest possible δ and deceive the classifier. C& W guarantees that the generated AEs will be misclassified, however, due to the large amount of calculation, the time overhead is large.

7) EAD attack. EAD (Elastic-net attacks to DNNs) [98] is an elastic network regularization attack framework for making AEs. It combines L_1 and L_2 metrics, provides rarely used L_1-oriented examples, and sets the best L_2 attack as a special case. The results show that the L_1-based example designed by EAD performs as well as other best attacks.

8) OptMargin attack. OptMargin [99] can avoid defense based on area classification in a limited input space. Different from previous research, its goal is a low-dimensional subspace, which is not limited by neighboring points around the space. The decision boundary of AEs produced by this method is different from benign samples. However, it cannot imitate benign samples. OptMargin is an extension of C & W's L_2 attack. It adds many objective functions around x.

9) DeepFool attack. DeepFool [66] iteratively generates the smallest normalized disturbance. They gradually pushed the image into the classification boundary until the symbol changed. Under the similar successful deception rate, the disturbance generated by DeepFool is smaller than that of FGSM.

10) NewtonFool attack. NewtonFool [100] proposed a strong assumption that the attacker can use the class probability vector $Z(x)$ output from the penultimate layer 2. Assuming $l = F\left(x_0\right)$, their purpose is to find a small δ such that $Z\left(\boldsymbol{x}_0 + \boldsymbol{\delta}\right)_l = 0$. They use an iterative method to reduce $Z\left(\boldsymbol{x}_0\right)_l$ to 0 as quickly as possible. The results show that it is faster than FGSM, JSMA and DeepFool.

11) UAP attack. UAP (universal adversarial pertur-bations) [67] can lead to misclassification of the target model with high probability on almost any input data. UAP is universal for data and network architecture. Let μ denote the dataset containing all samples. Its main purpose is to find the perturbation δ, which can deceive $F(\cdot)$ on almost all samples in μ. Hayes et al. [101] used Universal Adversarial Networks (UANs) to automatically generate UAP in targeted and untargeted attacks.

12) ATN attack. ATN [23] is a well-trained neural network that can efficiently and automatically attack another target. ATN converts any input to AE by adding a minimum disturbance. They used targeted white-box ATNs to generate AEs, and successfully converted 83% 92% of the image input into an adversarial attack on ImageNet.

E. Poisoning defense

Most poisoning attacks are concentrated on data and algorithms, so the defense method mainly considers protecting data and algorithms.

1) Protect data. Data protection mainly includes protecting the collected data from tampering, resisting rewriting attacks, preventing rejection, preventing data forgery, and detecting toxic data, etc. [102]–[104]. Olufowobi et al. [105] proposed a data source model for the Internet of Things system to improve the credibility and reliability of the data. The model describes the context of creating or modifying data points. Their future work is to integrate the model into the data source integrity detection algorithm of IoT devices. Chakarov et al. [106] adopted a method for detecting poisoning data by evaluating the effect of a single data point on the performance of the training model. They need to evaluate the model by comparing the performance on the trusted dataset. Baracaldo et al. [58] detected poisoning attacks by using source information as part of the filtering algorithm. This method improves the detection rate. They use the source of the training data points and the conversion context to identify harmful data. It is implemented on partially trusted and completely untrusted datasets.

2) Protect algorithm. Learning algorithms always make trade-offs between preventing regularization and reducing loss functions. This uncertainty may lead to the vulnerability of learning algorithms. Some poisoning attacks are implemented according to their own weaknesses, so studying robust machine learning algorithms is an effective way to prevent poisoning attacks. Candès et al. [107] first studied the robust PCA robust machine learning algorithm. It assumes that a small part of the underlying dataset is destroyed randomly, rather than targeted. Chen et al. [108] studied the robust linear regression problem against damage and Feng et al. [109] studied the robust logistic regression. They all need to make strong assumptions about feature independence and sub-Gaussian distribution. Goodfellow et al. [63] proposed a robust linear regression method, which relaxes the assumption of feature independence and low-variance sub-Gaussian noise, only assuming that the feature matrix can be approximated by a low-rank matrix. This method combines robust low-rank matrix approximation with robust principal components to obtain a strong performance guarantee. Jagielski et al. [79] added toxic data to train models during the training process, instead of simply deleting them. This method estimates the regression parameters iteratively and trains it on a subset of the points with the smallest residuals in each iteration. Essentially, it uses a pruned loss function calculated from different subsets of residuals in each iteration.

F. Adversarial defense

The defense methods against attacks mainly start from the two goals of preventing confrontation sample generation and detecting confrontation samples. This article summarizes the following seven methods.

1) Adversarial training. Adversarial training selects AEs as part of the training dataset, so that the trained model can learn the characteristics of AEs. Huang et al. [68] proposed an earlier defense method, which is to learn robust classifiers with strong opponents by generating AEs as an intermediate step. They also proposed a new search method for AEs. Kurakin et al. [95] applied adversarial training to larger datasets, such as ImageNet. Its main innovations are batch normalization, training dataset and relative weights. They also found that one-step attacks are more portable than iterative attacks. But this training loses part of its accuracy on normal samples. In addition, integrated adversarial training [94] contains every input transmitted from other pre-trained models. However, adversarial training can only make the training model robust to AEs in the training set, and the model cannot learn the characteristics of AEs outside the training set.

2) Region-based classification. Understanding the nature of the adversarial sample area and using a more robust area-based classification can also resist adversarial attacks. Cao et al. [110] developed new DNNs using region-based classification (RC) instead of point-based classification. They predict a label by randomly selecting a few points from a hypercube centered on the test sample. RC reduces the success rate of C & W attacks from 100% to 16%, but it is difficult for OptMargin attacks. Pang et al. [69] used a reverse cross entropy defense method. The classifier maps normal samples to the low-dimensional manifold neighborhood of the final hidden layer space. Ma et al. [111] proposed local intrinsic dimensions to characterize the dimensional characteristics of the confrontation zone. Based on the distance distribution between the sample and the neighborhood, they evaluated the space filling ability of the sample area. In addition, Mccoyd et al. [112] added a large number of background images of different categories to the training dataset to help detect AEs. They added background classes between the key classes in the EMNIST dataset, and the background classes filled the blank areas between the key classes. This method is easy to implement, but has no effect on C & W attacks.

3) Input data transformation. Changing or transforming inputs can defend adversarial attacks. Song et al. [70] found that AEs are mainly located in the low probability area of the training area. Therefore, they designed PixelDefend to purify AE by adaptively moving AE in the distribution direction. Guo et al. [113] explored the model-independent defense of image classification systems through image conversion. Their purpose is to eliminate the input disturbance. Their image conversion includes image cropping and rescaling, bit depth reduction, JPEG compression, total variance minimization and image stitching. Xie et al. [114] used randomization of input in the prediction process to defend against attacks and mitigate the effects, including random adjustment of image size and random padding. This method has a small amount of calculation and is compatible with other defense methods. In addition, Wang et al. [115] believe that AEs are more sensitive than normal samples. If a large number of random disturbances are added to the adversarial and normal samples, the proportion of label changes will be significantly different, so that adversarial samples can be identified. They realized the

difference between high accuracy and low cost on MNIST and CIFAR-10. Tian et al. [116] believe that AEs are more sensitive to certain image transformation operations than normal images. They used this method to resist white box C & W attacks in image classification. Buckman et al. [117] proposed a simple modification method TE (thermometer encoding) for neural networks. They found that discretization of TE and hot codes significantly improved the robustness of the network to AEs.

4) Gradient regularization. Gradient regularization (or gradient masking) is another effective defense method. Madry et al. [71] achieved this by optimizing the saddle point formula, which includes an internal maximum value solved by projected gradient descent (PGD) and an external minimum value solved by random gradient descent (SGD). But they found that this cannot be guaranteed within a reasonable time. Ross et al. [118] analyzed the regularization of input gradients, the purpose of which was to train differentiable models to punish small changes in input. The results show that the input gradient regularization enhances the robustness and is qualitatively different from defensive distillation and confrontation training.

5) Defensive distillation. Papernot et al. [72] proposed a defensive distillation method. Distillation mainly refers to transferring knowledge from a complex structure to a simple structure, thereby reducing the computational complexity of the DNN structure. This method can successfully suppress FGSM and AEs constructed based on Jacobian iterative attack. Papernot et al. [1] also used knowledge extracted by defensive distillation to smooth the model and reduce the size of the network gradient. A large network gradient means that small disturbances will cause large changes in the output results, which is helpful for finding adversarial samples.

6) Data processing. Liang et al. [119] introduced scalar quantization and smooth spatial filtering to reduce the impact of disturbances. They used image entropy as a metric and adaptively denoised various images. In literature [73], the bounded ReLU activation function is used to hedge the forward propagation of disturbance against disturbance, and the Gaussian data enhancement method is used to enhance the generalization ability. Xu et al. [120] proposed a counterexample detection method based on feature compression, including reducing the depth and spatial smoothing of the color bits on each pixel.

7) Defense network. Some studies use tools such as neural networks to automatically counteract AEs. Gu et al. [74] used Deep Contractive Networks (DCN) with contractive autoencoders (CAEs) and denoising autoencoders (DAEs), which can remove a large amount of anti-noise through additional noise corrosion and preprocessing. Akhtar et al. [121] proposed a perturbation rectification network as the pre-input layer of the target model for combating UAPs. It can provide defense for the deployed network without modifying the network and resist invisible hostile interference. MagNet [122] uses a detection network to detect AEs away from the manifold boundary, and uses a reformer to transform AEs near the boundary. This process does not require knowledge of AEs or generation processes.

VI. Discussion

In the future, we can still get a safer machine learning system through the following aspects:

1) Improve data quality and enhance data security. Machine learning collects polluted data, so the collected data should be cleaned to improve the quality of the data. On the one hand, manual methods can be used to remove the polluted data; on the other hand, the techniques of cleaning and protecting the dataset in the defense method can be used. Faced with insufficient data, similar data can be obtained by constructing generative models (such as GAN). In short, the higher the quality of training data, the safer the trained model obtained.

2) Ensure the privacy of personal data and prevent the model from misusing private information. In the current machine learning system, personal data is difficult to obtain security, and the model may infer a large amount of private information from personal data. In order to protect the privacy of users, we recommend that: introduce regulatory authorities to monitor the model, strictly monitor the use of data by the model, only allow the model to extract the features within the allowable range, and cannot extract and infer sensitive information without authorization; data source protection, The data collected by the model must be de-privatized to obscure irrelevant information; establish and improve relevant laws and regulations to supervise the process of data collection, storage, use and deletion.

VII. Conclusion

This paper investigates related papers in the field of machine learning security, and makes a complete and detailed division of the security problems encountered by machine learning systems. We divide the field into two major blocks, privacy and security, and divide the attacks into four categories according to the purpose of the attack, the target of the attack, and the attack process. Within each attack, according to the timeline and the technology used, the complicated research is summarized, different technologies are divided, and the advantages and disadvantages of the technologies are compared and analyzed. In terms of defense, we focus on protecting the privacy of machine learning systems and resisting security attacks, and summarize the defense technologies within each defense type, and introduce the adaptability of defense technologies to attack technologies. In addition, based on the summary and research of these attack and defense technologies, we also propose the experience of building a safe and robust machine learning system, protecting the privacy of all participants in machine learning, and also focusing on current hot issues in machine learning systems and artificial intelligence.

References

[1] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, “The security of machine learning,” *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete problems in ai safety,” *arXiv preprint arXiv:1606.06565*, 2016.

[3] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, “Towards the science of security and privacy in machine learning,” *arXiv preprint arXiv:1611.03814*, 2016.

[4] 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.

[5] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, “Sok: Security and privacy in machine learning,” in *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, 2018, pp. 399–414.

[6] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, “A survey on security threats and defensive techniques of machine learning: A data driven view,” *IEEE Access*, vol. 6, pp. 12 103–12 117, 2018.

[7] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[8] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, “Deepsec: A uniform platform for security analysis of deep learning model,” in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 673–690.

[9] S. Alfeld, X. Zhu, and P. Barford, “Data poisoning attacks against autoregressive models,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[10] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the gan: information leakage from collaborative deep learning,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 603–618.

[11] C. Song, T. Ristenpart, and V. Shmatikov, “Machine learning models that remember too much,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 587–601.

[12] M. Veale, R. Binns, and L. Edwards, “Algorithms that remember: model inversion attacks and data protection law,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2133, p. 20180083, 2018.

[13] J. Buolamwini and T. Gebru, “Gender shades: Intersectional accuracy disparities in commercial gender classification,” in *Conference on fairness, accountability and transparency*, 2018, pp. 77–91.

[14] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[15] D. Wang, M. Ye, and J. Xu, “Differentially private empirical risk minimization revisited: Faster and more general,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2722–2731.

[16] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, “Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy,” in *International Conference on Machine Learning*, 2016, pp. 201–210.

[17] X. Jiang, M. Kim, K. Lauter, and Y. Song, “Secure outsourced matrix computation and application to neural networks,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1209–1222.

[18] P. Mohassel and Y. Zhang, “Secureml: A system for scalable privacy-preserving machine learning,” in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 19–38.

[19] S. Wagh, D. Gupta, and N. Chandran, “Securenn: Efficient and private neural network training.” *IACR Cryptology ePrint Archive*, vol. 2018, p. 442, 2018.

[20] T. Lee, B. Edwards, I. Molloy, and D. Su, “Defending against machine learning model stealing attacks using deceptive perturbations,” *arXiv preprint arXiv:1806.00054*, 2018.

[21] B. Wang and N. Z. Gong, “Stealing hyperparameters in machine learning,” in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 36–52.

[22] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction apis,” in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 601–618.

[23] S. Baluja and I. Fischer, “Learning to attack: Adversarial transformation networks.” in *AAAI*, vol. 1, 2018, p. 3.

[24] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, “Prada: protecting against dnn model stealing attacks,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 512–527.

[25] N. Papernot, P. McDaniel, and I. Goodfellow, “Transferability in machine learning: from phenomena to black-box attacks using adversarial samples,” *arXiv preprint arXiv:1605.07277*, 2016.

[26] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, “Towards demystifying membership inference attacks,” *arXiv preprint arXiv:1807.09173*, 2018.

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[28] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, “Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models,” *arXiv preprint arXiv:1806.01246*, 2018.

[29] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[30] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, “Understanding membership inferences on well-generalized learning models,” *arXiv preprint arXiv:1802.04889*, 2018.

[31] K. S. Liu, B. Li, and J. Gao, “Generative model: Membership attack, generalization and diversity,” *CoRR, abs/1805.09898*, 2018.

[32] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, “Logan: Evaluating privacy leakage of generative models using generative adversarial networks,” 05 2017.

[33] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, “Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers,” *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, 2015.

[34] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, “Property inference attacks on fully connected neural networks using permutation invariant representations,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 619–633.

[35] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, “Inference attacks against collaborative learning,” *arXiv preprint arXiv:1805.04049*, 2018.

[36] K. Chaudhuri and C. Monteleoni, “Privacy-preserving logistic regression,” in *Advances in neural information processing systems*, 2009, pp. 289–296.

[37] J. Zhang, K. Zheng, W. Mou, and L. Wang, “Efficient private erm for smooth objectives,” *arXiv preprint arXiv:1703.09947*, 2017.

[38] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially private empirical risk minimization,” *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.

[39] D. Kifer, A. Smith, and A. Thakurta, “Private convex empirical risk minimization and high-dimensional regression,” in *Conference on Learning Theory*, 2012, pp. 25–1.

[40] K. Talwar, A. Thakurta, and L. Zhang, “Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry,” *arXiv preprint arXiv:1411.5417*, 2014.

[41] S. Song, K. Chaudhuri, and A. D. Sarwate, “Stochastic gradient descent with differentially private updates,” in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 245–248.

[42] R. Bassily, A. Smith, and A. Thakurta, “Private empirical risk minimization: Efficient algorithms and tight error bounds,” in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 2014, pp. 464–473.

[43] T. Zhang and Q. Zhu, “A dual perturbation approach for differential private admm-based distributed empirical risk minimization,” in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, 2016, pp. 129–137.

[44] J. Hamm, Y. Cao, and M. Belkin, “Learning privately from multiparty data,” in *International Conference on Machine Learning*, 2016, pp. 555–563.

[45] N. Hynes, R. Cheng, and D. Song, “Efficient deep learning on multi-source private data,” *arXiv preprint arXiv:1807.06689*, 2018.

[46] F. D. McSherry, “Privacy integrated queries: an extensible platform for privacy-preserving data analysis,” in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.

[47] J. Liu, M. Juuti, Y. Lu, and N. Asokan, “Oblivious neural network predictions via minionn transformations,” in *Proceedings of the 2017*

ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 619–631.

[48] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[49] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: Deep neural networks over encrypted data," *arXiv preprint arXiv:1711.05189*, 2017.

[50] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[51] L. Phong and T. Phuong, "Privacy-preserving deep learning for any activation function," *CoRR, vol. abs/1809.03272*, 2018.

[52] M. Kesarwani, B. Mukhoty, V. Arya, and S. Mehta, "Model extraction warning in mlaas paradigm," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 371–380.

[53] M. Brückner and T. Scheffer, "Nash equilibria of static prediction games," in *Advances in neural information processing systems*, 2009, pp. 171–179.

[54] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. A. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter." *LEET*, vol. 8, pp. 1–9, 2008.

[55] S. Mei and X. Zhu, "The security of latent dirichlet allocation," in *Artificial Intelligence and Statistics*, 2015, pp. 681–689.

[56] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?" in *International Conference on Machine Learning*, 2015, pp. 1689–1698.

[57] C. Liu, B. Li, Y. Vorobeychik, and A. Oprea, "Robust linear regression against training data poisoning," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 91–102.

[58] N. Baracaldo, B. Chen, H. Ludwig, and J. A. Safavi, "Mitigating poisoning attacks on machine learning models: A data provenance based approach," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 103–110.

[59] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 49–64.

[60] W. CarliniN, "Audioadversarialexamples: Targeted attackson speechgtogtext," in *Proc ofthe*, 2018.

[61] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 50–56.

[62] F. Kreuk, A. Barak, S. Aviv-Reuven, M. Baruch, B. Pinkas, and J. Keshet, "Deceiving end-to-end deep learning malware detectors using adversarial examples," *arXiv preprint arXiv:1802.04528*, 2018.

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

[64] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[65] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[66] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[67] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[68] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[69] T. Pang, C. Du, and J. Zhu, "Robust deep learning via reverse cross-entropy training and thresholding test," *arXiv preprint arXiv:1706.00633*, vol. 3, 2017.

[70] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv preprint arXiv:1710.10766*, 2017.

[71] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[72] N. Papernot and P. McDaniel, "On the effectiveness of defensive distillation," *arXiv preprint arXiv:1607.05113*, 2016.

[73] V. Zantedeschi, M.-I. Nicolae, and A. Rawat, "Efficient defenses against adversarial attacks," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 39–49.

[74] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv preprint arXiv:1412.5068*, 2014.

[75] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[76] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines." in *ECAI*, 2012, pp. 870–875.

[77] B. Biggio, I. Pillai, S. Rota Bulò, D. Ariu, M. Pelillo, and F. Roli, "Is data clustering in adversarial settings secure?" in *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, 2013, pp. 87–98.

[78] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[79] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 19–35.

[80] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 6103–6113.

[81] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Advances in neural information processing systems*, 2017, pp. 3517–3529.

[82] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 27–38.

[83] Y. Wang and K. Chaudhuri, "Data poisoning attacks against online learning," *arXiv preprint arXiv:1808.08994*, 2018.

[84] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," *arXiv preprint arXiv:1711.03280*, 2017.

[85] W. Huang and J. W. Stokes, "Mtnet: a multi-task neural network for dynamic malware classification," in *International conference on detection of intrusions and malware, and vulnerability assessment*. Springer, 2016, pp. 399–418.

[86] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 49–54.

[87] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1916–1920.

[88] M. Rigaki and S. Garcia, "Bringing a gan to a knife-fight: Adapting malware communication to avoid detection," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 70–75.

[89] W. Hu and Y. Tan, "Black-box attacks against rnn based malware detection algorithms," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[90] ——, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.

[91] A. Al-Dujaili, A. Huang, E. Hemberg, and U.-M. O'Reilly, "Adversarial deep learning for robust detection of binary encoded malware," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 76–82.

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

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

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

[95] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[96] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE*

conference on computer vision and pattern recognition, 2018, pp. 9185–9193.

[97] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[98] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[99] W. He, B. Li, and D. Song, "Decision boundary analysis of adversarial examples," 2018.

[100] U. Jang, X. Wu, and S. Jha, "Objective metrics and gradient descent algorithms for adversarial examples in machine learning," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 262–277.

[101] J. Hayes and G. Danezis, "Learning universal adversarial perturbations with generative models," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 43–49.

[102] X. Wang, K. Zeng, K. Govindan, and P. Mohapatra, "Chaining for securing data provenance in distributed information networks," in *MILCOM 2012-2012 IEEE Military Communications Conference*. IEEE, 2012, pp. 1–6.

[103] J. Lyle and A. Martin, "Trusted computing and provenance: better together." Usenix, 2010.

[104] R. Hasan, R. Sion, and M. Winslett, "The case of the fake picasso: Preventing history forgery with secure provenance."

[105] H. Olufowobi, R. Engel, N. Baracaldo, L. A. D. Bathen, S. Tata, and H. Ludwig, "Data provenance model for internet of things (iot) systems," in *International Conference on Service-Oriented Computing*. Springer, 2016, pp. 85–91.

[106] A. Chakarov, A. Nori, S. Rajamani, S. Sen, and D. Vijaykeerthy, "Debugging machine learning tasks," *arXiv preprint arXiv:1603.07292*, 2016.

[107] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[108] Y. Chen, C. Caramanis, and S. Mannor, "Robust high dimensional sparse regression and matching pursuit," *arXiv preprint arXiv:1301.2725*, 2013.

[109] J. Feng, H. Xu, S. Mannor, and S. Yan, "Robust logistic regression and classification," in *Advances in neural information processing systems*, 2014, pp. 253–261.

[110] X. Cao and N. Z. Gong, "Mitigating evasion attacks to deep neural networks via region-based classification," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 278–287.

[111] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," *arXiv preprint arXiv:1801.02613*, 2018.

[112] M. McCoyd and D. Wagner, "Background class defense against adversarial examples," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 96–102.

[113] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.

[114] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.

[115] J. Wang, J. Sun, P. Zhang, and X. Wang, "Detecting adversarial samples for deep neural networks through mutation testing," *arXiv preprint arXiv:1805.05010*, 2018.

[116] S. Tian, G. Yang, and Y. Cai, "Detecting adversarial examples through image transformation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[117] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," 2018.

[118] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[119] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[120] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[121] N. Akhtar, J. Liu, and A. Mian, "Defense against universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3389–3398.

[122] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 135–147.