Towards Security Threats of Deep Learning Systems: A Survey

Yingzhe He[1,2], Guozhu Meng[1,2], Kai Chen[1,2], Xingbo Hu[1,2], Jinwen He[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, China
[2]School of Cybersecurity, University of Chinese Academy of Sciences

Abstract—Deep learning has gained tremendous success and great popularity in the past few years. However, deep learning systems are suffering several inherent weaknesses, which can threaten the security of learning models. Deep learning's wide use further magnifies the impact and consequences. To this end, lots of research has been conducted with the purpose of exhaustively identifying intrinsic weaknesses and subsequently proposing feasible mitigation. Yet few are clear about how these weaknesses are incurred and how effective these attack approaches are in assaulting deep learning. In order to unveil the security weaknesses and aid in the development of a robust deep learning system, we undertake an investigation on attacks towards deep learning, and analyze these attacks to conclude some findings in multiple views. In particular, we focus on four types of attacks associated with security threats of deep learning: model extraction attack, model inversion attack, poisoning attack and adversarial attack. For each type of attack, we construct its essential workflow as well as adversary capabilities and attack goals. Pivot metrics are devised for comparing the attack approaches, by which we perform quantitative and qualitative analyses. From the analysis, we have identified significant and indispensable factors in an attack vector, *e.g.*, how to reduce queries to target models, what distance should be used for measuring perturbation. We shed light on 18 findings covering these approaches' merits and demerits, success probability, deployment complexity and prospects. Moreover, we discuss other potential security weaknesses and possible mitigation which can inspire relevant research in this area.

Index Terms—deep learning, poisoning attack, adversarial attack, model extraction attack, model inversion attack

1 Introduction

Deep learning has gained tremendous success and is the most significant driving force for artificial intelligence (AI). It fuels multiple areas including image classification, speech recognition, natural language processing, and malware detection. Due to the great advances in computing power and the dramatic increase in data volume, deep learning has exhibited superior potential in these scenarios, compared to traditional techniques. Deep learning excels in feature learning, deepening the understanding of one object, and unparalleled prediction ability. In image recognition, convolutional neural networks (CNNs) can classify different unknown images for us, and some even perform better than humans. In natural language processing, recurrent neural networks (RNNs) or long-short-term memory networks (LSTMs) can help us translate and summarize text information. Other fields including autonomous driving, speech recognition, and malware detection all have widespread application of deep learning. The Internet of things (IoT) and intelligent home systems have also arisen in recent years. As such, we are stepping into the era of intelligence.

However, deep learning-based intelligent systems around us are suffering from a number of security problems. Machine learning models could be stolen through APIs [220]. Intelligent voice systems may execute unexpected commands [262]. 3D-printing objects could fool real-world image classifiers [20]. Moreover, to ensure safety, technologies such as autonomous driving need lots of security testing before it can be widely used [217] [271]. In the past few years, the security of deep learning has drawn the attention of many relevant researchers and practitioners. They are exploring and studying the potential attacks as well as corresponding defense techniques against deep learning systems (DLS). Szegedy *et al.* [213] pioneered exploring the stability of neural networks, and uncovered their fragile properties in front of *imperceptible perturbations*. Since then, adversarial attacks have swiftly grown into a buzzing term in both artificial intelligence and security. Many efforts have been dedicated to disclosing the vulnerabilities in varying deep learning models (e.g., CNN [175] [156] [155], LSTM [67] [44] [177], reinforcement learning (RL) [94], generative adversarial network (GAN) [114] [190]), and meanwhile testing the safety and robustness for DLS [113] [146] [170] [211] [79] [250]. On the other hand, the wide commercial deployment of DLS raises interest in proprietary asset protection such as the training data [162] [181] [272] [10] and model parameters [107] [122] [92] [111]. It has started a war where privacy hunters exert corporate espionage to collect privacy from their rivals and the corresponding defenders conduct extensive measures to counteract the attacks.

Prior works have been conducted to survey security and privacy issues in machine learning and deep learning [13] [26] [175] [22]. They enumerate and analyze attacks as well as defenses that are relevant to both the training phase and prediction phase. However, these works mainly evaluate the attacks either in limited domains (*e.g.*, computer vision) or perspectives (*e.g.*, adversarial attack). Few studies can provide a systematical evaluation of these attacks in their entire life cycles, which include the general workflow, adversary model, and comprehensive comparisons between different approaches. This knowledge can help demystify

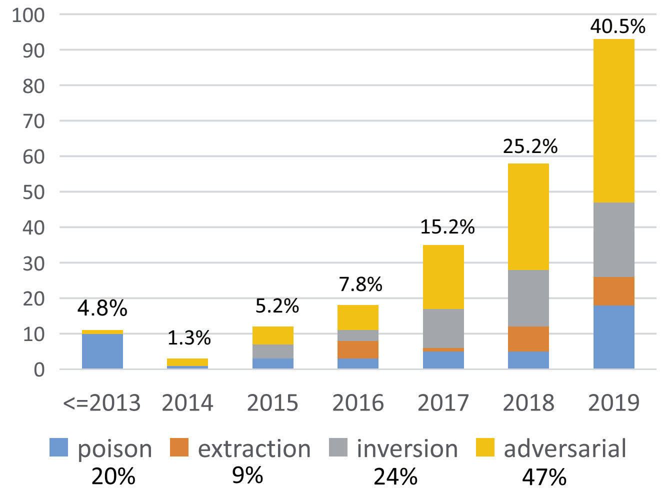

Fig. 1: Publications we surveyed of four attacks and corresponding defenses in deep learning. The X-axis represents the year, and the Y-axis represents the corresponding number of publications for every year.

how these attacks happen, what capabilities the attackers possess, and both salient and tiny differences in attack effects. This motivates us to explore a variety of characteristics for the attacks against deep learning. In particular, we aim to dissect attacks in a stepwise manner (*i.e.*, how the attacks are carried on progressively), identify the diverse capabilities of attackers, evaluate these attacks in terms of deliberate metrics, and distill insights for future research. This study is deemed to benefit the community threefold: 1) it presents a fine-grained description of attack vectors for defenders from which they can undertake cost-effective measures to enhance the security of the target model. 2) the evaluation on these attacks can unveil some significant properties such as success rate, capabilities. 3) the insights concluded from the survey can inspire researchers to explore new solutions.

Our Approach. To gain a comprehensive understanding of privacy and security issues in deep learning, we conduct extensive investigations on the relevant literature and systems. In total, 245 publications have been studied which are mainly spanning across four prevailing areas–image classification, speech recognition, natural language processing and malware detection. Overall, we summarize these attacks into four classes: *model extraction attack*, *model inversion attack*, *data poisoning attack*, and *adversarial attack*. In particular, model extraction and inversion attacks are targeting privacy (cf. Section 4,5), and data poisoning and adversarial attacks can influence prediction results by either downgrading the formation of deep learning models or creating imperceptible perturbations that can deceive the model (cf. Section 6,7). Figure 1 shows the publications we surveyed on these attacks in the past years. We collect papers from authoritative international venues, including artificial intelligence community, such as ICML, CVPR, NIPS, ICCV, ICLR, AAAI, IJCAI, ACL, and security community, such as IEEE S&P, CCS, USENIX Security, NDSS, TIFS, TDSC, Euro S&P, Asia CCS, RAID, and software engineering community, such as TSE, ASE, FSE, ICSE, ISSTA. We choose some keywords in the search process, including “security”, “attack”, “defense”, “privacy”, “adversarial”, “poison”, “inversion”, “inference”, “membership”, “backdoor”, “extract”, “steal”, “protect”, “detect”, and their variants. We also pay attention to the topics related to machine learning security in these venues. Furthermore, we also survey papers which cite or are cited by the foregoing papers, and include them if they have high citations. The number of related publications is experiencing a drastic increase in the past years. In our research, it gains 94% increase in 2017, 66% increase in 2018, and 61% increase in 2019. Adversarial attack is obviously the most intriguing research and occupies around 47% of researchers’ attention based on the papers we collected. It is also worth mentioning that there is an ever-increasing interest in model inversion attack recently, which is largely credited to the laborious processing of training data (More discussions can be found in Section 8).

In this study, we first introduce the background of deep learning, and summarize relevant risks and commercial DLS deployed in the cloud for public. For each type of attacks, we systematically study its capabilities, workflow and attack targets. More specifically, if one attacker is confronting a commercial deep learning system, what action it can perform in order to achieve the target, how the system is subverted step by step in the investigated approaches, and what influences the attack will make to both users and the system owner. In addition, we develop a number of metrics to evaluate these approaches such as *reducing query* strategies, *precision* of recovered training data, and *distance* with perturbed images. Based on a quantitative or qualitative analysis, we conclude many insights covering the popularity of specific attack techniques, merits and demerits of these approaches, future trends and so forth.

Takeaways. According to our investigation, we have drawn a number of insightful findings for future research. In black-box settings, attackers usually interact by querying certain inputs from the target DLS. How to reduce the number of queries for avoiding the security detection is a significant consideration for attackers (cf. Section 4). The substitute model can be a prerequisite for attacks, because of its similar behavior and transferability. Model extraction, model inversion and adversarial attacks can all benefit from it (cf. Section 4). Data synthesis is a common practice to represent similar training data. Either generated by the distribution or GAN, synthesized data can provide sufficient samples for training a substitute model (cf. Section 5). A more advanced way for poisoning purposes is to implant a backdoor in data and then attackers can manipulate the prediction results with crafted input (cf. Section 6). Most adversarial attacks have focused their main efforts on maximizing prediction errors but minimizing “distance”. However, “distance” can be measured in varying fashions and still need to be improved for better estimations and new applications (cf. Section 7). Moreover, we have discussed more security issues for modern DLS in Section 8, such as ethical considerations, system security, physical attacks and interpretability. We have investigated some works on deep learning defenses and summarized them in terms of attacks (cf. Section 8.6).

Contributions. We make the following contributions.

- **Systematic security analysis of deep learning.** We summarize 4 types of attacks. For each attack, we construct their attack vectors and pivot properties, *i.e.*, workflow,

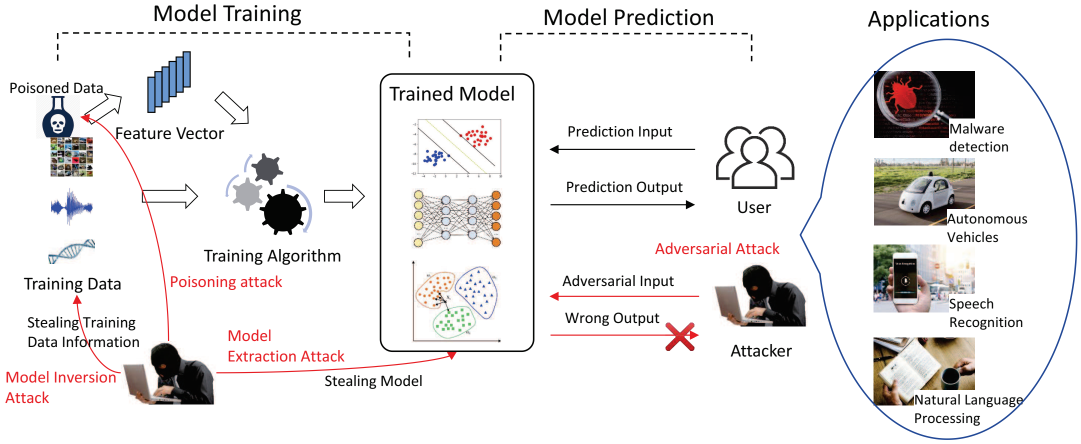

Fig. 2: Deep learning systems and the encountered attacks

adversary model (it contains attacker's capabilities and limitations), and attack goal. This could ease the understanding of how these attacks are executed and facilitate the development of counter measures.

- **Quantitative and qualitative analysis**. We develop a number of metrics that are pertinent to each type of attacks, for a better assessment of different approaches. These metrics also serve as highlights in the development of attack approaches that facilitate more robust attacks.
- **New findings.** Based on the analysis, we have concluded 18 findings that span the four attacks, and uncover implicit properties for these attack methods. Our findings summarize some results and analyze phenomena based on existing surveyed work, and predict the possible future direction of the field based on the summary results. All findings include quantitative or qualitative analysis. Beyond these attacks, we have discussed other related security problems in Section 8 such as secure implementation, interpretability, discrimination and defense techniques.

2 Related Work

There is a line of works that survey and evaluate attacks toward machine learning or deep learning.

Barreno *et al.* conduct a survey of machine learning security and present a taxonomy of attacks against machine learning systems [26]. They experiment on a popular statistical spam filter to illustrate their effectiveness. Attacks are dissected in terms of three dimensions, including workable manners, influence to input and generality. Amodei *et al.* [17] introduce five possible research problems related to accident risk and discuss probable approaches, with an example of how a cleaning robot works. Papernot *et al.* [176] study the security and privacy of machine learning. They summarize some attack and defense methods, and propose a threat model for machine learning. It introduces attack methods in training and inferring process, black-box and white-box model. However, they do not include much information about defenses or the most widely used deep learning models.

Bae *et al.* [22] review the attack and defense methods under security and privacy AI concept. They inspect evasion and poisoning attacks, in black-box and white-box. In addition, their study focuses on privacy with no mention of other attack types.

Liu *et al.* [138] aim to provide a literature review in two phases of machine learning, *i.e.*, the training phase and the testing/inferring phase. As for the corresponding defenses, they sum up with four categories. In addition, this survey focuses more on data distribution drifting caused by adversarial samples and sensitive information violation problems in statistical machine learning algorithms.

Akhtar *et al.* [13] conduct a study on adversarial attacks of deep learning in computer vision. They summarize 12 attack methods for classification, and study attacks on models or algorithms such as autoencoders, generative models, RNNs and so on. They also study attacks in the real world and summarize defenses. However, they only research the computer vision part of adversarial attack.

Huang *et al.* [96] research the safety and trustworthiness on the deployment of DNNs. They address the trustworthiness within a certification process and an explanation process. In certification, they study DNN verification and testing techniques, and in explanation, they consider DNN interpretability problems. Adversarial attack and defense techniques go through the whole procedure. Different from us, their security considerations pay more attention to ensure trustworthiness during the DNN deployment process.

Zhang *et al.* [270] summarize and analyze machine learning testing techniques. Testing can expose problems and improve the trustworthiness of machine learning systems. Their survey covers testing properties (such as correctness, robustness, fairness), testing components (such as data, learning program, framework), testing workflow (such as test generation, test evaluation), and application scenarios (such as autonomous driving, machine translation). Unlike us, their focus on safety is from a testing perspective.

3 Overview

3.1 Deep Learning System

Deep learning is inspired by biological nervous systems and is composed of thousands of neurons to transfer information. Figure 2 demonstrates a classic deep learning process. Typically, it exhibits to the public an overall process

TABLE 1: Notations used in this paper

Notation	Explanation
D	dataset
$x = \{x^1, \ldots, x^n\}$	inputs in D
$y = \{y^1, \ldots, y^n\}$	predicted labels of x
$y_t = \{y_t^1, \ldots, y_t^n\}$	true labels of x
$\|\|x - y\|\|^2$	the $Euclidean$ distance for x and y
F	model function
Z	output of second-to-last layer
$\mathcal{L}$	loss function
w	weights of parameters
b	bias of parameters
λ	hyperparameters
L_p	distance measurement
δ	perturbation to input x

including: 1) *Model Training*, where it converts a large volume of data into a trained model, and 2) *Model Prediction*, where the model can be used for prediction as per input data. Prediction tasks are widely used in different fields. For instance, image classification, speech recognition, natural language processing and malware detection are all pertinent applications for deep learning.

To formalize the process of deep learning systems, we present some notations in Table 1. Given a learning task, the training data can be represented as $(x, y_t) \in D$. Let F be the deep learning model and it computes the corresponding outcomes y based on the given input x, *i.e.*, $y = F(x)$. y_t is the true label of input x. Within the course of model training, there is a loss function $\mathcal{L}$ to measure the prediction error between predicted result and true label, and the training process intends to gain a minimal error value via fine-tuning parameters. There exist many loss functions to measure the differences. One commonly used loss function can be computed as $\mathcal{L} = \Sigma_{1 \leqslant i \leqslant n} ||y_t^i - y^i||^2$. So the process of model training can be formalized as [183]:

$$\arg\min_{F} \sum_{1 \leqslant i \leqslant n} ||y_t^i - y^i||^2 \tag{1}$$

3.2 Risks in Deep Learning

One deep learning system involves several pivotal assets that are confidential and significant for the owner. As per the phases in Figure 2, risks stem from three types of concerned assets in deep learning systems: 1) training dataset. 2) trained model including model structures, and model parameters. 3) inputs and results of predictions.

❶ Training dataset. High-quality training data is significant and vital for a better performance of the deep learning model. As a deep learning system has to absorb plenty of data to form a qualified model, mislabelled or inferior data can hinder this formation and affect the model's quality. These kinds of data can be intentionally appended to the benign data by attackers, which is referred to as *poisoning attack* (cf. Section 6). On the other hand, the collection of training data takes lots of human resources and time costs. Industry giants such as Google have far more data than other companies. They are more inclined to share their state-of-the-art algorithms [106] [55], but they barely share data. Therefore, training data is crucial and considerably valuable for a company, and its leakage means big loss of assets. However, recent research found there is an inverse flow from prediction results to training data [221]. It leads that one attacker can infer out the confidential information in training data, merely relying on authorized access to the victim system. It is literally noted as *model inversion attack* whose goal is to uncover the composition of the training data or its specific properties (cf. Section 5).

❷ Trained model. The trained model is an abstract representation of its training data. Modern deep learning systems have to cope with a large volume of data in the training phase, which has a rigorous demand for high performance computing and mass storage. Therefore, the trained model is regarded as the core competitiveness for a deep learning system, endowed with commercial value and creative achievements. Once it is cloned, leaked or extracted, the interests of model owners will be seriously damaged. More specifically, attackers have started to steal model parameters [220], functionality [167] or decision boundaries [174], which are collectively known as *model extraction attack* (cf. Section 4).

❸ Inputs and results of predictions. As for prediction data and results, curious service providers may retain user's prediction data and results to extract sensitive information. These data may also be attacked by miscreants who intend to utilize these data to make their own profits. On the other hand, attackers may submit carefully modified input to fool models, which is dubbed *adversarial example* [213]. An adversarial example is crafted by inserting slight perturbations into the original normal sample which are not easy to perceive. This is recognized as *adversarial attack* or *evasion attack* (cf. Section 7).

3.3 Commercial Off-The-Shelf

Machine learning as a Service (MLaaS) has gained momentum in recent years [130], and lets its clients benefit from machine learning without establishing their own predictive models. To ease the usage, the MLaaS suppliers make a number of APIs for clients to accomplish machine learning tasks, *e.g.*, classifying an image, recognizing a slice of audio or identifying the intent of a passage. Certainly, these services are the core competence which also charge clients for their queries. Table 2 shows representative COTS as well as their functionalities, outputs to the clients, and usage charges. Taking Amazon Image Recognition for example, it can recognize the person in a profile photo and tell his/her gender, age range, emotions. Amazon charges this service at 1,300 USD per one million queries.

4 Model Extraction Attack

4.1 Introduction of Model Extraction Attack

Model extraction attack attempts to duplicate a machine learning model through the provided APIs, without prior knowledge of training data and algorithms [220]. To formalize, given a specifically selected input x, one attacker queries the target model $\mathcal{F}$ and obtains the corresponding prediction results y. Then the attacker can infer or even extract the entire in-use model $\mathcal{F}$. With regard to an artificial neural network $y = wx + b$, model extraction attack can somehow approximate the values of w and b. Model extraction attacks cannot only destroy the confidentiality

TABLE 2: Commercial MLaaS systems and the provided functionalities, output for clients and charges per 1M queries

System	Functionality	Output	Cost/M-times
Alibaba Image Recognition	Image marking	label, confidence	2500 CNY
	scene recognition	label, confidence	1500 CNY
	porn identification	label, suggestion	1620 CNY
Amazon Image Recognition	Object & Scene Recognition	label, boundingbox, confidence	1300 USD
	face recognition	AgeRange, boundingbox, emotions, eyeglasses, gender, pose, etc	1300 USD
Google Vision API	label description	description, score	1500 USD

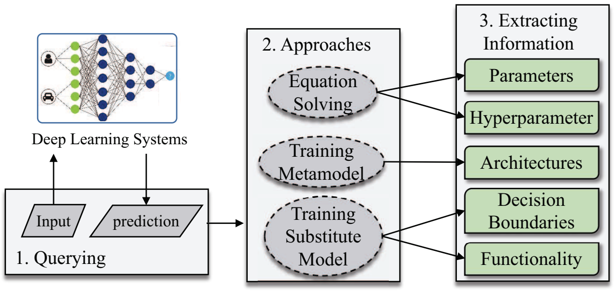

Fig. 3: Workflow of model extraction attack

of a model, and damage the interests of its owners, but also construct a near-equivalent white-box model for further attacks such as adversarial attack [174].

Adversary Model. This attack is mostly carried out under a black-box setting and attackers only have access to prediction APIs. The attacker can use an input sample to query the target model, and obtain the output including both predicted label and class probability vector. Their capabilities are limited in three ways: model knowledge, dataset access, and query frequency. Attackers have no idea about model architectures, hyperparameters, training process of the victim's model. They cannot obtain natural data with the same distribution of the target model training data. In addition, attackers may be blocked by API if submitting queries too frequently.

Workflow. Figure 3 shows a typical workflow of this attack. First, attackers submit inputs to the target model and get prediction values. Then they use input-output pairs and different approaches to extract the confidential data. More specifically, confidential data includes parameters [220], hyperparameters [226], architectures [164], decision boundaries [174] [107], and functionality [167] [54].

4.2 Approaches for Extracting Models

There are basically three types of approaches to extract models:

- **Equation Solving (ES)**. For a classification model computing class probabilities as a continuous function, it can be denoted as $F(x) = \sigma(w \cdot x + b)$ [220]. Hence, given sufficient samples $(x, F(x))$, attackers can recover the parameters (*e.g.*, w, b) by solving the equation $w \cdot x + b = \sigma^{-1}(F(x))$.
- **Training Metamodel (MM)**. Metamodel is a classifier for classification models [164]. By querying a classification model on the outputs y for certain inputs x, attackers train a meta-model F^m, mapping y to x, *i.e.*, $x = F^m(y)$. The trained model can further predict model attributes from the query outputs y.
- **Training Substitute Model (SM)**. Substitute model is a simulative model mimicking behaviors of the original model. With sufficient querying inputs x and corresponding outputs y, attackers train the model F^s where $y = F^s(x)$. As a result, the attributes of the substitute model can be near-equivalent to those of the original.

Stealing different information corresponds to different methods. In terms of time, equation solving is earlier than training meta- and substitute models. It can restore precise parameters but is only suitable for small scale models. Due to the increase of model size, it is common to train a substitute model to simulate the original model's decision boundaries or classification functionalities. However, precise parameters seem less important. Metamodel [164] is an inverse training with substitute model, as it takes the query outputs as input and predicts the query inputs as well as model attributes. Besides, it can be also used to explore more informative inputs that help infer more internal information of the model.

4.3 Different Extracted Information

4.3.1 Model Parameters & Hyperparameters

Parameters are variables that the model can learn automatically from the data, such as weights and bias. Hyperparameters are specific parameters whose values are set before the training process, including dropout rate, learning rate, mini-batch size, parameters in objective functions to balance loss function and regularization terms, and so on. In the early work, Tramèr *et al.* [220] tried equation solving to recover parameters in machine learning models, such as logistic regression, SVM, and MLP. They built equations about the model by querying APIs, and obtained parameters by solving equations. However, it needs plenty of queries and is not applicable to DNN. Wang *et al.* [226] tried to steal hyperparameter-λ on the premise of known model algorithm and training data. λ is used to balance loss functions and regularization terms. They assumed that the gradient of the objective function is $\vec{0}$ and thus got many linear equations through many queries. They estimated the hyperparameters through linear least square method.

4.3.2 Model Architectures

Architectural details include the number of layers in the model, the number of neurons in each layer, how are they connected, what activation functions are used, and so on. Recent papers usually train classifiers to predict attributes. Joon *et al.* [164] trained metamodel, a supervised classifier of classifiers, to steal model attributes (architecture, operation

TABLE 3: Evaluation on model extraction attacks as per stolen information. We sort them by the stolen "Information", corresponding to Section 4.3. "Approach" is the attack method, corresponding to Section 4.2. "Reducing Query" is the technique for reducing query number in this attack. "Recovery Rate" is the accuracy of extracted information. "SVM" is support vector machine. "DT" is decision tree. "LR" is logistic regression. "kNN" is K-nearest neighbor. "Queries" is the number of required queries for an attack.

Information	Paper	Approach	Reducing Query	Recovery Rate (%) for Models						Queries
				SVM	DT	LR	kNN	CNN	DNN	
Parameter	Tramer *et al.* [220]	ES	-	99	99	99	-	-	90	108,200
Hyperparameter	Wang *et al.* [226]	ES	-	99	-	99	-	-	-	200
Architecture	Joon *et al.* [164]	MM	KENNEN-IO	-	-	-	-	-	88	500
Decision Boundary	Papernot *et al.* [174]	SM	reservoir sampling [223]	-	-	-	-	-	84	800
	Papernot *et al.* [173]	SM	reservoir sampling [223]	83	61	89	85	-	89	800
	PRADA [107]	SM	-	-	-	-	-	-	91	300
Functionality	Silva *et al.* [54]	SM	-	-	-	-	-	98	-	-
	Orekondy *et al.* [167]	SM	random, adaptive sampling	-	-	-	-	98	-	60,000

time, and training data size). They submitted query inputs via APIs, and took corresponding outputs as inputs of metamodel, then trained metamodel to predict model attributes as outputs.

4.3.3 Model Decision Boundaries

Decision boundaries are the classification boundary between different classes. They are important for generating adversarial examples. In [174] [107] [173], they steal decision boundaries and generate transferable adversarial samples to attack a black box model. Papernot *et al.* [174] used Jacobian-based Dataset Augmentation (JbDA) to produce synthetic samples, which moved to the nearest boundary between the current class and all other classes. This technology aims not to maximize the accuracy of substitute models, but ensures that samples arrive at decision boundaries with small queries. Juuti *et al.* [107] extended JbDA to Jb-topk, where samples move to the nearest k boundaries between current class and any other class. They produced transferable targeted adversarial samples rather than untargeted [174]. In terms of model knowledge, Papernot *et al.* [173] found that model architecture knowledge was unnecessary because a simple model could be extracted by a more complex model, such as a DNN.

4.3.4 Model Functionalities

Similar functionalities refer to replicating the original model as much as possible on prediction results. The primary goal is to construct a predictive model that has closest input-output pairs with the original. In [167] [54], they try to improve the classification accuracy of a substitute model. Silva *et al.* [54] used a problem domain dataset, non-problem domain dataset, and their mixture to train a model respectively. They found the model trained with a non-problem domain dataset also did well in accuracy. Besides, Orekondy *et al.* [167] assumed attackers had no semantic knowledge over model outputs. They chose very large datasets and selected suitable samples one by one to query the black-box model. A reinforcement learning approach was introduced to improve query efficiency and reduce query counts.

4.4 Analysis of Model Extraction Attack

Model extraction attack is an emerging field of attack. In this study, we survey 8 related papers and classify them by extracted information as shown in Table 3. Based on the statistics, we draw the following conclusions.

Finding 1. *Training the substitute model (SM) is the dominant method in model extraction attacks with manifold advantages.*

The ES approach needs more than 100 thousand queries to attack a DNN model, while the SM method only needs hundreds of queries, and it can attack a more complex CNN network. Equation solving is deemed as an efficient way to recover parameters [220] or hyperparameters [226] in linear algorithms, since it has an upper bound for sufficient queries. However, the ES approach is hardly applicable to the non-linear deep learning models. Attacking DNN requires a huge amount of queries (108,200 in [220]). So researchers turn to the compelling training-based approach. For instance, [164] trains a classifier based on a target model, dubbed as metamodel, to predict structure information. This approach cannot cope with complex model attributes such as decision boundary and functionality. That drives the prevalence of substitute models (SM) which serve as an incarnation of the target model which behaves quite similarly. As such, the substitute model has approximated attributes and prediction results. The SM approach only needs 300 queries to attack DNN in [107]. For a more complex CNN, SM needs 60,000 queries in [167]. This shows that attacking more complex models requires more queries. Besides, it can be further used to steal model's training data [107] and generating adversarial examples [173].

Finding 2. *Reducing queries, which can save monetary costs for a pay-per-query MLaaS commercial system and also be resistant to attack detection, has become an intriguing research direction in recent years.*

The requirement of query reduction arises due to the high expense of queries and query amount limitation. In our investigated papers, [164] trains a metamodel–KENNEN-IO for optimizing the query inputs. [174], leverage *reservoir sampling* to select representative samples for querying, and [167] proposes two sampling strategies, *i.e.*, *random* and *adaptive* to reduce queries. Moreover, active learning [126], natural evolutionary strategies [99], optimization-based approaches [50] [193] have been adopted for query reduction.

Finding 3. *Model extraction attack is evolving from a puzzle solving game to a simulation game with cost-profit tradeoffs.*

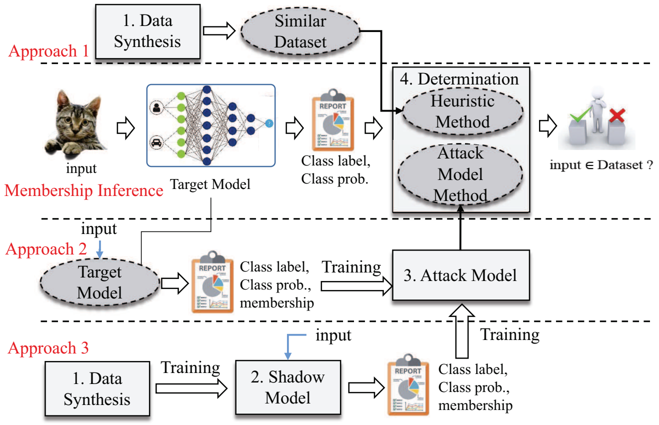

Fig. 4: Workflow of model inversion attack

MLaaS magnates like Amazon and Google have a tremendous scale of networks running behind services. It costs much to infer how many layers or neurons are in the neural networks. Therefore, it makes a remarkable dent in attackers' interest of solving model attributes. On the other hand, inferring decision boundary and model functionality emerge as new circumvention. Treating the target model as a black box, attackers observe the response by feeding it with crafted inputs, and finally construct a close approximation. Although the substitute model is likely simpler and underperforms in some cases, its prediction capabilities still make profits for attackers.

5 Model Inversion Attack

5.1 Introduction of Model Inversion Attack

In a typical model training process, lots of information is extracted and abstracted from the training data to the product model. However, there also exists one inverse information flow which allows attackers to infer the training data from the model since neural networks may remember too much information of the training data [205]. Model inversion attack leverages this information flow and restores data memberships or data properties, such as faces in face recognition systems through model prediction or its confidence coefficient. Model inversion can also be used to form physical watermarking to detect a replay attack [192].

Additionally, model inversion attack can be further refined into *membership inference attack (MIA)* and *property inference attack (PIA)*. We distinguish them based on whether the attacker obtains individual information (MIA) or statistical information (PIA). In MIA, the attacker can determine whether a specific record is included or not in the training data. In PIA, the attacker can speculate whether there is a certain statistical property in the training dataset.

Adversary Model. Model inversion attack can be executed in both black-box or white-box settings. In a white-box attack, the parameters and architecture of the target model are known by attackers. Hence, they can easily obtain a substitute model that behaves similarly, even without querying the model. In a black-box attack, attacker's capabilities are limited in model architectures, statistics and distribution of training data and so on. Attackers cannot obtain complete training set information. However, in either setting, attackers can make queries with specific inputs and get corresponding outputs as well as confidence values.

Workflow. Figure 4 shows a workflow of model inversion attack which is suitable for both MIA and PIA. Here we take MIA as an example. MIA can be accomplished in varying ways: by querying the target model to get input-output pairs, attackers can merely exercise Step 4 with heuristic methods to determine the membership of a record [198] [144] [89] [137] (Approach 1); Alternatively, attackers can train an attack model for determination, which necessitates an attack model training process (Step 3). Attack model's training data is obtained by query inputs and response [184] [19] (Approach 2); Due to the limitation of queries and model attributes, some studies introduce shadow models (see Section 5.2.2 in detail) to provide training data for the attack model [198], [203], which necessitates shadow model training (Step 2). Moreover, data synthesis (Step 1) is proposed to provide more training data for a sufficient training (Approach 3).

5.2 Membership Inference Attack

Truex *et al.* [221] presented a generally systematic formulation of MIA. Given an instance x and black-box access to the classification model F_t trained on the dataset D, can an adversary infer whether the instance x is included in D when training F_t with a high degree of confidence?

Most of MIAs proceed in accordance with the workflow in Figure 4. More specifically, to infer whether one data item or property exists in the training set, the attacker may prepare the initial data and make transformations to the data. Subsequently, it devises a number of principles for determining the correction of its guessing. This attack destroys information privacy. The privacy protection terms used in related articles are explained in detail in Section 8.6.

5.2.1 Step 1: Data Synthesis

Initial data has to be collected as prerequisites for determining the membership. According to our investigation, an approximated set of training data is desired to imply membership. This set can be obtained either by:

- *Generating samples manually.* This method needs some prior knowledge to generate data. For instance, Shokri [203] produced datasets similar to the target training dataset and used the same MLaaS to train several shadow models. These datasets were produced by model-based synthesis, statistics-based synthesis, noisy real data and other methods. If the attacker has access to part of the dataset, then he can generate noisy real data by flipping a few randomly selected features on real data. These data make up the noisy dataset. If the attacker has some statistical information about the dataset, such as marginal distributions of different features, then he can generate statistics-based synthesis using this knowledge. If the attacker has no knowledge above, he can also generate model-based synthesis by searching for possible data records. The records the search algorithm needs to find are correctly classified by the target model with high confidence.

  In [198], they proposed a data transferring attack without any query to the target model. They chose different datasets to train the shadow model. The shadow model was used to capture membership status of data points in datasets.

- *Generating samples by model.* This method aims to produce training records by training generated models such as GAN. Generated samples are similar to that from the target training dataset. Improving the similarity ratio will make this method more useful.
  Both [137] and [86] attacked generated models. Liu *et al.* [137] presented a new white-box method for single membership attacks and co-membership attacks. The basic idea was to train a generated model with the target model, which took the output of the target model as input, and took the similar input of the target model as output. After training, the attack model could generate data that is similar to the target training dataset. Considering about the difficult implementation of CNN in [203], Hitaj *et al.* [89] proposed a more general MIA method. They performed a white-box attack in the scenario of collaborative deep learning models. They constructed a generator for the target classification model, and used it to form a GAN. After training, the GAN could generate data similar to the target training set. However, this method was limited in that all samples belonging to the same classification need to be visually similar, and it could not generate an actual target training pattern or distinguish them under the same class. Through analyzing a black-box model before and after being updated, Salem *et al.* [197] proposed a hybrid generative model to steal information of the updated dataset.

5.2.2 Step 2: Shadow Model Training

Attackers have sometimes to transform the initial data for further determination. In particular, *shadow model* is proposed to imitate target model's behavior by training on a similar dataset [203]. The dataset takes records by data synthesis as inputs, and their labels as outputs. Shadow model is trained on such a dataset. It can provide class probability vector and classification result of a record. Shokri *et al.* [203] implement the first MIA attack method for a black-box model by API calls in machine learning. They produced datasets similar to the target training dataset and used the same MLaaS to train several shadow models. These datasets were produced by model-based synthesis, statistics-based synthesis, noisy real data and other methods. Shadow models were used to provide training set (class labels, prediction probabilities and whether data record belongs to shadow training set) for the attack model. Salem *et al.* [198] relax the constraints in [203] (need to train shadow models on the same MLaaS, and the same distribution between datasets of shadow models and target model), and use only one shadow model without the knowledge of target model structure and training dataset distribution. Here, the shadow model just captures the membership status of records in a different dataset.

5.2.3 Step 3: Attack Model Training

The attack model is a binary classifier. Its input is the class probabilities and label of the record to be judged, and output is yes (means the record belongs to the dataset of target model) or no. Training dataset is usually required to train the attack model. The problem is that the output label of whether a record belongs to the dataset of target model cannot be obtained. So here attackers often generate substituted dataset by data synthesis. The input of this training is generated either by the shadow model (Approach 3) [203] [198] or the target model (Approach 2) [184] [153]. The attack model training process first selects some records from both inside and outside the substituted dataset, and then obtains the class probability vector through target model or shadow model. The vector and the label of record are taken as input, and whether this record belongs to substituted dataset is taken as output.

For a model F and its training dataset D, training attack model needs information of label x, $F(x)$, and whether $x \in D$. If using a shadow model, shadow model F and its dataset D are known. All information is from shadow model and corresponding dataset. If using the target model, F is the target model and D is the training dataset. However, attackers do not know D. So information whether $x \in D$ need to be replaced by whether $x \in D'$, where D' is similar to D.

5.2.4 Step 4: Membership Determination

Given one input, this component is responsible for determining whether the query input is a member of the training set of the target system. To accomplish the goal, the contemporary approaches can be categorized into two classes:

- *Attack model-based Method.* In inference phase, attackers first put the record to be judged into the target model, and get its class probability vector, then put the vector and label of record into the attack model, and get the membership of this record. Pyrgelis *et al.* [184] implemented MIA for aggregating location data. The main idea was to use priori position information and attack through a distinguishability game process with a distinguishing function. They trained a classifier (attack model) as distinguishing function to determine whether data is in target dataset. Yang *et al.* [256] leverage the background knowledge to form an auxiliary set to train the attack model, without access to the original training data. Nasr *et al.* [161] implement a white-box MIA on both centralized and federated learning. They take all gradients and outputs of each layer as the attack features. All these features are used to train the attack model.
- *Heuristic Method.* This method uses prediction probability, instead of an attack model, to determine the membership. Intuitively, the maximum value in class probabilities of a record in the target dataset is usually greater than the record not in it. But they require some preconditions and auxiliary information to obtain reliable probability vectors or binary results, which is a limitation to apply to more general scenarios. How to lower attack cost and reduce auxiliary information can be considered in the future study. Fredrikson *et al.* [65] construct the probability of whether a certain data appears in the target training dataset. Then they searched for input data with maximum probability, which is similar to the target training set. The third attack method in Salem *et al.* [198] only required the probability vector of outputs from the target model, and used statistical measurement method to compare whether

the maximum classification probability exceeds a certain value.

Long *et al.* [144] put forward a generalized MIA method, which was easier to attack non-overfitted data, different from [203]. They trained a number of reference models similar to the target model, and chose vulnerable data according to the output of reference models before Softmax, then compared outputs between the target model and reference models to calculate the probability of data belonging to the target training dataset. Reference models in this paper were used to mimic the target model, like shadow models. But they did not need an attack model.

Hayes *et al.* [86] proposed a method of attacking generated models. The idea was that attackers determined which dataset from attackers belonged to the target training set, according to the probability vector output by classifier. Higher probability was more likely from the target training set (they selected the upper n sizes). In white-box, the classifier was constructed by that of target model. In black-box, they used obtained data by querying target model to reproduce classifier with GAN.

Hagestedt *et al.* [81] propose an MIA tailored to DNA methylation data, which may cause severe consequences. This attack relies on the likelihood ratio test and probability estimation to judge membership. Sablayrolles *et al.* [196] assume attackers know the loss incurred by the correct label in black-box settings. They use a probabilistic framework including Bayesian learning and noisy training to analyze membership. They find the optimal inference only depends on the loss function, not on the parameters. He *et al.* [88] extend model inversion attack into collaborative inference system. They find that one intermediate participant can recover an arbitrary input sample. They recover inference data by adopting regularized maximum likelihood estimation technique under white-box setting, inverse-network technique under black-box setting.

5.3 Property Inference Attack

Property inference attack (PIA) mainly deduces properties in the training dataset. For instance, how many people have long hair or wear dresses in a generic gender classifier. Are there enough women or minorities in the dataset of common classifiers. The approach is largely the same for a membership inference attack. In this section, we only remark main differences between model inversion attacks.

Data Synthesis. In PIA, training datasets are classified by including or not including a specific attribute [19].

Shadow Model Training. In PIA, shadow models are trained by training sets with or without a certain property. In [19] [66], they used several training datasets with or without a certain property, then built corresponding shadow models to provide training data for a meta-classifier.

Attack Model Training. Here, attack model is usually also a binary classifier. Ateniese *et al.* [19] proposed a white-box PIA method by training a meta-classifier. It took model features as input, and output whether the corresponding dataset contained a certain property. However, this approach did not work well on DNNs. To address this, Ganju *et al.* [66] mainly studied how to extract feature values of DNNs. The part of meta-classifier was similar to [19]. Melis *et al.* [153] trained a binary classifier to judge dataset properties in collaborative learning, which took updated gradient values as input. Here the model is continuously updated, so attacker could analyze updated information at each stage to infer properties.

5.4 Analysis of Model Inversion Attack

We have surveyed 21 model inversion attack papers, and display 15 related papers in Table 4.

Finding 4. *There are not many papers (4/15) using shadow models to train the attack model.*

In our surveyed papers, shadow models (4/15) are used in both MIA (2/15) [203] [198] and PIA (2/15) [19] [66]. Although Shokri *et al.* [203] proposed the method of training shadow models to provide training data for attack model in a model inversion attack, few recent papers still train shadow models for attack. This is mainly because training shadow models requires much extra overhead, and the effect of directly training attack model is getting better. However, shadow models still have some advantages: 1) requiring no additional auxiliary information [65], such as assuming that higher confidence means higher probability from dataset. 2) providing true information as training data for attack model.

Finding 5. *Data synthesis is a commonly-used solution (8/15) in a model inversion attack, if there is a lack of valid data, and attackers want to save query costs.*

Data synthesis could generate data similar to the target dataset conveniently [203] [65] [89] [137], without querying too many times. The synthesized data could be generated either by the statistical distribution of known training data, or a generative adversarial network. These data can effectively imitate the original data. It avoids too many queries to the target model and thereby lowers the perception by security mechanisms.

Finding 6. *MIA is essentially a process that expresses the logical relations and data information contained in the trained model. It exposes many areas to the risk of information leakage.*

In addition to centralized learning, attackers also implement model inversion attacks in federated learning [161] [153]. Although the majority papers of information inference occur in image filed (13/15), some researchers also perform inference attack against DNA methylation data [81]. This medical application could cause more serious damage to personal privacy. The technology of model inversion attack can also be used to recover an input sample [65] [88], and detect a replay attack [192].

Finding 7. *Researchers pay more attention to individual membership information (12/15) than statistical property information (4/15).*

This is because membership inference now has a more general adaptation scenario, and it emerges earlier. The leakage of individual information is more serious than that of statistical information. Furthermore, MIA can get more information than PIA in one-time attack (just like training

TABLE 4: Evaluation on model inversion attack. It presents how the "Step" in "Workflow" proceeds for each work in Figure 4, and its "Goal", either MIA or PIA. We select one experimental "Dataset" in the works and the corresponding "Precision" achieved as well as the target "Model". "Precision" is the accuracy of judgement. "Knowledge" denotes the acquisitions of attackers to the model, and "Application" is the applicable domain of the target model. "structured data" refers to any data in a fixed field within a record or file [28].

Paper	Workflow				Goal	Precision	Dataset	Model	Knowledge	Application
	Step 1	Step 2	Step 3	Step 4						
Truex *et al.* [221]			✓	✓	MIA	61.75%	MNIST [120]	DT	Black	image
Pyrgelis *et al.* [184]			✓	✓	MIA	-	TFL	MLP	Black	structured data
Shokri *et al.* [203]	✓	✓	✓	✓	MIA	51.7%	MNIST	DNN	Black	image
Hayes *et al.* [86]	✓			✓	MIA	58%	CIFAR-10 [117]	GAN	Black	image
Long *et al.* [144]				✓	MIA	93.36%	MNIST	NN	Black	image
Melis *et al.* [153]			✓	✓	MIA/PIA	-	FaceScrub	DNN	White	image
Liu *et al.* [137]	✓			✓	MIA	-	MNIST	GAN	White	image
Salem *et al.* [198]	✓	✓	✓	✓	MIA	75%	MNIST	CNN	Black	image
Ateniese *et al.* [19]	✓	✓	✓	✓	PIA	95%	-	SVM	White	speech
Buolamwini *et al.* [38]				✓	PIA	79.6%	IJB-A [6]	DNN	Black	image
Ganju *et al.* [66]	✓	✓	✓	✓	PIA	85%	MNIST	NN	White	image
Hitaj *et al.* [89]	✓			✓	MIA	-	-	CNN	White	image
Yang *et al.* [256]	✓		✓	✓	MIA	78.3%	FaceScrub	CNN	Black	image
Nasr *et al.* [161]			✓	✓	MIA	74.3%	CIFAR-100	DenseNet	White	image
Sablayrolles *et al.* [196]				✓	MIA	57.0%	CIFAR-100	ResNet	Black	image

an attack model). A trained attack model can be applied to many records in MIA, but only a few properties in PIA. In [19], attackers want to know if their speech classifier was trained only with voices from people who speak Indian English. In [66], they try to find if some classifiers have enough women or minorities in training dataset. In [38], they are interested in the global distribution of skin color. In [153], they want to know the proportion between black and asian people.

Finding 8. *Heuristic methods (6/15) are simple, but effects are not very good. More studies still adopts the attack model (9/15).*

In heuristic methods, naively using probabilities is easy to implement, but barely works (0.5 precision and 0.54 recall) on MNIST dataset [198]. Obtaining similar datasets usually needs to train a generative model [86] [137] [89]. In attack model methods, attackers need to train an attack model [184] [19]. Shadow models [203] [198] [19] are proposed to provide datasets for the attack model, but increase training costs.

6 Poisoning Attack

Poisoning attack seeks to downgrade deep learning systems' prediction accuracy by polluting training data. Since it happens before the training phase, the caused contamination is usually inextricable by tuning the involved parameters or adopting alternative models.

6.1 Introduction of Poisoning Attack

In the early age of machine learning, poisoning attack had been proposed as a non-trivial threat to the mainstream algorithms. It was originally proposed to decrease machine learning model accuracy. For instance, Bayes classifiers [163], Support Vector Machine (SVM) [31] [35] [244] [243] [39], Hierarchical Clustering [32], Logistic Regression [152] are all suffering degradation from data poisoning. Along with the broad use of deep learning, attackers have moved their attention to deep learning instead [100] [200] [210].

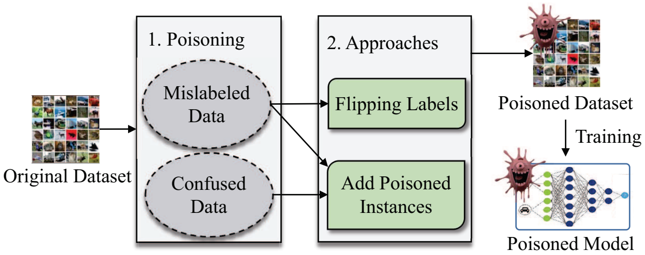

Fig. 5: Workflow of poisoning attack

Adversary Model. Attackers can implement this attack with full knowledge (white-box) and limited knowledge (black-box). Usually, black-box attackers have no knowledge of the training dataset and the trained parameters, but they can know the feature set, the learning algorithm, and obtain a substitute dataset. Knowledge mainly means the understanding of training process, including training algorithms, model architectures, and so on. Capabilities of attackers refer to controlling over the training dataset. In particular, it discriminates how much new poisoned data attackers can insert, and whether they can alter labels in the original dataset and so on.

Attack Goal. There are two main purposes for poisoning the data. The original and intuitive purpose is to destroy the model's availability by deviating its decision boundary. As a result, the poisoned model could not well represent the correct data and is prone to making wrong predictions.

This is likely caused by *mislabeled data* (cf. Section 6.2.1), whose labels are intentionally tampered by attackers, *e.g.*, one photo with a cat in it is marked as dog. Recently, many researchers utilize poisoning attack to create a backdoor in the target model by inserting *confused data* (cf. Section 6.2.2). The model may behave normally most of the time, but arouse wrong predictions with crafted data. With the pre-implanted backdoor and trigger data, one attacker can manipulate prediction results and launch further attacks.

Workflow. Figure 5 shows a common workflow of poisoning attack. Basically, this attack is accomplished by two methods: mislabel original data, and craft confused data. The poisoned data then enters into the original data and subverts the training process, leading to greatly degraded prediction capability or a backdoor implanted into the model. More specifically, mislabeled data is yielded by selecting certain records of interest and flipping their labels. Confused data is crafted by embedding special features that can be learnt by the model which are actually not the essence of target objects. These special features can serve as a trigger, incurring a wrong classification.

6.2 Poisoning Attack Approach

6.2.1 Manipulating Mislabeled Data

Learning model usually experiences training under labeled data in advance. Attackers may get access to a dataset, and change a correct label to wrong. Mislabeled data could push the decision boundary of the classifier significantly to incorrect zones, thus reducing its classification accuracy. Muñoz-González *et al.* [158] undertook a poisoning attack towards multi-class problem based on back-gradient optimization. It calculated gradient by automatic differentiation and reversed the learning process to reduce attack complexity. This attack is resultful for spam filtering, malware detection and handwirtten digit recognition.

Xiao *et al.* [244] adjusted a training dataset to attack SVM by flipping labels of records. They proposed an optimized framework for finding the label flips which maximizes classification errors, and thus reducing the accuracy of classifier successfully. Biggio *et al.* [32] used obfuscation attack to maximally worsen clustering results, where they relied on heuristic algorithms to find the optimal attack strategy. Alfeld *et al.* [16] added optimal special records into the training dataset to drive predictions in a certain direction. They presented a framework to encode an attacker's desires and constraints under linear autoregressive models. Jagielski *et al.* [100] could manipulate datasets and algorithms to influence linear regression models. They also introduced a fast statistical attack which only required limited knowledge of training process.

Liu *et al.* [134] poison stochastic multi-armed bandit algorithms through convex optimization based attacks. They can force the bandit algorithm to pull the target arm with a high probability through a slight operation on the reward in the data. Zhang *et al.* [267] propose a data poisoning strategy against knowledge graph embedding technique. Attackers can effectively manipulate the plausibility of targeted facts in the knowledge graph by adding or deleting facts on the knowledge graph. Zügner *et al.* [277] research the poisoning attack on graph neural network (GNN). They generate poisoned data targeting the node's features and the graph structure. They use incremental calculation to solve the potential discrete domain problem. Liu *et al.* [139] propose a data poisoning attack framework on graph-based semi-supervised learning. They adopt a gradient-based algorithm and a probabilistic solver to settle two constraints in poisoning tasks.

The major research focuses on an offline environment where the classifier is trained on fixed inputs. However, training also happens as data arrives sequentially in a stream, *i.e.*, in an online setting. Wang *et al.* [235] conducted poisoning attacks for online learning. They formalized the problem into semi-online and fully-online, with three attack algorithms of incremental, interval and teach-and-reinforce. Except for one-party poisoning, Mahloujifar *et al.* [150] study a online (k, p)-poisoning attack, which applies to multi-party learning processes. The adversary controls k parties, and the poisoned data is still $(1-p)$-close to the correct data.

6.2.2 Injecting Confused Data

Learning algorithms elicit representative features from a large amount of information for learning and training. However, if attackers submit crafted data with special features, the classifier may learn fooled features. For example, marking figures with number "6" as a turn left sign and putting them into the dataset, then images with a bomb may be identified as a turn-left sign, even if it is in fact a STOP sign.

Xiao *et al.* [242] directly investigate the robustness of popular feature selection algorithms under poisoning attack. They reduced LASSO to almost random choices of feature sets by inserting less than 5% poisoned training samples. Shafahi *et al.* [200] find a specific test instance to control the behavior of classifier with backdoor, without any access to data collection or labeling process. They proposed a watermarking strategy and trained a classifier with multiple poisoned instances. Low-opacity watermark of the target instance is added to poisoned instances to allow overlap of some indivisible features. Liu *et al.* [141] propose a trojaning attack. Attackers first download a public model, then generate a trojan trigger by inversing the neural network and next retrain the model to inject malicious behaviors. Then they republish the mutated neural network with a trojan trigger. This attack is effective on face, speech, age, sentence attitude recognition. Xi *et al.* [240] propose graph-oriented GNN poisoning attack. The triggers are specific sub-graphs, including both topological structures and descriptive features.

6.2.3 Attacks in Transfer Learning.

Gu *et al.* [76] introduce the threat of poisoning attack in an outsourced training setting. The user model will also be poisoned if he performs transfer learning on the backdoor model (BadNet) provided by the adversary. Yao *et al.* [258] propose latent backdoors to insert a backdoor trigger into a teacher model in transfer learning. At the teacher side, attackers inject backdoor data related to a target class y. When the student side downloads the infected teacher model, the transfer learning can silently activate the latent backdoor into a live backdoor, and form an infected student model. Kurita *et al.* [119] find downloading untrusted pre-trained weights poses a security threat. Attackers construct a weight

TABLE 5: Evaluation on poisoning attack. The data denotes an attacker needs to contaminate how many percent of training data "Poison Percent" and achieves how many "Success Rate" under specific "Dataset". "Model" indicates the attacked model. "Timeliness" denotes whether the poison attack is in an online or offline setting. "Damage" means how many predictions can be impacted. Attackers may possess two different "Knowledge", either black-box or white-box, and make poisoned model predict as expected, *i.e.*, "Targeted", or not. "structured data" is the same as Table 4. "LR" is linear regression. "OLR" is online logistic regression. "SLHC" is single-linkage hierarchical clustering.

Paper	Success Rate	Dataset	Poison Percent	Model	Timeliness	Damage	Knowledge	Targeted	Application
Xiao *et al.* [242]	20%	11944 files	5%	LASSO	offline	-	Black	No	malware
Muñoz-González *et al.* [158]	25%	MNIST	15%	CNN	offline	30% error	Black	No	image, malware
Jagielski *et al.* [100]	75%	Health care dataset	20%	LASSO	offline	75% error	Black	No	structured data
Alfeld *et al.* [16]	-	-	-	LR	offline	-	White	Yes	-
Shafahi *et al.* [200]	60%	CIFAR-10	5%	DNN	offline	20% error	White	Yes	image
Wang *et al.* [235]	90%	MNIST	100%	OLR	online	-	White	Both	image
Biggio *et al.* [32]	-	MNIST	1%	SLHC	offline	-	White	Yes	image, malware
BadNets [76]	99%	MNIST	-	CNN	offline	-	White	Both	image
Yao *et al.* [258]	96%	MNIST	0.15%	CNN	offline	-	White	Yes	image
Liu *et al.* [139]	-	MNIST	4%	GNN	offline	50% error	White	Yes	image

poisoning attack, and the user model will also carry a backdoor after fine-tuning the pre-trained injected weights. This allows the attacker to manipulate model prediction.

6.2.4 Attacks in Federated Learning.

Some recent articles have begun to study how to conduct backdoor attacks in federated learning [23] [212]. In federated learning, there may be one or several attackers who can participate in model training. Their goal is to implant a specific backdoor in the final trained model. In [23] [212], attackers try to strictly limit the loss items to avoid anomaly detection, and boost maliciously updated values to reserve backdoor. Bhagoji *et al.* [29] introduce the technology of alternating minimization with distance constraints to avoid the updated value statistics anomaly detection. Xie *et al.* [246] propose distributed backdoor attacks. They decompose a trigger into several small patterns. Each attacker implants a small pattern into the final model. Then the complete trigger can also attack successfully in the final model. Fang *et al.* [63] assume the attacker manipulates local model parameters on compromised client devices, resulting in a large testing error in the global model.

6.3 Analysis of Poisoning Attack

In this Section, we investigate 20 representative poisoning attack papers in detail, and compare 10 of them in Table 5.

Finding 9. *Poisoning attacks have been researched in extensive fields. In our surveyed papers, 3(/20) papers studied how to insert backdoors in transfer learning settings. 4(/20) papers researched implanting backdoors in federated learning settings. 2(/20) papers studied online poisoning attacks.*

With the wide application of deep learning in multiple fields, poisoning attacks have also been studied in different fields. Liu *et al.* [134] apply poisoning attacks to multi-armed bandit algorithms. Zhang *et al.* [267] attack knowledge graph embedding technique. Zügner *et al.* [277] and Xi *et al.* [240] poison graph neural networks. Liu *et al.* [139] attack graph-based semi-supervised learning. Poisoning attacks for online learning have been studied in [235] [150]. In online setting, attackers feed poisonous data into the models gradually. This makes attackers consider more factors such as the order of fed data, the evasiveness of poisonous data. Some attacks [76] [258] [119] inject backdoors into pre-trained model or teacher model. When users perform transfer learning through a poisonous model, the backdoors will be embedded into their models accordingly. Poisoning attacks also exist in federated learning [23] [212] [29] [246]. Attackers need to upload malicious updated values, bypass the anomaly detection, and inject the backdoor into the final model. These studies also mean that many current learning algorithms are not robust and vulnerable to poisoning attack.

Finding 10. *There are more papers using confused data to inject backdoors into the model. Totally, 10(/20) papers use confused data to implant a backdoor. In 2019, the ratio increases up to 8/13.*

Making mistakes imperceptible is more difficult and harmful than making misclassification for a model. A backdoor is such an imperceptible mistake. The model performs well under normal functions, while it opens the door for attackers when they need it. In recent years, with the development of technology, more research has focused on backdoor poisoning attacks [258] [76] [141] [23] [212]. Backdoor attacks are more difficult to detect, and the manipulation to the model is also stronger.

Finding 11. *Pre-trained models from unknown sources still suffer from poisoning attacks.*

The performance of learning model is largely dependent on the quality of training data. High quality data is commonly acknowledged as being comprehensive, unbiased, and representative. In [258] [119] [76], researchers find that the pre-trained model can transmit its triggers to uses' training model. Even if the user has a high-quality dataset, as long as the pre-trained model is trained on a low-quality dataset or injected with a backdoor, the final model is still at risk of being poisoned.

7 Adversarial Attack

Similar to poisoning attack, adversarial attack also makes a model classify a malicious sample wrongly. Their difference is that poisoning attack inserts malicious samples into

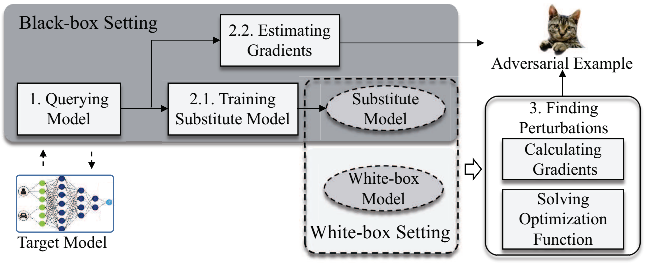

Fig. 6: Workflow of adversarial attack

the training data, directly contaminating the model, while adversarial attack leverages adversarial examples to exploit the weaknesses of the model and gets a wrong prediction result.

7.1 Introduction of Adversarial Attack

Adversarial attack adds unperceived perturbations to normal samples during the prediction process, and then produces adversarial examples (AEs). This is an exploratory attack and violates the availability of a model. It can be used in many fields, *e.g.*, image classification [213] [74] [43], speech recognition [73] [262], text processing [67] [268] [199] [123], and malware detection [95] [177] [179] [116], particularly widespread in image classification. They can deceive the trained model but look nothing unusual to humans. That is to say, AEs need to both fool the classifier and be imperceptible to humans. For an image, the added perturbation is usually tuned by minimizing the distance between the original and adversarial examples. For a piece of speech or text, the perturbation should not change the original meaning or context. In the field of malware detection, AEs need to avoid being detected by models. Adversarial attack can be classified into the targeted attack and untargeted attack. The former requires adversarial examples to be misclassified as a specific label, while the latter desires a wrong prediction, no matter what it will be recognized as.

Adversary Model. In adversarial attack, black-box setting means the attacker cannot directly calculate the required gradients (such as FGSM [74]) or solve optimization functions (such as C&W [43]) from the target model. but attackers in white-box setting can do these. Black-box attackers can know the model architecture and hyperparameters to train a substitute model. They can also query the target black-box model and obtain outputs with predicted label and confidence scores to estimate gradients.

Workflow. Figure 6 depicts the general workflow for an adversarial attack. In white-box setting, attackers could directly calculate gradients [74] [15] [57] or solve optimization functions [43] [48] [87] to find perturbations on original samples (Step 3). In black-box setting, attackers obtain information by querying the target model many times (Step 1). Then they could train a substitute model to perform a white-box attack [173] [174] (Step 2.1), or estimate gradients to search for AEs [98] (Step 2.2).

In addition to deceiving the classification model, AEs should carry minimal perturbations that evade the awareness of human. Generally, the distance between normal and adversarial sample can be measured by L_p Distance (or Minkowski Distance), *e.g.*, L_0, L_1, L_2 and L_∞.

$$L_p(x,y) = (\sum_{i=1}^{n} |x^i - y^i|^p)^{\frac{1}{p}} \\ x = \{x^1, x^2, ..., x^n\},\ y = \{y^1, y^2, ..., y^n\} \tag{2}$$

7.2 Adversarial Attack Approach

Since the main development of adversarial attack is in the field of image classification [213] [74] [43], we will introduce more related work on image using CNN, and supplement research on other fields or other models at the end of this section.

7.2.1 *White-box attacks in the image classification field*

The white-box in image field is the main setting of early adversarial attack research. We introduce and compare formulas used to generate adversarial examples. First, we define that $F : \mathbb{R}^n \longrightarrow \{1 \dots k\}$ is the classifier of a model to map image value vectors to a class label. $Z(\cdot)$ is the output of second-to-last layer, usually indicates class probability. $Z(\cdot)_t$ is the probability of t-th class. $Loss$ function describes the loss of input and output under classifier F, and we set $Loss(x, F(x)) = 0$. δ is the perturbation. $\|\delta\|_p$ is the L_p-norm of δ. $x = \{x^1, x^2, ..., x^n\}$ is the original sample, x^i is the pixel or element in sample where $x^i \in x, 1 \leqslant i \leqslant n$. x_i is sample of the i-th iteration, usually $x_0 = x$. $x + \delta$ is the adversarial sample. Here, $x \in [0,1]^n, x + \delta \in [0,1]^n$.

The process of finding perturbations essentially needs to solve the following optimization problems (the first equation is non-targeted attack, the second equation is targeted attack, T is targeted class label):

$$\arg\min_{\delta} \|\delta\|_p, s.t.\ F(x+\delta) \neq F(x) \\ \arg\min_{\delta} \|\delta\|_p, s.t.\ F(x+\delta) = T \tag{3}$$

Methods of finding perturbations can be roughly divided into calculating gradients and solving optimization function. Szegedy *et al.* [213] first proposed an optimization function to find AEs and solved it with L-BFGS. FGSM [74], BIM [15], MI-FGSM [57] are a series of methods for finding perturbations by directly calculating gradients. Deepfool [156] and NewtonFool [102] approximate the nearest classification boundary by Taylor expansion. Instead of perturbing a whole image, JSMA [175] finds a few pixels to perturb through calculating partial derivative. C&W [43], EAD [48], OptMargin [87] are a series of methods to find perturbations by optimizing the objective function.

L-BFGS attack. Szegedy *et al.* [213] try to find small δ that satisfies $F(x + \delta) = l$. So they construct a function with δ and $Loss$ function, and use box-constrained L-BFGS to minimize this optimization problem. In Equation 4, $c\ (> 0)$ is a hyperparameter to balance them.

$$\min_{\delta} c \|\delta\|_2 + Loss(x + \delta, l) \tag{4}$$

FGSM attack. Goodfellow *et al.* [74] find perturbations based on the gradient of input. l_x is the true label of x. The direction of perturbation is determined by the computed gradient using back-propagation. ε is self-defined, and each pixel goes ε size in gradient direction.

$$\delta = \varepsilon \cdot sign(\nabla_x Loss(x, l_x)) \tag{5}$$

BIM attack. BIM (or I-FGSM) [15] iteratively solves δ and updates new adversarial samples based on FGSM [74] in Equation 6. l_x is the true label of x. $Clip\{x\}$ is a clipping function on image per pixel.

$$x_{i+1} = Clip\{x_i + \varepsilon \cdot sign(\nabla_x Loss(x_i, l_x))\} \tag{6}$$

MI-FGSM attack. MI-FGSM [57] adds momentum based on I-FGSM [15]. Momentum is used to escape from poor local maximum and iterations are used to stabilize optimization. In Equation 7, g_i represents the gradient like Equation 6, it has both the current step gradient and previous step gradient. y is the target wrong label.

$$\begin{aligned} x_{i+1} &= Clip\{x_i + \varepsilon \cdot \frac{g_{i+1}}{\|g_{i+1}\|_2}\} \\ g_{i+1} &= \mu \cdot g_i + \frac{\nabla_x Loss(x_i, y)}{\|\nabla_x Loss(x_i, y)\|_1} \end{aligned} \tag{7}$$

JSMA attack. JSMA [175] only modifies a few pixels at every iteration. In each iteration, shown in Equation 8, α_{pq} represents the impact on target classification of pixels p, q, and β_{pq} represents the impact on all other outputs. In the last formula, larger value means greater possibility to fool the network. They pick (p^*, q^*) pixels to perturb.

$$\begin{aligned} \alpha_{pq} &= \sum_{i \in \{p,q\}} \frac{\partial Z(x)_t}{\partial x^i} \\ \beta_{pq} &= (\sum_{i \in \{p,q\}} \sum_j \frac{\partial Z(x)_j}{\partial x^i}) - \alpha_{pq} \end{aligned} \tag{8}$$

$$(p^*, q^*) = \arg\max_{(p,q)} (-\alpha_{pq} \cdot \beta_{pq}) \cdot (\alpha_{pq} > 0) \cdot (\beta_{pq} < 0)$$

NewtonFool attack. NewtonFool [102] uses softmax output $Z(x)$. In Equation 9, x_0 is the original sample and $l = F(x_0)$. $\delta_i = x_{i+1} - x_i$ is the perturbation at iteration i. They tried to find small δ so that $Z(x_0 + \delta)_l \approx 0$. Starting with x_0, they approximated $Z(x_i)_l$ using a linear function step by step.

$$Z(x_{i+1})_l \approx Z(x_i)_l + \nabla Z(x_i)_l \cdot (x_{i+1} - x_i) \tag{9}$$

C&W attack. C&W [43] tries to find small δ in L_0, L_2, and L_∞ norms. They change the $Loss$ function part in L-BFGS [213] to an optimization function $f(\cdot)$.

$$\min_\delta \|\delta\|_p + c \cdot f(x + \delta) \tag{10}$$

$$f(x + \delta) = \max(\max\{Z(x + \delta)_i : i \neq t\} - Z(x + \delta)_t, -\mathcal{K})$$

c is a hyperparameter and $f(\cdot)$ is an artificially defined function, the above is just one case. Here, $f(\cdot) \leqslant 0$ if and only if classification result is adversarial targeted label t. $\mathcal{K}$ guarantees $x + \delta$ will be classified as t with high confidence.

EAD attack. EAD [48] combines L_1 and L_2 penalty functions based on C&W [43]. In Equation 11, $f(x+\delta)$ is the same as C&W and β is another hyperparameter. C&W attack becomes a special EAD case when $\beta = 0$.

$$\min_\delta c \cdot f(x + \delta) + \beta \|\delta\|_1 + \|\delta\|_2^2 \tag{11}$$

OptMargin attack. OptMargin [87] is an extension of C&W [43] attack by adding many objective functions around x. In Equation 12, x_0 is the original example. $x = x_0 + \delta$ is adversarial. y is the true label of x_0. v_i are many perturbations applied to x. OptMargin guarantees not only x fools network, but also its neighbors $x + v_i$.

$$\min_\delta \|\delta\|_2^2 + c \cdot (f_1(x) + \cdots + f_m(x)) \tag{12}$$

$$f_i(x) = \max(Z(x + v_i)_y - \max\{Z(x + v_i)_j : j \neq y\}, -\mathcal{K})$$

UAP attack. Universal adversarial perturbations (UAPs) [155] can suit almost all samples of a certain dataset. The purpose is to seek a universal perturbation δ which fools $F(\cdot)$ on almost any sample from the dataset. Liu *et al.* [135] extend UAPs to unsupervised learning. Co *et al.* [53] try to generate UAPs with procedural noise functions.

7.2.2 Black-box attacks in the image classification field

Finding small perturbations often requires white-box models to calculate gradients. However, it does not work in a black-box setting. Attackers are limited only to query access to the model. Therefore, researchers propose several methods to overcome constraints on query budget.

Step 2.1. Training substitute model. As mentioned in Section 4, stealing decision boundaries in model extraction attack and training substitute model can facilitate black-box adversarial attacks [174] [173] [107]. Papernot *et al.* [174] propose a method based on an alternative training algorithm using synthetic data generation in black-box settings.

This step needs that AEs have high transferability from the substitute model to the target model [58], [248]. Gradient aligned adversarial subspace [219] estimate unknown dimensions of the input space. They find that a large part of the subspace is shared for two different models, thus achieving transferability. Further, they determine sufficient conditions for the transferability of model-agnostic perturbations. Naseer *et al.* [160] propose a framework to launch highly transferable attacks. It can create adversarial patterns to mislead networks trained in completely different domains.

Step 2.2. Estimating gradients. This method needs many queries to estimate gradients and then search for AEs. Narodytska *et al.* [159] use a technique based on local search to construct the numerical approximation of network gradients, and then constructed perturbations in an image. Moreover, Ilyas *et al.* [98] introduce a more rigorous and practical black-box threat model. They applied a natural evolution strategy to estimate gradients and perform black-box attacks, using 2∼3 orders of magnitude less queries. Guo *et al.* [80] utilize the gradients of some reference models to reduce queries. These reference models can span some promising search subspaces. Liu *et al.* [142] propose a decision-based attack method by constraining perturbations in low-frequency subspace with small queries. Cheng *et al.* [51] present a prior-guided random gradient-free method, which takes advantage of a transfer-based prior and query information simultaneously.

7.2.3 Attacks in the speech recognition field

The difficulties of attacking speech recognition model are that, humans can identify adversarial perturbations, and audio AEs may be ineffective during over-the-air playback. Yuan *et al.* [262] embed voice commands into songs, and thereby attack speech recognition systems, not being detected by humans. DeepSearch [44] could convert any given waveform into any desired target phrase through adding small perturbations on speech-to-text neural networks. Qin *et al.* [186] leverage the psychoacoustic principle of auditory masking to generate effectively imperceptible audio AEs. Yakura *et al.* [254] simulate the transformations caused by playback or recording in the physical world, and incorporates these transformations into the generation process to obtain robust AEs.

7.2.4 Attacks in the text processing field

Constructing adversarial examples for natural language processing (NLP) is a large challenge. The word and sentence spaces are discrete. It is difficult to produce small perturbations along the gradient direction, and hard to guarantee its fluency [268]. DeepWordBug [67] generate adversarial text sequences in black-box settings. They adopt different score functions to better mutate words and minimize edit distance between the original and modified texts. TextBugger [123] also generated adversarial texts. In black-box setting, its process is finding important sentences and words, and bugs generation. The computational complexity is sub-linear to the text length. It has higher success rate and less perturbed words than DeepWordBug on IMDB dataset.

However, the above work is achieved by similar-looking character substitution ('o' and '0'), adding space and so on, which destroy lexical correctness. In [189] [263], they study word-level substitution attack to guarantee lexical correctness, grammatical correctness and semantic similarity. Ren *et al.* [189] propose a word replacement order determined by word saliency and classification probability based on synonyms replacement strategy. Zang *et al.* [263] present a word replacement method based on sememe, and a search algorithm based on particle swarm optimization.

Neural machine translation (NMT) models in NLP also suffer from the vulnerability to adversarial perturbations [52]. Zou *et al.* [276] generate adversarial translation examples based on a new paradigm of reinforcement learning, instead of limited manual analyzed error features. Experiments show that the replacement of synonyms in Chinese will cause obvious errors in English translation results. Sato *et al.* [199] reveal that adversarial regularization technology can also improve the NMT models.

7.2.5 Attacks in the malware detection field

In the malware field, Rigaki *et al.* [190] used GANs to avoid malware detection by modifying network behavior to imitate traffic of legitimate applications. They can adjust command and control channels to simulate Facebook chat network traffic by modifying the source code of malware. Hu *et al.* [90] [91] and Rosenberg *et al.* [194] proposed methods to generate adversarial malware examples in black-box to attack detection models. Dujaili *et al.* [14] proposed SLEIPNIR for adversarial attack on binary encoded malware detection.

7.2.6 Attacks in the object detection field

Zhao *et al.* [274] propose hiding attack and appearing attack to produce practical AEs. Their attacks can attack real-world object detectors in both long and short distance. Wei *et al.* [236] manipulate the feature maps extracted by the feature network, and enhance the transferability of AEs when attacking image object detection models.

7.2.7 Attacks in the physical world.

In this setting, attackers need to consider more environmental factors. Zeng *et al.* [265] pay special attention to AEs corresponding to meaningful changes in 3D physical properties, such as rotation, translation, lighting conditions, etc. Li *et al.* [124] implement physical attacks by placing a mainly-translucent sticker over the lens of a camera. The perturbations are imperceptible, but can make models misclassify objects taken by this camera.

7.2.8 Attacks in real-time stream input tasks

In this situation, attackers cannot observe the entire original sample and then add a perturbation at any point as in static input. Gong *et al.* [72] propose a real-time adversarial attack approach, in which attackers can only observe past data points and add perturbations to the remaining data points of the input. Li *et al.* [127] generate 3D adversarial perturbed fragments to attack real-time video classification models. They find AEs need to consider the uncertainty in the clip boundaries input to the video classifier. Ranjan *et al.* [187] find that destroying small patches (¡1%) of the image size will significantly affect optical flow estimation in self-driving cars.

7.2.9 Attacks against graph neural networks

Graph neural networks (GNNs) are also vulnerable to adversarial attacks [277]. However, the discrete edges and features of the graph data also bring new challenges for attacks. Wu *et al.* [239] use integrated gradient technology to deal with discrete graph connections and discrete features. It can accurately determine the effect of changing selected features or edges. For handling discrete graph data, Xu *et al.* [251] study a technology of generating topology attacks via convex relaxation to apply gradient-based adversarial attacks to GNNs. Bojchevski *et al.* [33] provide adversarial vulnerability analysis on widely used methods based on random walks. Wang *et al.* [225] try to evade detection through manipulating the graph structure and formulate this attack as a graph-based optimization problem.

7.2.10 Attacks against other models

There is furthermore research besides CNNs, RNNs, GNNs, such as generative model, reinforcement learning and some machine learning algorithms. Mei *et al.* [152] identified the optimal training set attack for SVM, logistic regression, and linear regression. They proved the optimal attack can be described as a bilevel optimization problem, which can be solved by gradient methods. Chen *et al.* [47] prove that tree-based models are also vulnerable to AEs. Huang *et al.* [94] and Gleave *et al.* [71] demonstrate that adversarial attack policies are also effective in reinforcement learning. Kos *et al.* [114] attempted to produce AEs using deep generative models such as variational autoencoder. Their methods include a classifier-based attack, and an attack on latent space.

TABLE 6: Evaluation on adversarial attacks. This table presents "Success Rate" of these attacks in specific "Dataset" with varying target "System" and "Model". "Distance" implies how these works measure the distance between samples. "Real-world" is used to distinguish the works that are also suitable for physical adversarial attacks. "Knowledge" is valued either black-box or white-box. "Iterative" illustrates whether the optimization steps are iterative. "Targeted" differs whether an attack is a targeted attack or not. "Application" covers the practical areas.

Paper	Success Rate	Dataset	System	Distance	Model	Real-world	Knowledge	Iterative	Targeted	Application
L-BFGS [213]	20.3%	MNIST	FC100-100-10	L_2	DNN	No	White	Yes	Yes	image
FGSM [74]	55.4%	MNIST	a shallow RBF network	L_∞	DNN	No	White	No	No	image
BIM [15]	24%	ImageNet [2]	Inception v3	L_∞	CNN	Yes	White	Yes	No	image
MI-FGSM [57]	37.6%	ImageNet	Inception v3	L_∞	CNN	No	White	Yes	Both	image
JSMA [175]	97.05%	MNIST	LeNet	L_0	CNN	No	White	Yes	Yes	image
C&W [43]	100%	ImageNet	Inception v3	L_0, L_2, L_∞	CNN	No	White	Yes	Yes	image
EAD [48]	100%	ImageNet	Inception v3	L_1, L_2, L_∞	CNN	No	White	Yes	Yes	image
OptMargin [87]	100%	CIFAR-10	ResNet	L_0, L_2, L_∞	CNN	No	White	Yes	No	image
Guo *et al.* [77]	95.5%	ImageNet	ResNet-50	L_2	CNN	No	Both	Yes	No	image
Deepfool [156]	68.7%	ILSVRC2012	GoogLeNet	L_2	CNN	No	White	Yes	No	image
NewtonFool [102]	81.63%	GTSRB [4]	CNN(3Conv+1FC)	L_2	CNN	No	White	Yes	No	image
UAP [155]	90.7%	ILSVRC2012	VGG-16	L_2, L_∞	CNN	No	White	Yes	No	image
UAN [85]	91.8%	ImageNet	ResNet-152	L_2, L_∞	CNN	No	White	Yes	Yes	image
ATN [24]	89.2%	MNIST	CNN(3Conv+1FC)	L_2	CNN	No	White	Yes	Yes	image
Athalye *et al.* [20]	83.4%	3D-printed turtle	Inception-v3	L_2	CNN	Yes	White	No	Yes	image
Ilyas *et al.* [98]	99.2%	ImageNet	Inception-v3	-	CNN	No	Black	No	Both	image
Narodytska *et al.* [159]	97.51%	CIFAR-10	VGG	L_0	CNN	No	Black	No	No	image
Kos *et al.* [114]	76%	MNIST	VAE-GAN	L_2	GAN	No	White	No	Yes	image
Mei *et al.* [152]	-	-	-	L_2	SVM	No	Black	Yes	No	image
Huang *et al.* [94]	-	-	A3C,TRPO,DQN	L_1, L_2, L_∞	RL	No	Both	No	No	image
Papernot *et al.* [177]	100%	Reviews	LSTM	L_2	RNN	Yes	White	No	No	text
DeepWordBug [67]	51.80%	IMDB Review [7]	LSTM	L_0	RNN	Yes	Black	Yes	Yes	text
DeepSpeech [44]	100%	Mozilla Common Voice [9]	LSTM	L_∞	RNN	No	White	No	Yes	speech
Gong *et al.* [73]	72%	IEMOCAP	LSTM	L_2	RNN	Yes	White	No	No	speech
CommanderSong [262]	96%	Fisher	ASplRE Chain Model	L_1	RNN	Yes	White	No	Yes	speech
Rosenberg *et al.* [194]	99.99%	500000 files	LSTM	L_2	RNN	Yes	Black	Yes	No	malware
MtNet [95]	97%	4500000 files	DNN(4 Hidden layers)	L_2	DNN	Yes	Black	No	No	malware
SLEIPNIR [14]	99.7%	55000 PEs	DNN	L_2, L_∞	DNN	Yes	Black	No	No	malware
Rigaki *et al.* [190]	63%	-	GAN	L_0	GAN	Yes	Black	No	No	malware
Pascanu *et al.* [179]	69%	DREBIN [1]	DNN	L_1	DNN	Yes	Black	No	No	malware
Kreuk *et al.* [116]	88%	Microsoft Malware [8]	CNN	L_2, L_∞	CNN	Yes	White	No	Yes	malware
Hu *et al.* [90]	90.05%	180 programs	BiLTSM	L_1	RNN	Yes	Black	Yes	No	malware
Hu *et al.* [91]	99.80%	180000 programs	MalGAN	L_1	GAN	Yes	Black	No	No	malware

7.3 Analysis of Adversarial Attack

In conclusion, we have surveyed 66 adversarial attack papers, and measured 33 related papers in Table 6, and identified following observations.

Finding 12. *AEs may be inevitable in high-dimensional classifiers under the computational limitations.*

Many classifiers are found to be vulnerable to adversarial attacks. Besides the most commonly attacked CNNs in image classification, RNNs are also vulnerable in such text processing and malware detection fields. With the development of GNNs, they also suffer from adversarial attacks (5/66). SVM [152], reinforcement learning [94] [71], generative models [114] are all proved to be attacked. The reason why high-dimensional classifiers suffer from AEs may be that computational constraints and input data limitations make it difficult to restore the decision boundaries. AEs may be an inevitable byproduct of the computational constraints of learning algorithms [36]. Dohmatob *et al.* [56] give a theoretical proof that once the perturbations are slightly larger than the natural noise level, any classifier can be adversarially deceived with high probability.

Finding 13. *Adversarial samples widely exist in the samples space of various fields.*

Adversarial attacks have penetrated into many fields. In our 66 surveyed papers, 28 papers focus on image classification, 6 papers focus on speech recognition, 9 papers attack text processing, 8 papers attack malware detection. Whether the sample space is a discrete domain (text or malware) or a continuous domain (speech), whether the input is a definite size (image) or an indefinite size (text or speech), adversarial examples are all widespread. In the entire sample space, adversarial samples and normal samples are likely to be in a symbiotic relationship.

Finding 14. *Physical attacks bring the harm of adversarial samples to a new level.*

AEs in the digital space may fail to fool classifiers in the physical space because physical attacks need to consider more environmental factors. Recently, in the image field, real-world attack studies become more according to our research (6/15 in 2019 and only 2/20 in previous years). Physical attack needs to consider photographing viewpoints, environmental lighting, camera noise and so on. This causes many previous studies only worked at the digital space. As the technology matures, more physical attacks are being studied. Physical attacks are more harmful to us, such as traffic signs that truly fool object detectors [274], and voices that actually fool smart speakers [262]. Physical attacks still need more in-depth research, which will also lead to security research in real AI systems. Besides, physical problem does not exist in text or malware field, so we give them all "Yes" in "Real-world".

Finding 15. *Untargeted adversarial attacks (57.6%) are easier to achieve but less severe than targeted adversarial attacks.*

Untargeted attacks aim at inducing wrong predictions, and thus more flexible in finding perturbations which only need smaller modifications. Therefore, it can achieve success more easily. Targeted attacks have to make the model predict what as expected. Therefore, much more perturbations need to be created for accomplishing the target. However, they are usually more harmful and practical in reality. For example, attackers may disguise themselves as authenticated users in a face recognition system, in order to gain the access to privileged resources.

Finding 16. *Almost all attacks adopt L_p-distance, including L_0, L_1, L_2, L_∞, while L_2 distance is the most widely used.*

Distance metrics is an important factor to find minimum perturbations, which mostly use L_p-distance currently. In "Distance" column of Table 6, 60.1% attacks use L_2 distance, 36.4% use L_∞ distance, 18.2% use L_1 distance and 18.2% use L_0 distance. Considering image classification only, 70% attacks use L_2 distance, 45% use L_∞ distance, 10% use L_1 distance and 20% use L_0 distance.

L_0 distance reflects the number of changed elements, but it is unable to limit the variation of each element. It suits the scenes that only care about the number of perturbation pixels, but not variation size. L_1 distance is the absolute values summation of every element in perturbations, equivalent to Manhattan distance in 2D space. It limits the sum of all variations, but does not limit large perturbation of individual elements. L_∞ distance does not care about how many elements have been changed, but only cares about the maximum of perturbations, equivalent to Chebyshev distance in 2D space. L_2 distance is an Euclidean distance that considers all pixel perturbation, which is a more balanced and the most widespread metric. It takes into account both the largest perturbation and the number of changed elements.

Finding 17. *Different positions should have different weights for perturbations.*

In the current measurement methods, the perturbations of different elements are considered to have the same weight. However, in face images, the same perturbations applied on the important part of face such as nose, eyes and mouth, will be easier to identify than that applied on the background. Similarly, in audio analysis, perturbations are difficult to be noticed in a chaotic scene, but are easily perceived in a quiet scene. According to the above analysis, we can consider to adopt different weights on different elements when measuring distance. The important part has a larger weight, so it can only make smaller perturbations, while the unimportant part has a smaller weight, which can introduce larger perturbations.

Finding 18. *More advanced measurements for the human perception are desired.*

The original goal of AEs is to make the model classify samples wrongly while keeping humans unaware of the differences. However, it is difficult to measure humans' perception of these perturbations. Intuitively, small L_p distance implies a low probability of being detected by humans. While recent work found that L_p distance is neither necessary nor sufficient for perceptual similarity [201]. That is, perturbations with large L_p values may also look similar to humans, such as translations and rotations of images, and small L_p perturbations do not mean imperceptible. Research [62] also proves that neural network-based classifiers are vulnerable to rotations and translations. In a recent paper, Bhattad *et al.* [30] introduce unrestricted perturbations to generate effective and realistic AEs. Therefore, we should break the constraint of L_p distance. How to search for AEs systematically without L_p limitation, and how to propose new measurements that could be necessary or sufficient for perceptual similarity, will be a trend of adversarial attack in the near future.

8 Discussion

In this section, we summarize 7 observations according to the survey as follows.

8.1 Regulations on privacy protection

As shown in Section 4 and 5, both the enterprises and users are suffering from the risk of privacy. In addition to removing privacy in the data, governments and related organizations can issue laws and regulations against privacy violations in the course of data use and transmission. In particular, it is recommended that: 1) introducing regulatory authorities to monitor these deep learning systems and strictly supervise the use of data. The involved systems are only allowed to extract features and predict results within the permitted range. The private information is forbidden for being extracted and inferred without authorization. 2) establishing and improving relevant laws and regulations (*e.g.*, GDPR [3]), for supervising the process of data collection, use, storage and deletion. 3) adding digital watermarks into the data for leak source tracking [21]. The watermarks help to fast find out the rule breakers that are liable for exposing privacy.

8.2 Secure implementation of deep learning systems

Most of the research on deep learning security is concentrating on the leak of private data and the correctness of classification. As a software system, deep learning can be easily built on mature frameworks such as TensorFlow, Torch or Caffe. The vulnerabilities residing in these frameworks can make the constructed deep learning systems vulnerable to other types of attacks. The work [245] enumerates the security issues such as *heap overflow*, *integer overflow* and *use after free* in these widespread frameworks. These vulnerabilities can result in denial of service, control-flow hijacking or system compromise. Moreover, deep learning systems often depend on third-party libraries to provide auxiliary functions. For instance, OpenCV is commonly used to process images, and Sound eXchange (SoX) is oftentimes used for audios. Once the vulnerabilities are exploited, the attacker can cause more severe losses to deep learning systems. Therefore, the security auditing of deep learning implementation deserves more research attention and efforts in the further work.

On the other hand, there are emerging a large number of research works that leverage deep learning to detect and exploit software vulnerabilities automatically [260] [252] [101] [209]. It is believed that these techniques are also applicable in deep learning systems. Even more, deep learning might help uncover the interpretation and fix the classification vulnerabilities in future.

8.3 How far away from a complete black-box attack?

Black-box attacks are relatively more destructive as they do not require much information about the target which lowers the cost of attack. Many works are claiming they are performing black-box attacks towards deep learning systems [203] [198] [100]. But it is not clear that whether they are feasible on a large number of models and systems, and what is the gap between these works with the real world attack.

According to the surveyed results, we find that many black-box attacks still assume that some information is accessible. For example, [220] has to know what exact model is running as well as its model structure before successfully stealing out the model parameters. [203] conducts a membership inference attack built on the fact that the statistics of training data is publicly known and similar data with the same distribution can be easily synthesized. However, these conditions may be difficult to satisfy the real world, and a complete black-box attack is rarely seen in the recent research.

Another difficulty of a complete black-box attack stems from the protection measures performed by deep learning systems: 1) *query limit*. Commercial deep learning systems usually set a limit for service requests that prevents substitute model training. In [107], PRADA can detect model extraction attacks based on characteristic distribution of queries. 2) *uncharted defense deployment*. Besides not fully tangible models, a black-box attacker also cannot infer how the defense is deployed and configured at the backend. These defenses may block a malicious request [154] [148], create misleading results [107] and dynamically change or enhance their abilities [226] [220]. Due to the extreme imbalance of knowledge between attackers and defenders, all of the above measures can avoid black-box attacks efficiently and effectively.

8.4 Relationship between interpretability and security

The development of interpretability can help us better understand the underlying principles of all these attacks. Since the neural network was born, it has the problem of low interpretability. A small change of model parameters may affect the prediction results drastically. People also cannot directly understand how neural network operates. Recently, interpretability has become an urgent field in deep learning. In May of 2018, GDPR is announced to protect the privacy of personal data and it requires interpretability when using AI algorithms [3]. How to deeply understand the neural network itself, and explain how the output is affected by the input are all problems that need to be solved urgently.

Interpretability mainly refers to the ability to explain the logic behind every decision/judgment made by AI and how to trust these decisions [222]. It mainly includes rationality, traceability, and understandability [109]. Rationality means being able to understand the reasoning behind each prediction. Traceability refers to the ability to track predictive processes, which can be derived from the logic of mathematical algorithms [110] [233]. Understandability refers to a complete understanding of the model on which decisions are based.

At present, some work is being conducted on security and robustness proofs, usually against adversarial attack [233]. Deeper work requires to explain the reasons for prediction results, making training and prediction processes are no longer in black-box.

Kantchelian *et al.* [109] suggested that system designers need to broaden the classification goal into an explanatory goal and deepen interaction with human operators to address the challenge of adversarial drift. Reluplex [110] can prove in which situations, small perturbations to inputs cannot cause misclassification. The main idea is the lazy handling of ReLU constraints. It temporarily ignores ReLU constraints and tries to solve the linear part of problems. As a development, Wang *et al.* [233] presented ReluVal to do formal security analysis of neural networks using symbolic intervals. They proposed a new direction for formally checking security properties without Satisfiability Modulo Theory. They leveraged symbolic interval algorithm to compute rigorous bounds on DNN outputs through minimizing over-estimations. AI^2 [69] attempts to do abstract interpretation in AI systems, and tries to prove the security and robustness of neural networks. They constructed almost all perturbations, made them propagate automatically, and captured the behavior of convolutional layers, max pooling layers and fully connected layers. They also solved the state space explosion problem. DeepStellar [59] characterizes RNN internal behaviors by modeling a RNN as an abstract state transition system. They design two trace similarity metrics to analyze RNNs quantitatively and also detect AEs with very small perturbations.

The interpretability cannot only bring security, but also uncover the mystery of neural network and make us understand its working mechanism easily. However, this is also beneficial to attackers. They can exclude the range of input proved secure, thus reducing the retrieval space and finding AEs more efficiently. They can also construct targeted attacks through an in-depth understanding on models. In spite of this, this field should not be stagnant. Because a black-box model does not guarantee security [205]. Therefore, with the improvement of interpretability, deep learning security may rise in a zigzag way.

The development of interpretability is also conductive to solving the hysteresis of defensive methods. Since we have not yet achieved a deep understanding of DNN (it is not clear why a record is predicted to the result, and how different data affect model parameters), finding vulnerabilities for attack is easier than preventing in advance. So there is a certain lag in deep learning security. If we can understand models thoroughly, it is believed that defense will precede or synchronize with attack [110] [233] [69].

8.5 Discrimination in AI

AI system may seem rational, neutral and unbiased, but actually, AI and algorithmic decisions can lead to unfair and discrimination [34]. For example, amazon's AI hiring tool taught itself that male candidates were preferable [82]. There are also discrimination in crime prevention, online shops [34], bank loan [5], and so on. There are two main reasons causing AI discrimination [5]: 1) Imbalanced training data; 2)Training data reflects past discrimination.

In order to solve this problem and make AI system better benefit humans, what we need to do is: 1) balancing dataset, by adding/removing data about under/over represented subsets. 2) modifying data or trained model where training data reflects past discrimination [5]; 3) importing testing techniques to test the fairness of models, such as symbolic execution and local interpretability [11]; 4) enacting non-discrimination law, and data protection law, such as GDPR [3].

8.6 Corresponding defense methods

There is a line of approaches for preventing the aforementioned attacks.

Model extraction defense. Blurring the prediction results is an effective way to prevent model stealing, for instance, rounding parameters [226] [220], adding noise into class probabilities [122] [107]. On the other hand, detecting and prevent abnormal queries can also resolve this attack. Kesarwani *et al.* [111] recorded all requests made by clients and calculated the explored feature space to detect attack. PRADA [107] detected attack based on sudden changes in the distribution of samples submitted by a given customer. Orekondy *et al.* [168] proposed an active defense which perturbs predictions targeted at attacking the training objective.

Model inversion defense. To defend with model inversion attacks, researchers propose the following approaches:

- *Differential privacy (DP)*, which is a cryptographic scheme designed to maximize the accuracy of data queries while minimizing the opportunity to identify their records when querying from a statistical database [61]. Individual features are removed to preserve user privacy. It is first proposed in [60] and proved to be effective in privacy preservation in database. In model privacy preserving, DP strategy can be applied to model parameters [180], prediction outputs [46] [83] [229] [269] [97], loss function [112] [214], and gradients [207] [27] [214] [10] [269] [273]. Yu *et al.* [261] propose concentrated DP to analyze and optimize privacy loss.
- *Homomorphic encryption (HE)*, which is an encryption function and enables the following two operations are value-equivalent [191]: exercising arithmetic operations $\oplus$ on the ring of plain text and encrypting the result, encrypting operators first and then carry on the same arithmetic operations, *i.e.*, $En(x) \oplus En(y) = En(x + y)$. In this way, clients can encrypt their data and then send it to MLaaS. The server returns encrypted predictions without learning anything about the plain data. In the meantime, the clients have no idea about the model attributes [70] [136] [108] [105]. BAYHENN [249] uses HE to protect the client data, and uses Bayesian neural network to protect DNN weights, realizing secure DNN inference.
- *Secure multi-party computation (SMC)*, stemming from Yao's Millionaires' problem [257] and enabling a safe calculation of contract functions without trusted third parties. In the context of deep learning, it extends to that multiple parties collectively train a model and preserve their own data [224] [202] [181] [182] [188]. As such, the training data cannot be easily inferred by attackers residing at either computing servers or the client side. Helen [275] is a cooperative learning system that allows multiple parties to train a linear model without revealing data. DCOP [215] can protect privacy under the assumption of an honest majority and is not affected by collusion.
- *Training reconstitution*. Cao *et al.* [42] put forward machine unlearning, which makes ML models completely forget a piece of training data and recover the effects to models and features. Ohrimenko *et al.* [165] proposed a data-oblivious machine learning algorithm. Osia *et al.* [169] broke down large, complex deep models to enable scalable and privacy-preserving analytics by removing sensitive information with a feature extractor. MemGuard [104] adds noise to each confidence score vector predicted by the target classifier. Song *et al.* [206] find adversarial defense methods even increase the risk of target model against membership inference attack.

Poisoning defense. Poisoning attack can be mitigated through two aspects:

- *Protecting data.* This method includes avoiding data tampering, denial and falsification, and detecting poisonous data [234] [145] [84]. Through perturbing inputs, Gao *et al.* [68] observed the randomness of their predicted classes from a given model. The low entropy in predicted classes violates the input dependency property of a benign model and implies the existence of a trojan input. Olufowobi *et al.* [166] described the context of creation or modification of data points to enhance trustworthiness and dependability of the data. Chakarov *et al.* [45] evaluated the effect of individual data points on the performance of trained model. Baracaldo *et al.* [25] used source information of training data points and the transformation context to identify poisonous data.
- *Protecting algorithm.* This method adjusts training algorithms, *e.g.*, robust PCA [40], robust linear regression [49] [132], and robust logistic regression [64]. Wang *et al.* [227] detect poisoning techniques via input filters, neuron pruning and machine unlearning. ABS [140] analyze the changes inside the neurons to detect trojan triggers, when introducing different levels of stimuli to neurons. FABA algorithm [241] can eliminate outliers in the uploaded gradient and obtain a gradient close to the true gradient in distributed learning. Qiao *et al.* [185] explore all possible backdoor triggers space formed by the pixel values and remove the triggers from a backdoored model.

Adversarial defense. As adversarial attack draws the major attention, defensive work is more comprehensive and ample accordingly. The mainstream defense approaches are as follows:

- *Model robustness.* Model robustness means that small perturbations to the input will not cause the network to misclassify. Certified robustness is an effective method to defend against adversarial attack. In order to verify the model robustness, Anderson *et al.* [18] combine the gradient-based optimization method of AEs search with the abstract-based proof search. PixelDP [121] is a certified defense that scales to large networks and datasets. It is based on a connection between robustness against AEs and differential privacy. Ma *et al.* [147] analyze the internal structures of DNN under various attacks, and propose a method of extracting DNN invariants to detect AEs at runtime. Liu *et al.* [133] aim to seek certified adversary-free regions around data points as large as possible. Research [204] proves that adversarial vulnerability of networks increases as gradients, and gradients grow as the input image dimension. PROVEN [237] provides probability certificates of the neural network robustness when the input perturbation obeys the distribution characteristics. For improving the provable error bound, Robustra [125] utilizes the adversarial space to solve the minmax game between attackers and defenders.
- *Adversarial training*. This method selects AEs as part of the training dataset to make trained model learn characteristics of AEs [93] [118] [103] [157]. Furthermore, Ensemble Adversarial Training [218] contained each turbine input transferred from other pre-trained models. Wang *et al.* [228] introduce adversarial noise to the output embedding layer while training neural language models. Ye *et al.* [259] propose a framework for simultaneous adversarial training and weights pruning, which can compress the model while maintaining robustness. Wang *et al.* [232] propose bilateral adversarial training, which both perturbs both the image and the label. Zhang *et al.* [266] generate adversarial images for training by feature scattering in the latent space. Wong *et al.* [238] successfully trained robust models using a weaker and cheaper adversary, which saves much time. Li *et al.* [128] proved that adversarial training indeed promotes robustness through theoretical insights.
- *Region-based method*. Understanding properties of adversarial regions and using more robust region-based classification could also defend adversarial attack. Cao *et al.* [41] develop DNNs using region-based classification instead of point-based. They predicted labels through randomly selecting several points from the hypercube centered at the testing sample. In [171], the classifier mapped normal samples to the neighborhood of low-dimensional manifolds in the final-layer hidden space. Local Intrinsic Dimensionality [148] characterized dimensional properties of adversarial regions and evaluated the spatial fill capability. Background Class [151] added a large and diverse class of background images into datasets.
- *Transformation*. Transforming inputs can defend adversarial attack to a large extent. Song *et al.* [208] found that AEs mainly lay in the low probability regions of the training regions. So they purified an AE by moving it back towards the distribution adaptively. Guo *et al.* [78] explored model-agnostic defenses on image-classification systems by image transformations. Xie *et al.* [247] used randomization at inference time, including random resizing and padding. Tian *et al.* [216] considered that AEs are more sensitive to certain image transformation operations, such as rotation and shifting, than normal images. Wang *et al.* [231] [230] thought AEs are more sensitive to random perturbations than normal. Buckman *et al.* [37] used thermometer code and one-hot code discretization to increase the robustness of network to AEs. Kou *et al.* [115] trained a separate lightweight distribution classifier to recognize different features of transformed images.
- *Gradient regularization/masking*. This method hides gradients or reduces the sensitivity of models. Madry *et al.* [149] realized it by optimizing a saddle point formulation, which included solving an inner maximization solved and an outer minimization. Ross *et al.* [195] trained differentiable models that penalized the degree to infinitesimal changes in inputs.
- *Distillation*. Papernot *et al.* [172] proposed Defensive Distillation, which could successfully mitigate AEs constructed by FGSM and JSMA. Papernot *et al.* [178] also used the knowledge extracted in distillation to reduce the magnitude of network gradient. Liu *et al.* [143] propose feature distillation, a JPEG-based defensive compression framework to rectify AEs.
- *Data preprocessing*. Liang *et al.* [129] introduced scalar quantization and smooth spatial filtering to reduce the effect of perturbations. Zantedeschi *et al.* [264] used bounded ReLU activation function for hedging forward propagation of adversarial perturbation. Xu *et al.* [253] proposed feature squeezing methods, including reducing the depth of color bit on each pixel and spatial smoothing. Yang *et al.* [255] preprocess images by randomly removing pixels from the image, and using matrix estimation to reconstruct it.
- *Defense network*. Some studies use networks to automatically fight against AEs. Gu *et al.* [75] used deep contractive network with contractive autoencoders and denoising autoencoders, which can remove amounts of adversarial noise. Akhtar *et al.* [12] proposed a perturbation rectifying network as pre-input layers to defend against UAPs. MagNet [154] used detector networks to detect AEs which are far from the boundary of manifold, and used a reformer to reform AEs which are close to the boundary. Liu *et al.* [131] propose a defense model which uses feature prioritization of the nonlinear attention module and the L_2 feature regularization.

8.7 Future direction of attack and defense

It is an endless war between attackers and defenders, and neither of them can win an absolute victory. But both sides can research new techniques and applications to gain advantages. From the attacker's point of view, one effective way is to explore new attack surfaces, find out new attack scenarios, seek for new attack purposes and broaden the scope of attack effects. In particular, main attack surfaces on deep learning systems include malformed operational input, malformed training data and malformed models [245].

In adversary attack, L_p-distance is not an ideal measurement. Some images with big perturbations are still indistinguishable for humans. However, unlike L_p-distance, there is no standard measure for large L_p perturbations. This will be a hot point for adversarial learning in future. In model extraction attack, stealing functionality of complex models needs massive queries. How to come up with a better method to reduce the number of queries in order of magnitude will be the focus of this field.

The balance of attack cost and benefit is also an important factor. Some attacks, even can achieve fruitful targets, have to perform costly computation or resources [220]. For example, in [203], the attacker has to train a number of shadow models that simulate the target model, and then undertake membership inference. They need 156 queries to produce a data point on average.

Attack cost and attack benefit are a trade-off process [152]. Generally, the cost of attack contains time, computation resources, acquired knowledge, and monetary expense. The benefit from an attack include economic payback, rivals' failure and so forth. In this study, we will not give a uniform formula to quantify the cost and benefit as the importance of each element is varying in different scenarios. Nevertheless, it is usually modeled as an optimization problem where the cost is minimized while the benefit is maximized, like a min-max game [162].

As for defenders, a combination of multiple defense techniques is a good choice to reduce the risk of being attacked. But the combination may incur additional overhead on the system that should be solved in design. For example, in [136] [108], they adopted a mixed protocol combining HE and MPC, which improved performance but with high bandwidth.

9 Conclusion

In this paper, we conduct a comprehensive and extensive investigation on attacks towards deep learning systems. Different from other surveys, we dissect an attack in a systematical way, where interested readers can clearly understand how these attacks happen step by step. We have compared the investigated works on their attack vectors and proposed a number of metrics to compare their performance. Based on the comparison, we then proceed to distill a number of insights, disclosing advantages and disadvantages of attack methods, limitations and trends. The discussion covering the difficulties of these attacks in the physical world, security concerns in other aspects and potential mitigation for these attacks provide a platform on which future research can be based.

References

[1] Drebin dataset. https:// www. sec. tu- bs. de/ ~danarp/ drebin/, 2016.

[2] Imagenet dataset. http:// www. image- net. org, 2017.

[3] General data protection regulation. https:// gdpr- info. eu, May 2018.

[4] Gtsrb dataset. http:// benchmark. ini. rub. de/? section=gtsrb& subsection=dataset, 2019.

[5] Human bias and discrimination in ai systems. https:// ai-auditingframework. blogspot. com/ 2019/ 06/ human- bias- and- discrimination- in- ai. html, 2019.

[6] Ijb-a dataset. https:// www. nist. gov/ itl/ iad/ image- group/ ijb- dataset- request- form, 2019.

[7] Imdb review dataset. https:// www. kaggle. com/ utathya/ imdb- review- dataset, 2019.

[8] Microsoft kaggle dataset. https:// www. kaggle. com/ c/ microsoft- malware- prediction, 2019.

[9] Mozilla common voice. https:// voice. mozilla. org/ en, 2019.

[10] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS), Vienna, Austria, 2016*, pages 308–318, October 24-28, 2016.

[11] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha. Black box fairness testing of machine learning models. In *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019.*, pages 625–635.

[12] N. Akhtar, J. Liu, and A. S. Mian. Defense against universal adversarial perturbations. *CoRR*, abs/1711.05929, 2017.

[13] N. Akhtar and A. S. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[14] A. Al-Dujaili, A. Huang, E. Hemberg, and U. O'Reilly. Adversarial deep learning for robust detection of binary encoded malware. In *IEEE Security and Privacy Workshops, San Francisco, CA, USA*, pages 76–82, May 24, 2018.

[15] Alexey, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[16] S. Alfeld, X. Zhu, and P. Barford. Data poisoning attacks against autoregressive models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA.*, pages 1452–1458, February 12-17, 2016.

[17] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[18] G. Anderson, S. Pailoor, I. Dillig, and S. Chaudhuri. Optimization and abstraction: a synergistic approach for analyzing neural network robustness. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2019, Phoenix, AZ, USA, June 22-26, 2019*, pages 731–744.

[19] 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. *IJSN*, 10(3):137–150, 2015.

[20] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholmsmässan, Stockholm, Sweden*, pages 284–293, July 10-15, 2018.

[21] A. Awad, J. Traub, and S. Sakr. Adaptive watermarks: A concept drift-based approach for predicting event-time progress in data streams. In *22nd International Conference on Extending Database Technology (EDBT), Lisbon, Portugal*, pages 622–625, March 26-29, 2019.

[22] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, and S. Yoon. Security and privacy issues in deep learning. *CoRR*, abs/1807.11655, 2018.

[23] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, pages 2938–2948.

[24] S. Baluja and I. Fischer. Learning to attack: Adversarial transformation networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*, February 2-7, 2018.

[25] 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, AISec@CCS, Dallas, TX, USA*, pages 103–110, November 3, 2017.

[26] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, Nov 2010.

[27] R. Bassily, A. D. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS, Philadelphia, PA, USA*, pages 464–473, October 18-21, 2014.

[28] V. Beal. What is structured data? webopedia definition. https:// www. webopedia. com/ TERM/ S/ structured_ data. html, Aug. 2018.

[29] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. B. Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 634–643.

[30] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth. Unrestricted adversarial examples via semantic manipulation. In *International Conference on Learning Representations*, 2020.

[31] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML, Edinburgh, Scotland, UK*, June 26 - July 1, 2012.

[32] B. Biggio, I. Pillai, S. R. Bulò, D. Ariu, M. Pelillo, and F. Roli. Is data clustering in adversarial settings secure? In *AISec'13, Proceedings of the ACM Workshop on Artificial Intelligence and Security, Co-located with CCS, Berlin, Germany*, pages 87–98, November 4, 2013.

[33] A. Bojchevski and S. Günnemann. Adversarial attacks on node embeddings via graph poisoning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 695–704.

[34] F. Z. Borgesius. Discrimination, artificial intelligence, and algorithmic decision-making. https:// rm. coe. int/ discrimination- artificial- intelligence- and- algorithmic- decision- making/ 1680925d73, 2018.

[35] M. Brückner and T. Scheffer. Nash equilibria of static prediction games. In *23rd Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 171–179, December 7-10, 2009.

[36] S. Bubeck, Y. T. Lee, E. Price, and I. P. Razenshteyn. Adversarial examples from computational constraints. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 831–840.

[37] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

[38] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT, New York, NY, USA*, pages 77–91, February 23-24, 2018.

[39] C. Burkard and B. Lagesse. Analysis of causative attacks against svms learning from data streams. In *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics, IWSPA@CODASPY, Scottsdale, Arizona, USA*, pages 31–36, March 24, 2017.

[40] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011.

[41] 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, Orlando, FL, USA*, pages 278–287, December 4-8, 2017.

[42] Y. Cao and J. Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy, SP, San Jose, CA, USA*, pages 463–480, May 17-21, 2015.

[43] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[44] N. Carlini and D. A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE Security and Privacy Workshops, SP Workshops, San Francisco, CA, USA*, pages 1–7, May 24, 2018.

[45] A. Chakarov, A. V. Nori, S. K. Rajamani, S. Sen, and D. Vijaykeerthy. Debugging machine learning tasks. *CoRR*, abs/1603.07292, 2016.

[46] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 289–296, December 8-11, 2008.

[47] H. Chen, H. Zhang, D. S. Boning, and C. Hsieh. Robust decision trees against adversarial examples. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1122–1131.

[48] P. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*, pages 10–17, February 2-7, 2018.

[49] Y. Chen, C. Caramanis, and S. Mannor. Robust high dimensional sparse regression and matching pursuit. *CoRR*, abs/1301.2725, 2013.

[50] M. Cheng, T. Le, P. Chen, H. Zhang, J. Yi, and C. Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *7th International Conference on Learning Representations, ICLR, New Orleans, LA, USA*, May 6-9, 2019.

[51] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu. Improving black-box adversarial attacks with a transfer-based prior. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 10932–10942.

[52] Y. Cheng, L. Jiang, and W. Macherey. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 4324–4333.

[53] K. T. Co, L. Muñoz-González, S. de Maupeou, and E. C. Lupu. Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 275–289.

[54] J. R. C. da Silva, R. F. Berriel, C. Badue, A. F. de Souza, and T. Oliveira-Santos. Copycat CNN: stealing knowledge by persuading confession with random non-labeled data. In *International Joint Conference on Neural Networks, IJCNN, Rio de Janeiro, Brazil*, pages 1–8, July 8-13, 2018.

[55] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

[56] E. Dohmatob. Generalized no free lunch theorem for adversarial robustness. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1646–1654.

[57] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Salt Lake City, UT, USA*, pages 9185–9193, June 18-22, 2018.

[58] Y. Dong, T. Pang, H. Su, and J. Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4312–4321.

[59] X. Du, X. Xie, Y. Li, L. Ma, Y. Liu, and J. Zhao. Deepstellar: model-based quantitative analysis of stateful deep learning systems. In *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019.*, pages 477–487.

[60] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia*, pages 486–503, May 28-June 1, 2006.

[61] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC, New York, NY, USA*, pages 265–284, March 4-7, 2006.

[62] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. Exploring the landscape of spatial robustness. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1802–1811.

[63] M. Fang, X. Cao, J. Jia, and N. Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1605–1622. USENIX Association, Aug. 2020.

[64] J. Feng, H. Xu, S. Mannor, and S. Yan. Robust logistic regression and classification. In *Annual Conference on Neural Information Processing Systems, Montreal, Quebec, Canada*, pages 253–261, December 8-13, 2014.

[65] 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, Denver, CO, USA*, pages 1322–1333, October 12-16, 2015.

[66] 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 ACM SIGSAC Conference on Computer and Communications Security, CCS, Toronto, ON, Canada*, pages 619–633, October 15-19, 2018.

[67] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *IEEE Security and Privacy Workshops, SP Workshops, San Francisco, CA, USA*, pages 50–56, May 24, 2018.

[68] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal. STRIP: a defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC 2019, San Juan, PR, USA, December 09-13, 2019*, pages 113–125.

[69] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. T. Vechev. AI2: safety and robustness certification of neural networks with abstract interpretation. In *IEEE Symposium on Security and Privacy, SP, San Francisco, CA, USA*, pages 3–18, May 21-23, 2018.

[70] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 201–210, 2016.

[71] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

[72] Y. Gong, B. Li, C. Poellabauer, and Y. Shi. Real-time adversarial attacks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4672–4680.

[73] Y. Gong and C. Poellabauer. Crafting adversarial examples for speech paralinguistics applications. *CoRR*, abs/1711.03280, 2017.

[74] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[75] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *CoRR*, abs/1412.5068, 2014.

[76] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[77] C. Guo, J. S. Frank, and K. Q. Weinberger. Low frequency adversarial perturbation. *CoRR*, abs/1809.08758, 2018.

[78] C. Guo, M. Rana, M. Cissé, and L. van der Maaten. Countering adversarial images using input transformations. *CoRR*, abs/1711.00117, 2017.

[79] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun. Dlfuzz: differential fuzzing testing of deep learning systems. In *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2018, Lake Buena Vista, FL, USA, November 04-09, 2018*, pages 739–743.

[80] Y. Guo, Z. Yan, and C. Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3820–3829.

[81] I. Hagestedt, Y. Zhang, M. Humbert, P. Berrang, H. Tang, X. Wang, and M. Backes. Mbeacon: Privacy-preserving beacons for DNA methylation data. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*.

[82] I. A. Hamilton. Amazon built an ai tool to hire people but had to shut it down because it was discriminating against women. https:// www. businessinsider. com/ amazon- built- ai- to- hire- people- discriminated- against- women- 2018- 10, Oct. 2018.

[83] J. Hamm, Y. Cao, and M. Belkin. Learning privately from multiparty data. In *Proceedings of the 33nd International Conference on Machine Learning, ICML, New York City, NY, USA*, pages 555–563, June 19-24, 2016.

[84] R. Hasan, R. Sion, and M. Winslett. The case of the fake picasso: Preventing history forgery with secure provenance. In *7th USENIX Conference on File and Storage Technologies, February 24-27, 2009, San Francisco, CA, USA. Proceedings*, pages 1–14, 2009.

[85] J. Hayes and G. Danezis. Learning universal adversarial perturbations with generative models. In *IEEE Security and Privacy Workshops, SP Workshops, San Francisco, CA, USA*, pages 43–49, May 24, 2018.

[86] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro. LOGAN: evaluating privacy leakage of generative models using generative adversarial networks. *CoRR*, abs/1705.07663, 2017.

[87] W. He, B. Li, and D. Song. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*, 2018.

[88] Z. He, T. Zhang, and R. B. Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC 2019, San Juan, PR, USA, December 09-13, 2019*.

[89] B. Hitaj, G. Ateniese, and F. Pérez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, Dallas, TX, USA*, pages 603–618, October 30-November 03, 2017.

[90] W. Hu and Y. Tan. Black-box attacks against RNN based malware detection algorithms. In *The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*, pages 245–251, February 2-7,.

[91] W. Hu and Y. Tan. Generating adversarial malware examples for black-box attacks based on GAN. *CoRR*, abs/1702.05983, 2017.

[92] W. Hua, Z. Zhang, and G. E. Suh. Reverse engineering convolutional neural networks through side-channel information leaks. In *Proceedings of the 55th Annual Design Automation Conference, DAC, San Francisco, CA, USA*, pages 4:1–4:6, June 24-29, 2018.

[93] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári. Learning with a strong adversary. *CoRR*, abs/1511.03034, 2015.

[94] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel. Adversarial attacks on neural network policies. *CoRR*, abs/1702.02284, 2017.

[95] W. Huang and J. W. Stokes. Mtnet: A multi-task neural network for dynamic malware classification. In *13th International Conference, Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA, San Sebastián, Spain*, pages 399–418, July 7-8, 2016.

[96] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Comput. Sci. Rev.*, 37:100270, 2020.

[97] N. Hynes, R. Cheng, and D. Song. Efficient deep learning on multi-source private data. *CoRR*, abs/1807.06689, 2018.

[98] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Query-efficient black-box adversarial examples. *CoRR*, abs/1712.07113, 2017.

[99] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholmsmässan, Stockholm, Sweden*, pages 2142–2151, 2018.

[100] 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 *IEEE Symposium on Security and Privacy, SP, San Francisco, California, USA*, pages 19–35, May 21-23, 2018.

[101] S. Jan, A. Panichella, A. Arcuri, and L. C. Briand. Automatic generation of tests to exploit XML injection vulnerabilities in web applications. *IEEE Trans. Software Eng.*, 45(4):335–362, 2019.

[102] 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, Orlando, FL, USA, December 4-8, 2017*, pages 262–277.

[103] Y. Jang, T. Zhao, S. Hong, and H. Lee. Adversarial defense via learning to generate diverse attacks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2740–2749.

[104] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 259–274.

[105] X. Jiang, M. Kim, K. E. 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, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 1209–1222, 2018.

[106] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, April 2017.

[107] M. Juuti, S. Szyller, A. Dmitrenko, S. Marchal, and N. Asokan. PRADA: protecting against DNN model stealing attacks. *CoRR*, abs/1805.02628, 2018.

[108] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium, USENIX Security, Baltimore, MD, USA*, pages 1651–1669, August 15-17, 2018.

[109] A. Kantchelian, S. Afroz, L. Huang, A. C. Islam, B. Miller, M. C. Tschantz, R. Greenstadt, A. D. Joseph, and J. D. Tygar. Approaches to adversarial drift. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security, AISec, Berlin, Germany*, pages 99–110, November 4, 2013.

[110] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Towards proving the adversarial robustness of deep neural networks. In *Proceedings First Workshop on Formal Verification of Autonomous Vehicles, FVAV@iFM, Turin, Italy, 19th September.*, pages 19–26, 2017.

[111] M. Kesarwani, B. Mukhoty, V. Arya, and S. Mehta. Model extraction warning in mlaas paradigm. *CoRR*, abs/1711.07221, 2017.

[112] D. Kifer, A. D. Smith, and A. Thakurta. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. In *The 25th Annual Conference on Learning Theory, COLT, Edinburgh, Scotland*, pages 25.1–25.40, June 25-27, 2012.

[113] J. Kim, R. Feldt, and S. Yoo. Guiding deep learning system testing using surprise adequacy. In *Proceedings of the 41st International Conference on Software Engineering, ICSE 2019, Montreal, QC, Canada, May 25-31, 2019*, pages 1039–1049.

[114] J. Kos, I. Fischer, and D. Song. Adversarial examples for generative models. In *IEEE Security and Privacy Workshops, SP Workshops, San Francisco, CA, USA, May 24, 2018*, pages 36–42.

[115] C. Kou, H. K. Lee, E.-C. Chang, and T. K. Ng. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In *International Conference on Learning Representations*, 2020.

[116] 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. 2018.

[117] A. Krizhevsky, V. Nair, and G. Hinton. CIFAR dataset. https:// www. cs. toronto. edu/ ~kriz/ cifar. html, 2019.

[118] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016.

[119] K. Kurita, P. Michel, and G. Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2793–2806.

[120] Y. LeCun, C. Cortes, and C. Burges. Mnist dataset. http:// yann. lecun. com/ exdb/ mnist/, 2017.

[121] M. Lécuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672.

[122] T. Lee, B. Edwards, I. Molloy, and D. Su. Defending against model stealing attacks using deceptive perturbations. *CoRR*, abs/1806.00054, 2018.

[123] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019.*

[124] J. Li, F. R. Schmidt, and J. Z. Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3896–3904.

[125] L. Li, Z. Zhong, B. Li, and T. Xie. Robustra: Training provable robust neural networks over reference adversarial space. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4711–4717.

[126] P. Li, J. Yi, and L. Zhang. Query-efficient black-box attack by active learning. In *IEEE International Conference on Data Mining, ICDM , Singapore, November 17-20, 2018*, pages 1200–1205.

[127] S. Li, A. Neupane, S. Paul, C. Song, S. V. Krishnamurthy, A. K. Roy-Chowdhury, and A. Swami. Stealthy adversarial perturbations against real-time video classification systems. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019.*

[128] Y. Li, E. X.Fang, H. Xu, and T. Zhao. Implicit bias of gradient descent based adversarial training on separable data. In *International Conference on Learning Representations*, 2020.

[129] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang. Detecting adversarial image examples in deep networks with adaptive noise reduction. 2017.

[130] R. Light. Ai trends: Machine learning as a service (mlaas). https:// learn. g2. com/ trends/ machine- learning- service- mlaas, Jan. 2018.

[131] C. Liu and J. JáJá. Feature prioritization and regularization improve standard accuracy and adversarial robustness. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2994–3000.

[132] 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, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 91–102, 2017.

[133] C. Liu, R. Tomioka, and V. Cevher. On certifying non-uniform bounds against adversarial attacks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 4072–4081.

[134] F. Liu and N. B. Shroff. Data poisoning attacks on stochastic bandits. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 4042–4050.

[135] H. Liu, R. Ji, J. Li, B. Zhang, Y. Gao, Y. Wu, and F. Huang. Universal adversarial perturbation via prior driven uncertainty approximation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2941–2949.

[136] J. Liu, M. Juuti, Y. Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, Dallas, TX, USA, October 30 - November 03, 2017*, pages 619–631.

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

[138] 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*, 6:12103–12117, 2018.

[139] X. Liu, S. Si, J. Zhu, Y. Li, and C. Hsieh. A unified framework for data poisoning attack to graph-based semi-supervised learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 9777–9787.

[140] Y. Liu, W. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang. ABS: scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 1265–1282.

[141] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018.

[142] Y. Liu, S. Moosavi-Dezfooli, and P. Frossard. A geometry-inspired decision-based attack. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4889–4897.

[143] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen. Feature distillation: Dnn-oriented JPEG compression against adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 860–868.

[144] 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. *CoRR*, abs/1802.04889, 2018.

[145] J. Lyle and A. P. Martin. Trusted computing and provenance: Better together. In *2nd Workshop on the Theory and Practice of Provenance, TaPP'10, San Jose, CA, USA, February 22, 2010*, 2010.

[146] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang. Deepgauge: multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, Montpellier, France, September 3-7, 2018*, pages 120–131.

[147] S. Ma, Y. Liu, G. Tao, W. Lee, and X. Zhang. NIC: detecting adversarial samples with neural network invariant checking. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019.*

[148] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. N. R. Wijewickrema, M. E. Houle, G. Schoenebeck, D. Song, and J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *CoRR*, abs/1801.02613, 2018.

[149] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017.

[150] S. Mahloujifar, M. Mahmoody, and A. Mohammed. Data poisoning attacks in multi-party learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 4274–4283.

[151] M. McCoyd and D. A. Wagner. Background class defense against adversarial examples. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 96–102, 2018.

[152] S. Mei and X. Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2871–2877, 2015.

[153] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 691–706.

[154] D. Meng and H. Chen. Magnet: A two-pronged defense against adversarial examples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS, Dallas, TX, USA, October 30 - November 03, 2017*, pages 135–147.

[155] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 86–94, 2017.

[156] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582.

[157] C. K. Mummadi, T. Brox, and J. H. Metzen. Defending against universal perturbations with shared adversarial training. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4927–4936.

[158] 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, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 27–38, 2017.

[159] N. Narodytska and S. P. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Honolulu, HI, USA, July 21-26, 2017*, pages 1310–1318.

[160] M. Naseer, S. H. Khan, M. H. Khan, F. S. Khan, and F. Porikli. Cross-domain transferability of adversarial perturbations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 12885–12895.

[161] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 739–753.

[162] M. Nasr, R. Shokri, and A. Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 634–646, 2018.

[163] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. A. Sutton, J. D. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET*, 2008.

[164] S. J. Oh, M. Augustin, M. Fritz, and B. Schiele. Towards reverse-engineering black-box neural networks. In *International Conference on Learning Representations*, 2018.

[165] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa. Oblivious multi-party machine learning on trusted processors. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 619–636, 2016.

[166] 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 *Service-Oriented Computing, ICSOC 2016 Workshops, Banff, AB, Canada, October 10-13, 2016.*, pages 85–91.

[167] T. Orekondy, B. Schiele, and M. Fritz. Knockoff nets: Stealing functionality of black-box models. June 2019.

[168] T. Orekondy, B. Schiele, and M. Fritz. Prediction poisoning: Towards defenses against dnn model stealing attacks. In *International Conference on Learning Representations*, 2020.

[169] S. A. Ossia, A. S. Shamsabadi, A. Taheri, H. R. Rabiee, N. D. Lane, and H. Haddadi. A Hybrid Deep Learning Architecture for Privacy-Preserving Mobile Analytics. *CoRR*, abs/1703.02952, 2017.

[170] R. Pan. Static deep neural network analysis for robustness. In *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019.*, pages 1238–1240.

[171] T. Pang, C. Du, and J. Zhu. Robust deep learning via reverse cross-entropy training and thresholding test. *CoRR*, abs/1706.00633, 2017.

[172] N. Papernot and P. D. McDaniel. On the effectiveness of defensive distillation. *CoRR*, abs/1607.05113, 2016.

[173] N. Papernot, P. D. McDaniel, and I. J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

[174] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519.

[175] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387, 2016.

[176] N. Papernot, P. D. 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, London, United Kingdom, April 24-26, 2018*, pages 399–414, 2018.

[177] N. Papernot, P. D. McDaniel, A. Swami, and R. E. Harang. Crafting adversarial input sequences for recurrent neural networks. In *2016 IEEE Military Communications Conference, MILCOM 2016, Baltimore, MD, USA, November 1-3, 2016*, pages 49–54, 2016.

[178] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 582–597, 2016.

[179] 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 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 1916–1920, 2015.

[180] N. Phan, M. N. Vu, Y. Liu, R. Jin, D. Dou, X. Wu, and M. T. Thai. Heterogeneous gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4753–4759.

[181] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Information Forensics and Security*, 13(5):1333–1345, 2018.

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

[183] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, 51(5):92:1–92:36, Sept. 2018.

[184] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro. Knock knock, who's there? membership inference on aggregate location data. 2017.

[185] X. Qiao, Y. Yang, and H. Li. Defending neural backdoors via generative distribution modeling. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 14004–14013.

[186] Y. Qin, N. Carlini, G. W. Cottrell, I. J. Goodfellow, and C. Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5231–5240.

[187] A. Ranjan, J. Janai, A. Geiger, and M. J. Black. Attacking optical flow. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2404–2413.

[188] D. Reich, A. Todoki, R. Dowsley, M. D. Cock, and A. C. A. Nascimento. Privacy-preserving classification of personal text messages with secure multi-party computation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3752–3764.

[189] S. Ren, Y. Deng, K. He, and W. Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 1085–1097.

[190] 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, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 70–75, 2018.

[191] R. L. Rivest, L. Adleman, M. L. Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[192] R. Romagnoli, S. Weerakkody, and B. Sinopoli. A model inversion based watermark for replay attack detection with output tracking. In *2019 American Control Conference, ACC 2019, Philadelphia, PA, USA, July 10-12, 2019*, pages 384–390.

[193] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach. Low resource black-box end-to-end attack against state of the art API call based malware classifiers. *CoRR*, abs/1804.08778, 2018.

[194] I. Rosenberg, A. Shabtai, L. Rokach, and Y. Elovici. Generic black-box end-to-end attack against state of the art API call based malware classifiers. In *21st International Symposium, Research in Attacks, Intrusions, and Defenses, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings*, pages 490–510.

[195] A. S. Ross and F. Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 1660–1669, 2018.

[196] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5558–5567.

[197] A. Salem, A. Bhattacharya, M. Backes, M. Fritz, and Y. Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1291–1308. USENIX Association, Aug. 2020.

[198] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *CoRR*, abs/1806.01246, 2018.

[199] M. Sato, J. Suzuki, and S. Kiyono. Effective adversarial regularization for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 204–210.

[200] 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. *CoRR*, abs/1804.00792, 2018.

[201] M. Sharif, L. Bauer, and M. K. Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1605–1613.

[202] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 1310–1321, 2015.

[203] 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 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18, 2017.

[204] C. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, and D. Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5809–5817.

[205] C. Song, T. Ristenpart, and V. Shmatikov. Machine learning models that remember too much. In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 587–601.

[206] L. Song, R. Shokri, and P. Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 241–257.

[207] S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013, Austin, TX, USA, December 3-5, 2013*, pages 245–248, 2013.

[208] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *CoRR*, abs/1710.10766, 2017.

[209] A. Stasinopoulos, C. Ntantogian, and C. Xenakis. Commix: automating evaluation and exploitation of command injection vulnerabilities in web applications. *Int. J. Inf. Sec.*, 18(1):49–72, 2019.

[210] J. Steinhardt, P. W. Koh, and P. S. Liang. Certified defenses for data poisoning attacks. In *Annual Conference on Neural Information Processing Systems, 4-9 December 2017, Long Beach, CA, USA*, pages 3520–3532, 2017.

[211] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, and D. Kroening. Concolic testing for deep neural networks. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, Montpellier, France, September 3-7, 2018*, pages 109–119.

[212] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan. Can you really backdoor federated learning? *CoRR*, abs/1911.07963, 2019.

[213] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[214] K. Talwar, A. Thakurta, and L. Zhang. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *CoRR*, abs/1411.5417, 2014.

[215] T. Tassa, T. Grinshpoun, and A. Yanai. A privacy preserving collusion secure DCOP algorithm. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4774–4780.

[216] S. Tian, G. Yang, and Y. Cai. Detecting adversarial examples through image transformation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4139–4146, 2018.

[217] Y. Tian, K. Pei, S. Jana, and B. Ray. Deeptest: automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*, pages 303–314.

[218] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. *CoRR*, abs/1705.07204, 2017.

[219] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. The space of transferable adversarial examples. *CoRR*, abs/1704.03453, 2017.

[220] 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, Austin, TX, USA, August 10-12, 2016.*, pages 601–618, 2016.

[221] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Towards demystifying membership inference attacks. *CoRR*, abs/1807.09173, 2018.

[222] M. Veale, R. Binns, and L. Edwards. Algorithms that remember: Model inversion attacks and data protection law. *CoRR*, abs/1807.04644, 2018.

[223] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

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

[225] B. Wang and N. Z. Gong. Attacking graph-based classification via manipulating the graph structure. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2023–2040.

[226] B. Wang and N. Z. Gong. Stealing hyperparameters in machine learning. In *IEEE Symposium on Security and Privacy (SP), San Francisco, California, USA, 21-23 May 2018*, pages 36–52, 2018.

[227] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 707–723.

[228] D. Wang, C. Gong, and Q. Liu. Improving neural language modeling via adversarial training. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6555–6565.

[229] D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4-9 December 2017*, pages 2719–2728, 2017.

[230] J. Wang, G. Dong, J. Sun, X. Wang, and P. Zhang. Adversarial sample detection for deep neural network through model mutation testing. In *Proceedings of the 41st International Conference on Software Engineering, ICSE 2019, Montreal, QC, Canada, May 25-31, 2019*, pages 1245–1256.

[231] J. Wang, J. Sun, P. Zhang, and X. Wang. Detecting adversarial samples for deep neural networks through mutation testing. *CoRR*, abs/1805.05010, 2018.

[232] J. Wang and H. Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6628–6637.

[233] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana. Formal security analysis of neural networks using symbolic intervals. In *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018.*, pages 1599–1614, 2018.

[234] X. O. Wang, K. Zeng, K. Govindan, and P. Mohapatra. Chaining for securing data provenance in distributed information networks. In *31st IEEE Military Communications Conference, MILCOM, Orlando, FL, USA, October 29 - November 1, 2012*, pages 1–6.

[235] Y. Wang and K. Chaudhuri. Data poisoning attacks against online learning. *CoRR*, abs/1808.08994, 2018.

[236] X. Wei, S. Liang, N. Chen, and X. Cao. Transferable adversarial attacks for image and video object detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 954–960.

[237] L. Weng, P. Chen, L. M. Nguyen, M. S. Squillante, A. Boopathy, I. V. Oseledets, and L. Daniel. PROVEN: verifying robustness of neural networks with a probabilistic approach. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6727–6736.

[238] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

[239] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4816–4823.

[240] Z. Xi, R. Pang, S. Ji, and T. Wang. Graph backdoor. *CoRR*, abs/2006.11890, 2020.

[241] Q. Xia, Z. Tao, Z. Hao, and Q. Li. FABA: an algorithm for fast aggregation against byzantine attacks in distributed neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4824–4830.

[242] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1689–1698, 2015.

[243] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

[244] H. Xiao, H. Xiao, and C. Eckert. Adversarial label flips attack on support vector machines. In *20th European Conference on Artificial Intelligence (ECAI), Montpellier, France, August 27-31, 2012*, pages 870–875, 2012.

[245] Q. Xiao, K. Li, D. Zhang, and W. Xu. Security risks in deep learning implementations. In *2018 IEEE Security and Privacy (SP) Workshops, San Francisco, CA, USA, May 24, 2018*, pages 123–128.

[246] C. Xie, K. Huang, P. Chen, and B. Li. DBA: distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[247] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille. Mitigating adversarial effects through randomization. *CoRR*, abs/1711.01991, 2017.

[248] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2730–2739.

[249] P. Xie, B. Wu, and G. Sun. BAYHENN: combining bayesian deep learning and homomorphic encryption for secure DNN inference. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4831–4837.

[250] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See. Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2019, Beijing, China, July 15-19, 2019.*, pages 146–157.

[251] K. Xu, H. Chen, S. Liu, P. Chen, T. Weng, M. Hong, and X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3961–3967.

[252] L. Xu, W. Jia, W. Dong, and Y. Li. Automatic exploit generation for buffer overflow vulnerabilities. In *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS) Companion, Lisbon, Portugal, July 16-20, 2018*, pages 463–468.

[253] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018.

[254] H. Yakura and J. Sakuma. Robust audio adversarial example for a physical attack. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5334–5341.

[255] Y. Yang, G. Zhang, Z. Xu, and D. Katabi. Me-net: Towards effective adversarial robustness with matrix estimation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 7025–7034.

[256] Z. Yang, J. Zhang, E. Chang, and Z. Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 225–240.

[257] A. C. Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982.

[258] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2041–2055.

[259] S. Ye, X. Lin, K. Xu, S. Liu, H. Cheng, J. Lambrechts, H. Zhang, A. Zhou, K. Ma, and Y. Wang. Adversarial robustness vs. model compression, or both? In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 111–120.

[260] W. You, P. Zong, K. Chen, X. Wang, X. Liao, P. Bian, and B. Liang. Semfuzz: Semantics-based automatic generation of proof-of-concept exploits. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), Dallas, TX, USA, October 30 - November 03, 2017*, pages 2139–2154.

[261] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 332–349, 2019.

[262] 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 2018, Baltimore, MD, USA, August 15-17, 2018.*, pages 49–64, 2018.

[263] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, and M. Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Asso-*

ciation for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 6066–6080.

[264] V. Zantedeschi, M. Nicolae, and A. Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 39–49, 2017.

[265] X. Zeng, C. Liu, Y. Wang, W. Qiu, L. Xie, Y. Tai, C. Tang, and A. L. Yuille. Adversarial attacks beyond the image space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4302–4311.

[266] H. Zhang and J. Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1829–1839.

[267] H. Zhang, T. Zheng, J. Gao, C. Miao, L. Su, Y. Li, and K. Ren. Data poisoning attack against knowledge graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4853–4859.

[268] H. Zhang, H. Zhou, N. Miao, and L. Li. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 5564–5569.

[269] J. Zhang, K. Zheng, W. Mou, and L. Wang. Efficient private ERM for smooth objectives. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3922–3928, 2017.

[270] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, PP(99):1–1, 2020.

[271] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, Montpellier, France, September 3-7, 2018*, pages 132–142.

[272] T. Zhang, Z. He, and R. B. Lee. Privacy-preserving machine learning through data obfuscation. *CoRR*, abs/1807.01860, 2018.

[273] T. Zhang and Q. Zhu. A dual perturbation approach for differential private admm-based distributed empirical risk minimization. In *Proceedings of ACM Workshop on Artificial Intelligence and Security (AISec), Vienna, Austria, October 28, 2016*, pages 129–137.

[274] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 1989–2004.

[275] W. Zheng, R. A. Popa, J. E. Gonzalez, and I. Stoica. Helen: Maliciously secure coopetitive learning for linear models. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 724–738.

[276] W. Zou, S. Huang, J. Xie, X. Dai, and J. Chen. A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3486–3497.

[277] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6246–6250.