Federal Office
for Information Security

Security of AI-Systems: Fundamentals

Adversarial Deep Learning

Version	Date	Editor	Description
$0.\bar{9}$	14.06.2022	S. Jacob	initial Version

Contents

Introduction

Over the past years, Artificial Intelligence (AI) systems based on Deep Learning (DL) models, such as Neural Networks (NNs), have become a part of our daily lives: voice assistants understand spoken conversations, autonomous systems navigate in the physical world, and medical AI supports the diagnosis of diseases. With the widespread adoption of such systems, questions about their reliability and safety have gained importance, in particular in light of new threat models and attack vectors that are specific to DL. This field of research is subsumed under the more general term adversarial machine learning. Among these threats are evasion attacks, i.e., specifically crafted inputs that shift the model's output, poisoning and backdoor attacks, i.e., weaknesses implanted in the model, and privacy attacks, which extract information from the model.

The more AI systems are deployed in critical areas, the higher the damage when successful attacks are carried out on the system. Some design choices promote attacks, e.g., when heavily relying on external sources. The effort and resources needed for collecting and curating in-house data sets, training big models from scratch, as well as finding new architectures are some of the aspects that make reusing pretrained models, existing architectures, and publicly available data sets attractive alternatives. However, this can increase chances for mounting successful DL-specific attacks like poisoning, e.g., by distributing backdoored trained models, inserting poisoned samples into the training data, or white-box adversarial attacks. Moreover, the limited theoretical understanding of deep NNs may lead to using overfitted models that memorized training data, which increases the chances of successful attacks extracting private information from the models. In turn, this can facilitate more effective evasion attacks. The interconnections between the individual threats show the importance of generally applicable defense strategies.

In light of the existing threats, attempts to safeguard and ultimately certify AI systems gain importance and are in the focus of research, society, and standardization bodies. Note that methods from software testing cannot be applied directly to test an AI system; and IT security methods alone cannot fully protect against the attacks mentioned. Therefore, new techniques are needed to address such Machine Learning (ML)-specific vulnerabilities. In this document, we discuss adversarial robustness, i.e., when measuring an AI's resilience against evasion (adversarial) attacks, and model resilience to poisoning and privacy attacks.

We present best practice guidelines for certification and verification of NNs, as well as defense techniques against evasion, poisoning, backdoor, and privacy attacks. Moreover, we provide readers with a broad literature study of the aforementioned fields, enabling them to navigate these broad and fast-paced fields of research. This includes a preliminary analysis of some of the reviewed methods, giving practitioners an overview for their implementation as well as outlining a possible empirical framework to tackle research questions derived from the literature review.

The mentioned attacks can happen in the digital space or in real-world scenarios (e.g., printed patterns). Based on the amount and popularity of research we focus on defenses protecting against attacks in the digital space. Our goal is to raise awareness for threats and attack vectors on AI systems. Moreover, we identify limitations and open problems as well as unmitigated risks and shortcomings in the current research related to security in DL applications. In particular, we lay the foundation for a constantly adaptable set of guidelines, which can be extended with latest advances in research. This is crucial given the constant race between attacks and defenses and the fast development in adversarial ML. The guidelines and considerations in this document can also serve as basis for further work in the field of general AI system certification and regulation approaches for trustworthy and reliable AI [90, 89, 343]. We enable the reader to perform an in depth analysis of the approaches described in the aforementioned best practice guidelines and to evaluate their applicability and suitability by listing and summarizing the relevant literature and describing an extendable implementation pipeline.

With the amount of research presented in the area of adversarial ML, it is difficult for stakeholders to select and configure defenses optimally protecting their systems. This even more applies to possible future attacks. The presented guidelines solely focus on currently existing defense methods. Hence, newly introduced attacks may pose a threat to the presented best practices and proposed defenses. We therefore encourage the reader to continuously reassess the presented contents with respect to developments in the field. Based on the popularity of research focusing on image-processing models, in this document we mainly focus on concepts presented and tested in this domain. Adapting techniques for different data sets and domains is a challenging task, as we describe in more detail in Section 1.3 and Chapter 3.

The remainder of the document is structured as follows: First, we describe the best practice guideline to protect DL-based AI applications from evasion, poisoning, backdoor, and privacy attacks in Chapter 1. This guideline relies on the AI life cycle and raises awareness for attack vectors at each of the stages. We conclude the best practice guidelines by giving a detailed description of limitations and open problems, outlining unmitigated risks. In Chapter 2 we present a detailed overview of the literature on the topic of adversarial ML, summarizing relevant papers according to our developed taxonomy. In Chapter 3, we present an experimental framework and outline its use to address research questions, in particular on the intersection of the different fields identified in the literature overview.

Acronyms

ADMM Alternating Direction Method of Multipliers

AI Artificial Intelligence

AMLS Adaptive Multi-Level Splitting

AMP Automatic Mixed Precision

BaB Branch-and-Bound

CNN Convolutional Neural Network

DAG Directed Acyclic Graph

DDS Dataset Definition Standard

DL Deep Learning

DNN Deep Neural Network

DNNV Deep Neural Network Verification

DRS Dataset Requirements Standard

DVC Data Version Control

DVP Dataset Verification Plan

FGP Fast Geometric Projections

FGSM Fast Gradient Sign Method

FNN Feedforward Neural Network

IBP Interval Bound Propagation

LMI Linear Matrix Inequality

LP Linear Programming

LSTM Long Short-Term Memory

MILP Mixed-Integer Linear Programming

ML Machine Learning

NLP Natural Language Processing

NN Neural Network

ODD Operational Design Domain

ONNX Open Neural Network Exchange

ReLU Rectifier Linear Unit

RNN Recurrent Neural Network

SAT Satisfiability

SMC Satisfiability Modulo Convex

SMT Satisfiability Modulo Theories

Chapter 1

Best Practices

L. Adilova, *Fraunhofer IAIS*
Dr. K. Böttinger, *Fraunhofer AISEC*
V. Danos, *TÜViT*
S. Jacob, *BSI*
F. Langer, *TÜViT*
T. Markert, *TÜViT*
Dr. M. Poretschkin, *Fraunhofer IAIS*
J. Rosenzweig, *Fraunhofer IAIS*
J.-P. Schulze, *Fraunhofer AISEC*
P. Sperl, *Fraunhofer AISEC*

In this chapter, we give engineers in industry and research as well as users involved in designing, training, testing, and deploying AI systems a guideline to increase their systems' reliability. Throughout this chapter, we adopt a DL-specific attack-oriented view, i.e., we consider intentionally harmful actions against DL parts of a given AI system. As a result, natural perturbations or "difficult" examples, which were not specifically constructed to fool a model, are not investigated here. Nonetheless, we recommend to analyze the robustness against such examples separately. Also, our guidelines are meant to be an addition to the general best practices for developing and applying AI, including the use of state-of-the-art IT security measures that are crucial for the safe operation of AI systems. Effective IT security measures help to protect against some of the threats considered here and thus are assumed to be in place at all times. Although not explicitly discussed here, a trustworthy AI system is expected to follow the best practices in fairness, accountability and transparency, among others, and fulfill regulatory requirements.
The chapter is structured as follows: In Section 1.1, we provide basic knowledge about AI systems and their security-relevant properties. This includes threat models that are aligned with the life cycle of a system and approaches to quantify the robustness and resilience of NNs. Section 1.2 lays out the structure of the best practices and explains how to use the introduced guidelines. Further, the best practices and respective limitations are discussed in detail, followed by com-

mon limitations for all defenses. Finally, in Section 1.3, we take a holistic view on the individual best practices and outline possibilities and challenges to combine the recommended methods.

1.1 Preliminaries

In Section 1.1.1, we describe the most important aspects defining DL-based AI systems. We divide these aspects into domain-determining as well as security-relevant ones. While the former determine the NN's architecture, the latter need to be considered during the definition of the individual threat model related to the system. Threat models describe the capabilities of potential adversaries and therefore determine the resulting attack vectors. In Section 1.1.2, we give a general introduction to threat models and describe the attacker goals. We outline which individual attack types can allow adversaries to reach these goals. Additionally, we show at which step of the AI system life cycle the discussed attacks are applicable. This gives the reader a concise overview of relevant threats and therefore recommended best practices depending on the current life cycle stage. Finally, in Section 1.1.3, we discuss how the resilience of NNs against the introduced attacks can be measured.

1.1.1 Overview of AI Systems

Different aspects arise when deploying AI-based systems, influencing each step of the life cycle. In this section, we describe the most prominent of them and distinguish between two categories. The first category summarizes general domain-determining aspects with minimal influence on the security of the entire system. Complementary, the second category includes features of AI systems as well as the environmental aspects, which are highly security relevant. This builds the basis for our best practice guideline useful during the design and deployment of AI systems. Generally, we distinguish between the AI system and its inner "intelligent" ML core, i.e., a NN in the scope of this document. The overall AI system is deployed to its final use case and may feature multiple components next to the NN, e.g., sensors or actors. This chapter discusses general security aspects of the entire AI system, whereas our best practices in Section 1.2 specifically address defenses against attacks on the NN core of the system. When combined, both perspectives provide a holistic view on the security of NN-based AI systems.

Domain-determining Aspects

The type of the underlying AI is largely determined by a chain of design choices. First and foremost, the deployment area determines the input and output of the overall AI system. Based on this embedding, the data types, which the AI must process, differ. Then, the AI practitioner selects the suitable architecture for her DL algorithm. All these aspects influence the general domain of the underlying AI, not directly its security aspects. In other words, the following choices determine which ML tools can be used and therefore only indirectly influence the overall level of security.

Deployment Area As initial step in the design process of AI systems, the deployment area will be defined. Among others this consideration includes the desired functionality, i.e., how the input data is processed, the system components, i.e., the building blocks of the overall AI system, the anticipated location, i.e., where the final product is located. Each design decision influences which ML algorithms may act as foundation and how much in-house development will be necessary. Once the deployment area is determined, AI practitioners will consider the involved data types and the general architectural properties of their system.

Data Type The general nature of the input influences design choices in the downstream AI algorithm. For images, for example, the semantic information between the input pixels is important. In practice, the data may contain a time dependency, e.g., sensor readings or videos. For such sequential data, each sample contains information about its predecessor and successor. Each data type mandates an individual DL architecture suitable for the task.

Architectural Properties Combining the knowledge of the available type of data as well as the information on the underlying use case determine the design of the AI component and its integration into the system. Here, we distinguish between the basic concept as well as the finally applied architecture. Throughout this project, we focus on DL-based AI systems. Note that classical ML models as well as statistical analyses of the data still play a vital role in real-world applications. Yet the high complexity of DL-based systems leads to a set of unmitigated vulnerabilities. Neural networks can consist of different building blocks and architectures. For instance, Convolutional Neural Networks (CNNs) are typically used in image-processing tasks. On the other hand, Recurrent Neural Networks (RNNs), e.g., based on Long Short-Term Memory (LSTM), achieve remarkable results when processing sequential data.

Security-relevant Aspects

In this section, we summarize security-relevant aspects of AI systems. Unlike the domain-determining aspects above, these points may directly facilitate attacks on the AI system. AI practitioners should assess the potential risks and possible attacks occurring during the development and deployment. Therefore, we elaborate on the impact on the security of the overall AI system and give specific examples for each feature. In Table 1.1, we summarize the examples and divide them into instances of low, medium, and high impact on the risk the system is exposed to.

Deployment Environment The physical deployment environment influences which parts of the AI system can be accessed by whom. Whereas the input to AI systems in a lab environment has well-defined boundaries, an AI deployed to a website will pose a higher risk. Also the embedding of the hardware influences the AI system's security: an attacker with access to the sensors or even the operating system can easily influence the AI's decision. Thus, we suggest thinking about possible attack vectors based on the deployment environment. Important follow-up questions arise on the access rights and input origins as discussed in the following.

Risk	**Low**	**Medium**	**High**
Deployment Environment	Known lab environment	Confined open-world deployment	Publicly available online service
Training Data Origins	Well established data sets	Self-created data sets	Data sets of unknown origin
Inference Data Origins	Local, e.g., a fixed data set	Air gap, e.g., a camera system	External queries, e.g., the internet
Degree of Autonomy	Recommendation to human expert	Decision observed by human expert	Autonomous decision by the AI
Output Type	System decision	Discretized model output	Full model output
User Access Origins	Local admins	All internal users	Global access
User Access	Single query	Limited number of queries	Unlimited queries
Model Origins	Locally trained model with custom architecture	Locally retrained public architecture	Publicly available model
Learning Strategy	Single training	Continuous and regular updates	Continuous and spontaneous updates

Table 1.1: Examples for the risk impact of the security-relevant aspects of AI-based systems and environmental settings. We distinguish between the risk classes of low, medium, and high impact. There are also possible links between different risks, which will give different results.

Training Data Origins Independent of the type of data, the origin of the samples has an impact on the potential risks of the overall AI system. This concerns the training phase and ultimately affects the inference. Two aspects need to be emphasized: insufficient and spoofed training data. First, the quality of the data influences the performance, generalization, and ultimately the robustness of the models. If the training data set is not representative, there may be significant performance drops during inference. These blind spots in the training data can lead to low reliability as well. Second, attackers may introduce intentionally altered samples or backdoors. Ultimately, such attacks could lead to low real-world performance or vulnerabilities, which are exploited during inference. Specialized use cases, e.g., federated learning, might be subject to data poisoning from multiple data origins.

Inference Data Origins During inference, the AI system processes yet unseen data. Depending on the origin of these inputs, serious security threats might arise. If the origin and integrity of the input data cannot be verified, attackers may mount attacks influencing the output of the AI. For a full security assessment, the properties of the sources need to be defined. Such sources may range from direct human input, pictures taken by cameras, or other sensors providing measurements of the physical world.

Degree of Autonomy The AI's outputs may further influence other system parts. Misclassifications of the AI may thus have a severe impact on the overall functionality. These errors may arise due to a weak AI model or are actively provoked by an attacker. Especially in AI systems that interact with their environment, e.g., in autonomous or medical systems, it is vital to increase the robustness against attacks. Furthermore, the AI system's output may be used to deduce insights about the NN itself. We thus recommend to carefully assess how the output can be observed and how it is further processed. Systems that directly act on the AI's output may pose a high risk when being attacked.

Output Type Some attacks become easier the more knowledge about the output is available. Especially information extraction attacks profit from a detailed view on the AI's output. AI systems may return the entire output probability distribution, e.g., object detectors indicating the likelihood that some objects are within the processed input image. In contrast, some systems may only show the implication of the AI's decision, e.g., an autonomous vehicle following a specific path. Less information about the AI's output complicate attacks, but it may still be possible to estimate the underlying decision process well enough to mount successful attacks.

User Access After the training and deployment of the AI system, evasion and extraction attacks pose the most relevant threats. The majority of attacks require multiple iterations in which the model is being queried. Hence, the access of the user and the allowed number of queries fundamentally influence the choice of attack methods. Note, even for systems with highly restricted user queries, one step evasion attacks (e.g., FGSM) or black-box attacks still pose a potential risk. Furthermore, we point to general IT security practices, e.g., limiting the access right (also physically) to the AI system.

Model Origins The origins of the model influence the risks and potential vulnerabilities exploitable by attackers. Here, we distinguish between in-house trained models, pretrained ones from, e.g., online sources, or externally trained ones, e.g., by a service provider. When using pretrained or externally trained models, the following risk factors arise: First, the model itself can be poisoned, allowing the attackers to potentially trigger backdoors. Similarly, the performance of the model might have been intentionally influenced by attackers, diminishing the applicability of the model in the desired use case. Secondly, the architecture and weights of the models might also be directly available to the attackers. This eases the process of generating adversarial examples and mounting the final attack, as well as producing shadow models for different kinds of privacy attacks. Note, the aforementioned factors need to be considered when in-house retraining the model, i.e., during transfer learning. For the other case, when the model is locally trained without preset weights, we again need to distinguish between two cases. In the first case, the model is built using a custom architecture, while in the second scenario a commonly used architecture is used. Following the latter, transfer attacks can be performed with a higher chance of producing adversarial examples fooling the maintainer's model. It was shown that surrogate models with a high architectural level of similarity to the originally attacked model result in higher success rates during transfer attacks [441, 418].

Learning Strategy Based on the learning strategy, the AI system may heavily depend on external resources, which impacts the attack surface. Often, existing models are retrained and used as basis for the derived AI system, called transfer learning. Potential vulnerabilities of the base model may be still existing in the new one. Environmental changes as well as task adaptations may lead to the necessity of updating the deployed model, often called continuous learning. We argue that any update to the model running in the field should be seen as the deployment of a new model. Retraining the model, updating the embedding, or reformulating the task may lead to new vulnerabilities and potential threat vectors. Therefore, the schedule in which the model is updated, regularly or irregularly, should be precisely monitored. Similar risks apply when using training samples from multiple sources, e.g., in federated learning. When not all data sources can be trusted, it may pose a serious threat to the integrity of the overall AI model. We recommend to adapt the defense method to the learning strategy of the underlying AI system.

1.1.2 Defining a Threat Model

The threat model describes the attacker's goals, knowledge and capabilities. Before applying any defense strategy, it is mandatory to define which threats the AI system is exposed to. Only then the AI engineer can choose from the wide range of methods and finally check if the applied defenses indeed have a positive impact on the resilience against attacks. Although inherent to the respective use case, there are three major components a threat model should address [56]:

1. Attacker Goals: What outcome does the adversary intend by mounting the attack?
   The goals of the attacker determine the set of relevant attack types, which we will discuss in Section 1.1.2.

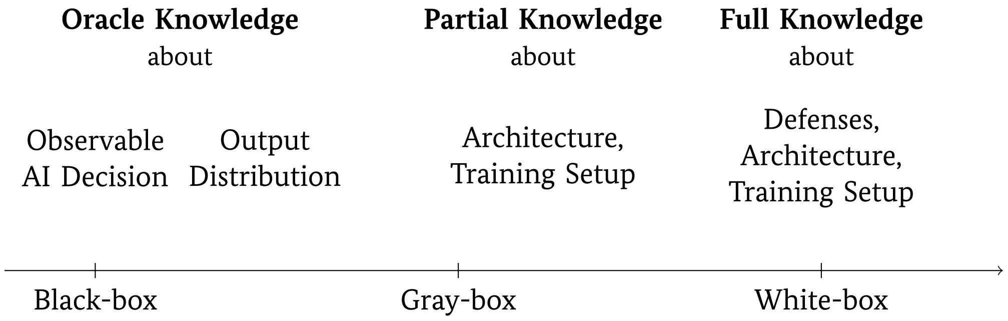


Figure 1.1: Levels of attacker knowledge, ranging from black-box to white-box access.

2. Attacker Capability: What parts of the AI system can the adversary change to reach the attack goals?
   In Section 1.1.1, we discuss examples of exploitable properties. Complex systems may contain multiple interrelated parts, which must be considered separately.

3. Attacker Knowledge: How much knowledge does the attacker have about the AI system? Even with zero knowledge in the beginning an attacker will often acquire some system details over time. We thus recommend evaluating both a white-box, i.e., absolute knowledge, and a black-box, i.e., no prior knowledge, scenario.

The aforementioned attacker knowledge can roughly be categorized into black-box, gray-box or white-box scenarios. We visualize the three levels of increasing knowledge about the AI system in Figure 1.1. In black-box settings, attackers solely have access to the outputs of the observed models. This may range from the final decision to the output distributions useful for further calculations. If only partial information about the architecture or the training setup, potentially including training data information, is known, we usually speak about gray-box access. White-box access refers to the attacker's full knowledge of the model architecture and training setup, as well as to also knowing the used defense mechanisms. Here it should be noted that throughout the development of evasion attack algorithms the notion of "white-box" changed. Initially full access would be assumed when the attacker knows everything about the target model. With introduction of attacks that adapt to the used defenses, knowing the defense became a necessary part of the "white-box' setup'.

Attack Types

In our work, we discuss the impact of active attackers on AI systems. We loosely grouped the attacks based on their occurrence in an AI system's life cycle, see Figure 1.2. In Section 2.1.1, we give a holistic overview about attack categories considered in research.

Evasion Attacks Successfully attacks cause a wrong prediction, i.e., a misclassification of the AI system. We distinguish between targeted attacks, i.e., the AI should predict a specific class, and untargeted attacks, i.e., the AI should predict any other class than the original one. Evasion attacks are applied during inference, i.e., on a trained DL model.

Poisoning Attacks Successful attacks diminish the performance of the AI system. The poisoning samples are injected during training of the AI and cause severe performance degradation. Poisoning attacks may influence the performance of single or multiple classes.

Backdoor Attacks Successful attacks insert weaknesses in the AI system during training, which can then be triggered by the attacker during inference. A specific output is forced when the trigger is part of the input.

Model Extraction Attacks Successful attacks allow the attacker to gain knowledge of the used model. Information on the intellectual property possibly extracted by the attacker ranges from the applied architecture of the model to even the weights set during training.

Data Extraction Attacks Successful attacks allow the attacker to extract information on the used training data. In membership inference attacks, the adversary determines if specific data samples were part of the training process. More powerful attacks lead to the capability of extracting complete training samples from the applied model.
In Table 1.2, we link our introduction on security-relevant aspects from Section 1.1.1 and the attacker goals. With this table, we give the reader an overview of the specific attack surfaces based on the introduced aspects.

Points of Attacks in the Life Cycle of AI Systems

Throughout the development of AI systems, there are multiple points prone to attacks. Identifying potential threats at each step is vital for a comprehensive threat model, and in the end for applying the appropriate defenses. We give an overview about the life cycle of an AI system in Figure 1.2. At each step, we show the possible attacks, which we introduced in the previous section. Generally, we distinguish between training and inference. We advise AI practitioners to carefully assemble a threat model for each relevant step in the life cycle of their AI system.
While training, the AI adapts to the given training data and objective. Data poisoning is the main security threat at this stage. By altering the training data, the attacker can shift the AI's functionality at her will. Implications may be a degradation of performance or general malfunctions of the system. DL algorithms are especially data-demanding, i.e., benefit from large data sets for training, which often require external resources. This, however, may allow attackers to insert security backdoors, which later can be triggered during inference.
After training, the model is deployed to inference operation. Here, it serves its main task, being applied to real-time external inputs. Main security threats are evasion, model inversion, and extraction attacks. Spoofed inputs, e.g., adversarial attacks, may shift the AI's decision at the attacker's will. Also, inserted backdoors might be triggered by data points presented at inference.

	Evasion	Poisoning	Extraction
Publicly Available AI Service	●	◑	●
Unknown Training Data Origins	◑	●	◑
Unknown Inference Data Origins	●	◑	●
Autonomous Decision Done by AI	●	◑	●
Full Model Output Observable	●	◑	●
Unrestricted Model Access	●	●	●
Publicly Available Model	●	●	●
Continuous Learning	◑	●	◑

Table 1.2: Impact of exemplary high-risk security-relevant aspects on the success of attacks. Here “Poisoning” unites both poisoning and backdoor attacks, and “Extraction” model and data extraction attacks. Some aspects have a direct impact on the attack success (●), some may indirectly influence it under certain conditions (◑).

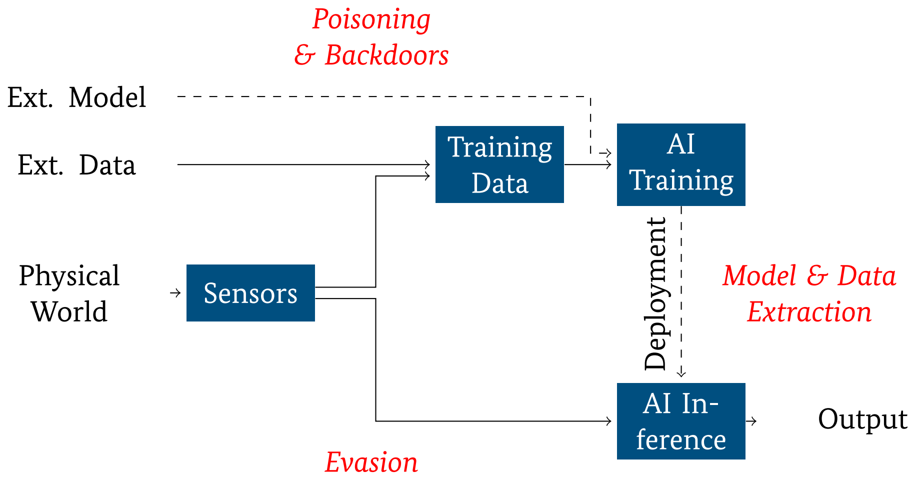


Figure 1.2: High-level scheme of the life cycle of an AI system and the attacks influencing the respective stage.

Moreover, while observing the AI's output on a given input, an attacker may be able to see if certain data points belong to the training data, or even extract the underlying model.

1.1.3 Quantifying the Robustness and Resilience of Neural Networks

To measure the effectiveness of defenses, appropriate metrics are required. These quantify the robustness increase and allow the AI practitioner to evaluate and adapt the applied countermeasures. We present metrics relevant for evasion, poisoning, and information extraction attacks. As the term robustness is mainly associated with and used for adversarial attacks (i.e., adversarial robustness) in literature, we speak about resilience in case we want to emphasize that we are addressing privacy, poisoning and backdoor attacks on models. Overall, the terms can be used interchangeably when the attack type is clear from the context.

Adversarial Robustness

Adversarial robustness measures the resilience of an NN against adversarial attacks. Suitable defenses increase the adversarial robustness. As discussed, there is no absolute remedy against adversarial attacks – however, the attacker's effort should increase with each countermeasure applied. Robustness metrics quantify this security enhancement. In the following, we introduce the standard empirical approach to allow an assessment of this security enhancement. Here, the AI's response on attacks crafted by state-of-the-art attack methods is recorded. For this purpose, we distinguish between bounded (e.g., FGSM [152], PGD [276]) and unbounded attacks (e.g., DF [297], C&W [62]) and introduce the means to measure the robustness for both sets of methods. The metrics are based on a threat model where the attacker changes the entire input to shift the AI's output at her will. For other scenarios, e.g., patch-based attacks, similar metrics can be derived reflecting the respective attacker goals. Generally, the metrics assume a single input, which is used as attack, thus reflecting the local robustness view. Further tests may summarize the robustness across specific classes or entire sets of data, i.e., the global view. We recommend calculating the mean robustness metrics for multiple inputs to allow a more reliable estimation of the models' robustness.

For Bounded Attacks Bounded attacks are limited by their fixed perturbation budget ϵ, i.e., the maximum distortion added to the entire input measured in some l_p-norm. During the attack, the algorithm optimizes the position of the distortion to cause misclassification. For bounded attacks, we thus measure the adversarial robustness from two angles: First, using the minimal number of iteration steps $\hat{i}$ until the attack is successful: $\hat{i} = \min_i \delta_i : f(x + \delta_i) \neq f(x), \|\delta_i\|_p \leq \epsilon$. The more robust the NN, the more attack steps are needed by an attacker to mount a successful attack. Secondly, using the attack success rate for multiple inputs and a fixed number of attack iterations. The attack success rate is calculated via the fraction of successfully fooling adversarial examples among all perturbed samples.

For Unbounded Attacks Unbounded attacks are not limited in their perturbation budget (thus they will always be successful), but minimize the distortion during the attack. Thus, we use the minimal distortion $\hat{\epsilon}$ as robustness metric: $\hat{\epsilon} = \lim_{i \to \infty} \|\delta_i\|$ because $\|\delta_{i+1}\| \leq \|\delta_i\|$, i.e., the induced change is minimized for each iteration of the attack. In practice, we observe $\hat{\epsilon}$, when δ_i does not change for multiple epochs, or again for a fixed number of attack iterations. The more robust the NN, the higher $\hat{\epsilon}$ will be, i.e., a higher level perturbation is needed for a successful attack.

A Word on Distance Metrics

Distance metrics quantify the difference between two data points. In adversarial ML, we usually use an l_p-norm between the original input x and its adversarially perturbed version $x + \delta$. In other words, we quantify how much the input was changed by mounting the attack. Formally, we define the l_p-norm as: $\|x\|_p = (\sum |x_i|^p)^{1/p}$. Although regularly used in research, there are certain drawbacks of the l_p-norm, which should be known when evaluating the robustness of NNs.

In adversarial ML, a suitable metric reflects the perceptibility of attacks. Intuitively, the attacker's goal is to mount a successful attack while hiding the intentions, i.e., to minimize the applied adversarial perturbation. The l_p-norm measures the pixel-wise distance in a p-dimensional space. As the error is aggregated among all dimensions, a few severely altered pixels may result in a small l_p-norm when other pixels are not altered. However, this behavior does not well align with the human visual perception, which focuses on semantic relations. A human observer may find the adversarial perturbation of an l_p-norm more obvious than semantic changes, e.g., a transformed version of the original image. Recent research has transferred adversarial attacks to other approaches, e.g., based on Wasserstein distances [473, 475] or small deformations [9]. Instead of applying pixel-wise additive perturbations, these attacks transform the existing pixels.

As result, the semantic meaning between the pixels is preserved, making these changes harder to detect for human observers.
As we will discuss in the limitations, see Section 1.2.2, the used distance metrics have a severe impact on the success of defenses: when defending against attacks of a certain metric, the AI system may still be vulnerable against other attacks. We thus recommend to carefully assess potential risks and embed the findings in the threat model.

Resilience Against Poisoning, Backdoor and Privacy Attacks

For all the attacks it is common to measure the success rate, meaning how many of the manipulated inputs were misclassified. In the case when a model is not protected, this directly characterizes its resilience – otherwise it can reflect the effectiveness of an applied defense.

For Poisoning and Backdoor Attacks Pang et al. [323] list a number of so-called defense-utility metrics which can be considered for evaluating the effectiveness of defenses. For poisoning attacks with the goal of availability reduction we can consider the amount of poisoned data that has to be injected in the training data set for achieving a desired drop in performance. On the other hand, backdoor attacks are characterized by the amount of presented triggers that led to the intentional wrong output. In case of the trigger optimization approaches, the amount of iterations needed to get desirable result is a viable measure as well.

For Privacy Attacks Important characteristics of privacy attacks are accuracy of the extracted parameters/data points, e.g., correlation between targeted labels and those extracted by the attack. For membership inference attacks one can measure the success rate of attacks with different level of attacker's knowledge [305].

1.2 Best Practice Guidelines

We start our discussion of best practices by providing general recommendations on assessing and identifying potential threats. AI systems are vulnerable at multiple points in their life cycle, each requiring carefully chosen countermeasures. Our guidelines embed the theoretical aspects into the workflow of AI engineers. Before implementing a specific defense, we recommend thinking about the embedding of the AI. As initial source of information, we refer to Section 1.1.1. The following points help developers to mitigate possible security risks and increase the resilience of their AI system.

1. Consider Standard IT Security Best Practices:
   The implementation of the used data pipeline, model embedding, and further required components should be done with respect to common IT security related best practices.

2. Identify Relevant Stage in the Life Cycle:
   Depending on the currently relevant stage in the life cycle, different threat models and

therefore best practices apply. After completing a life cycle stage, a re-evaluation of the security and consideration of the best practices should take place.

(a) Analyze Risk of the AI system:
Assemble an overview about potential risks and their severity on the overall AI system. This includes an assessment of attack points and attack probabilities among others.

(b) Define Concrete Threat Models:
Summarize the identified risks along with the attacker goals in threat models. Consider at least a white-box, i.e., an omniscient attacker, and a black-box, i.e., a zero-knowledge attacker, to assess upper and lower bounds on the resilience.

(c) Consider Relevant Parts of Best Practices:
Depending on the life cycle stage and the definition of the relevant threat models, different best practices apply. Carefully choose the relevant defenses discussed in the best practice guide.

(d) Test Robustness Increase by Applied Defenses: Check if the applied defenses increased the AI's resilience under each given threat model. Iteratively enhance the defense strategy until the required level of resilience is reached. It is always desirable to use more than one measurement of resilience, since they are usually complementary and each one alone cannot guarantee perfect correlation with the degree of protection.

3. Continuous Reassessment:
With each step in the life cycle, changes in the architecture, or model retraining, reconsider the points above.

1.2.1 Certifying Robustness

Robustness certification plays an important role when it comes to safety and security of NNs. As mentioned before, the robustness of a NN describes its ability to tolerate input perturbations of a specified range (bounded attacks, see Section 1.1.3), so that the original model prediction for the unperturbed input is not changed. Robustness certification as well as robustness verification, both stand for the procedure of evaluating a model's resilience against manipulated inputs. In scientific literature, there is no distinct differentiation between the two terms. In this document robustness certification refers to algorithms that provide robustness bounds by approximation or metrics for estimation of a model's robustness, while verification means the formal verification of a model, i.e., the detection of precise and definite bounds.
Regarding the AI life cycle, robustness certification is beneficial during the model development process. In this manner, a vulnerability against adversarial attacks can be identified early on, so that adjustments, i.e., application of defense mechanisms as proposed in Section 1.2.2, can be made. After development, a certification can provide a demonstrable guarantee for a minimal robustness of a NN. The proven robustness level might be utilized in security audits, e.g., within the scope of standardization. Re-training or continuous training during deployment present a specific challenge, so that repetitive or continuous certification might be necessary.

There are mainly two ways of manipulating a ML model's prediction: Evasion attacks on the one side and data poisoning / backdoor attacks on the other. The majority of robustness estimation approaches deal with robustness against evasion attacks. The estimation of resilience against data poisoning and integrated backdoors is more complex due to the manipulation of the training process. Only few publications [466, 235, 364] exist that address the topic and further research is needed. The main approach should be an avoidance of data poisoning in the first place paired up with detection methods for training data and model as proposed in Section 1.2.3. In this document, the focus lies on certification and verification of robustness against evasion attacks or adversarial robustness.
There are several approaches for robustness estimation ranging from complete verification, overapproximation, to partial proof of robustness. In Figure 1.3, some representative approaches, which influenced research, are presented for the specific categories, that were chosen for this best practice approach. A comprehensive overview of all categories for certification and verification algorithms is given in Section 2.1.2. Section 2.2.2 gives summaries of the related literature.

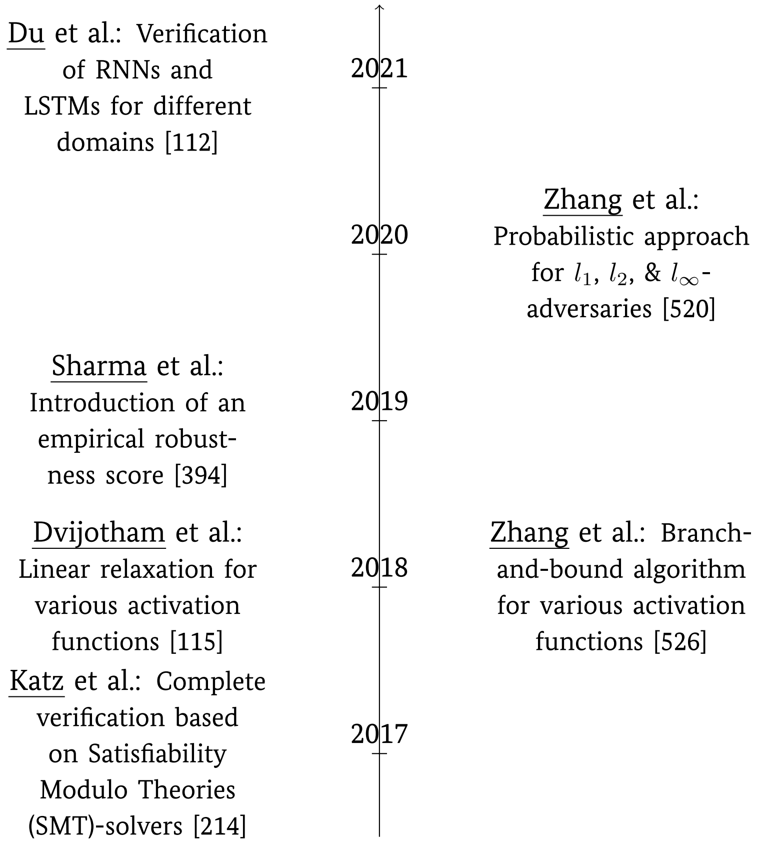

Du et al.: Verification of RNNs and LSTMs for different domains [112]
2021
Zhang et al.: Probabilistic approach for l_1, l_2, & l_∞-adversaries [520]
2020
Sharma et al.: Introduction of an empirical robustness score [394]
2019
Dvijotham et al.: Linear relaxation for various activation functions [115]
Zhang et al.: Branch-and-bound algorithm for various activation functions [526]
2018
Katz et al.: Complete verification based on Satisfiability Modulo Theories (SMT)-solvers [214]
2017

Figure 1.3: Milestones in adversarial robustness certification literature over the past years

In the following, a best practice for robustness certification and verification is proposed. Advise is given on how to choose a suitable technique, while taking into consideration the given model,

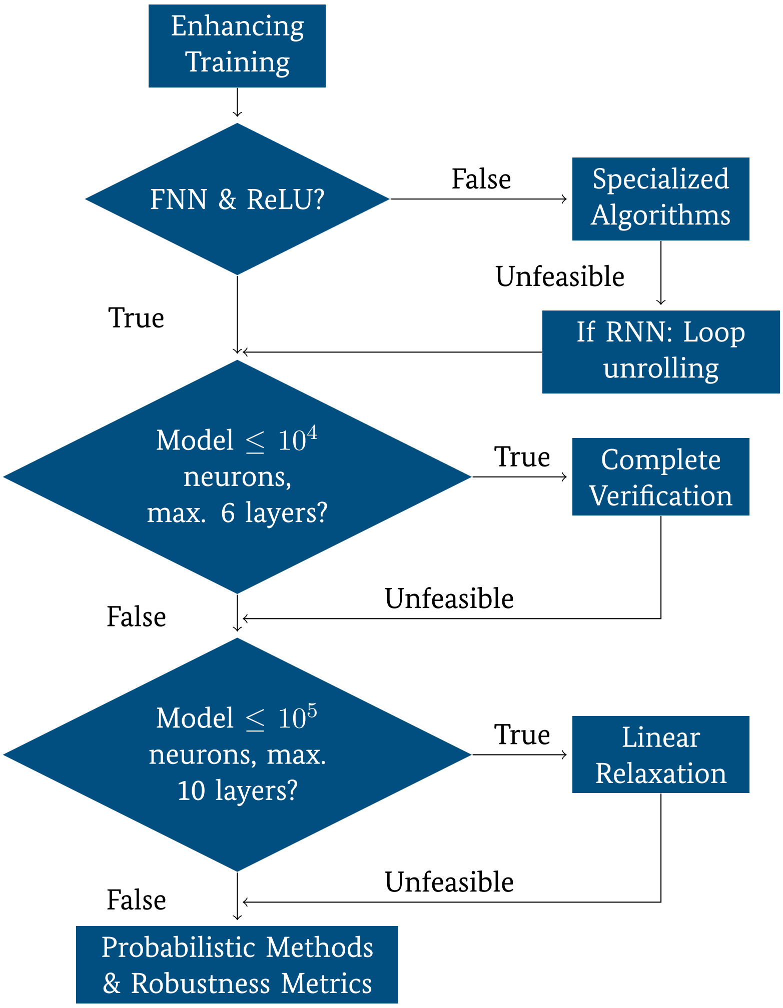


Figure 1.4: Best practice for certification

data sets, and capabilities of an attacker. Figure 1.4 provides a schematic illustration and combines the introduced aspects to form a procedure. Additionally, the limitations of the specific methods are highlighted. This best practice approach should be seen as current state-of-the-art which can change with advancing (especially regarding model sizes) and new verification and certification techniques.

Enhancing Training A specially customized training procedure can allow or improve robustness measuring attempts regarding scalability and tightness of bounds [524, 493, 397]. For example, it enables Branch-and-Bound (BaB) algorithms, a complete verification technique, to verify models with up to 10^5 neurons, e.g., ResNet [164], within several minutes. Without adjusted training, only models with up to 10^4 neurons would be verifiable. A suitable training procedure has to be selected according to the chosen certification technique.

For probabilistic approaches and robustness metrics, a special training procedure is not effective since these methods do not evaluate the inner NN model.

Specialized Algorithms The model architecture is of significant importance when choosing an algorithm for robustness certification. The majority of techniques presented in the following is build for Feedforward Neural Networks (FNNs) using Rectifier Linear Unit (ReLU) activation. For CNNs, while also feedforward and therefore principally verifiable by standard certification algorithms, it can be computational beneficial to use specifically customized approaches [141, 39].

Due to its recursive connections, it is more difficult to verify RNNs compared to basic FNNs. By unrolling, the recursive loops can be dissolved and the RNN is converted into a feedforward structure without losing any functionality. But unrolling comes with an increase of complexity. The newly created FNN will be much deeper, which in the following, impedes the application of a certification or verification technique.

Besides general structure, also the used activation functions play an important role. As stated, most certification approaches rely on partially linearity of ReLUs. For other activation function, specialized methods, e.g., DeepPoly [405], CRC [407], and the framework proposed by Dvijotham et al. [115], or methods independent of model architecture, i.e., probabilistic methods, have to be considered.

Complete Verification To gain a robustness estimation as accurate as possible, a method from the class of complete verification algorithms is the preferable way. A complete verification of a NN can provide a guarantee, that crafting adversarial examples for an input is not feasible within a specific perturbation space. This leads to exact bounds for the robustness of a model.

Principally, there are two techniques for complete verification. The first one is based on solvers for mathematical problems, e.g., Satisfiability Modulo Theories (SMT) and Mixed-Integer Linear Programming (MILP). Most NNs are based on affine transformations and ReLU activation functions, which can be expressed by a set of linear inequalities. Logical solvers can decide on their satisfiability for a given input and thus, give an assertion if the NN is robust.

The second technique for complete verification are Branch-and-Bound algorithms. They are again based on the piecewise-linear property of NNs using ReLU activation. These types of algorithms traverse the model and alternately apply a branching and a bounding step. The bounding is an incomplete verification leading to a lower and an upper bound for the deviation between original prediction and a prediction coming from a manipulated input. If the lower bound is high enough, i.e., the difference between original and manipulated prediction is small, the model is verified. On the other hand, if the upper bound is not high enough, so there is a high disjuncture between original and manipulated prediction, the model is not verified. If neither is the case, then the branching step comes into play. The model's neurons are iterated over, and the single neuron respectively its ReLU activation function is split into two linear constraints. Then, again bounding is executed on both branches. So the branching and bounding steps are applied recursively until every branch either lead to a verification decision or was transformed completely into linear constraints. The latter can then be solved by linear programming.

In general, BaB algorithms are more scalable than solver-based techniques, but feasibility of complete verification is depending on the size of the model. For further information, see the limitations below.

Linear Relaxation Linear relaxation is part of certification algorithms and allows robustness estimation for larger and more complex NNs compared to complete verification. Once again, the piecewise-linear property of ReLU NNs is exploited. As stated before, ReLUs can be expressed by linear equations and constraints. The linear constraints can be approximated in polytopes. With the help of these, the overall NN's output space is described which in turn is useful for robustness estimation.
Though linear relaxation offers enhanced certification capabilities, it is still constraint when facing a certain model size. Again, for more information, see the limitations below.

Probabilistic Methods The principle of probabilistic methods is the extension of the model to be verified by adding random noise to its input. This process is called "Randomized Smoothing". The smoothed model is a transformation of the original classifier's decisions with the highest probability when confronted with the noisy input. It can be verified by estimating the probability that its decision is the same as the original model's for the non-noisy input. The certification holds for the smoothed model version. The used noise distribution has perceptible influence on model performance and is responsible for the tightness of robustness bounds. This category of methods is independent of model architecture, activation function and model size, but provides only probabilistic certification results valid for the smoothed model.

Empirical Robustness Metrics Empirical Robustness metrics do not grant guaranteed robustness bounds, but will provide an indication of the robustness of the model. As mentioned in Section 1.1.3, input perturbations are generated by several state-of-the-art attack methods. Especially, testing of input corner cases should take place. The model performance is then evaluated on the crafted inputs to obtain an estimation of the model's robustness against them. Berghoff et al. [31], Hartl et al. [159] and Sharma et al. with CERTIFAI [394] present approaches for the generation of such empirical robustness scores.
Just like probabilistic methods, the application of robustness metrics is independent of model architecture and size, since the technique solely operates on inference.
This category of certification merely covers a small portion of the possible input and perturbation space and therefore provides only an incomplete verification of a model's robustness.

Limitations

Model Architecture The algorithms implementing complete verification, as well as linear relaxation are utilizing the piecewise-linear property of FNNs and ReLUs, and are therefore mostly limited to this kind of architecture. Specialized methods provide verification or certification for FNNs with arbitrary activation functions or even other architecture types, e.g., RNNs, CNNs, but are still limited regarding scalability.

Scalability & Tightness of Bounds The problem finding a complete verification is NP-complete [214] and only small and simple NNs, can be verified by this category of methods. It is possible to verify models with up to 6 layers and $\leq 10^4$ neurons. Linear relaxation allows certification of models with $\leq 10^5$ neurons and up to 10 layers. It is not reliably feasible to verify more complex models, due to the accumulating imprecision of the overapproximation.
For BaB algorithms and for those based on linear relaxation, there is a trade-off between scalability and performance, i.e., tightness of bounds and run-time. The more complex a NN is and the more neurons and layers it has, the worse the ability to find accurate robustness bounds. On the other hand, tighter bounds can be obtained with increasing run-time of the certification, whereby this relation is likely non-linear. When dealing with models of a certain size, sometimes only loose bounds can be verified or certification may even be computationally unfeasible. Therefore, it is important to identify a suitable goal for robustness bounds, e.g., prescribed by standards, verify if the goal is met, and if necessary try different certification methods. The same applies for probabilistic approaches. These are independent of model size, but sufficient similarity of the smoothed and the original model should be ensured.
Robustness metrics do not provide fully robustness bounds, but only a partly verification of the examined input space covered by the generated adversarial examples.

Attacker Capabilities Most certification and verification approaches only provide robustness predicates for a defined set of capabilities of an adversary. Computed robustness bounds are always related to and only valid for the assessed input perturbations. As stated in Section 1.1.3, the distortion of an input can be measured by the l_p-norm. In the following, the characteristics of the introduced methods regarding the capabilities of an attacker are presented.
All techniques related to complete verification can efficiently verify NNs for l_∞-adversaries. Additionally, BaB algorithms cover l_2, and SMT-solvers l_2 and l_1-adversaries. Linear relaxation methods also often provide certification for l_1, l_2 and l_∞-adversaries, whereby uncertain tightness of bounds for l_1 and l_2 has to be taken into account. Probabilistic approaches cover l_1 and l_2 efficiently, for l_∞ only non-trivial bounds can be verified. Robustness metrics provide certification regarding the complexity of the adversarial attacks used to generate the utilized robustness scores.

1.2.2 Defending against Evasion Attacks

During deployment, evasion attacks shift the output of unprotected AI models at the attacker's will. Attackers design specifically crafted inputs, which are visually close to a benign sample, yet cause misclassification. The discrepancy in perception between human and AI-based observers make evasion attacks a severe security threat in real-world scenarios. Especially in security-relevant applications as autonomous driving or the medical sector, a misclassification may have serious implications. AI practitioners are advised to carefully adapt the countermeasures discussed in the following. For a detailed overview of recently published methods, we refer to Section 2.1.1 and to Section 2.2.3 for a categorization of the topic.

Output Bounds by a Certification or Verification Method Certification methods calculate certain guarantees on the output distribution given the current input. A suitable method thus reveals possible attack vectors – or, in case only the target class is reachable within certain constraints, certifies that the AI model is indeed robust against evasion attacks. Repeated for likely inputs, certification methods give the most thorough overview about the AI's robustness. However, as of now certification methods have high computational costs and are not applicable to all NN architectures. Except for high-risk applications, AI practitioners will defer to other defense method – future research may allow the application of certification methods in a wider context.

Adversarial Retraining Adversarial retraining in considered to be the most feasible defense against evasion attacks. It is easily applicable during the training of NNs, yet boosts the adversarial robustness significantly. Practitioners are advised to consider adversarial retraining for all AI models exposed to external inputs. During adversarial training, the AI is actively attacked, but the generated adversarial examples are added to the training data. As consequence, the AI learns to classify the adversarially perturbed samples as the original class. It is considered best practice to use PGD [276] or improved versions of FGSM [472] for adversarial retraining. The latter significantly reduces the computational costs involved. Note that attacks optimized for different l_p-norms lead to different attack vectors. Hence, multiple l_p-norms should be incorporated while retraining.

Introduce Randomness during Training The training data gives a partial view on the data distribution of a system. In other words, we can only partially observe all phenomena happening in real-world scenarios. Training data augmentation allows to increase the training data size by certain extents, giving the AI a broader world view. Random transformations within the bounds of natural phenomena may thus be a way to protect the AI against certain trivial attacks, e.g., misclassification due to rotated inputs. Although not secure against white-box attacks, randomness shows promising results against black-box attacks [392]. Due to the low implementation overhead, random transformations should be incorporated in the training process.

Use more Training Data The training data determines the world view of the AI system. A diverse and qualitative training data set allows not only better performance in general, but also more adversarial robustness up to certain extends. Although measurable improvements in adversarial robustness require very large data sets [383], the additional training data does not necessarily need labels [449, 64]. In practice, labels are costly as human observers need to be involved – unlabelled data may, however, be easily obtained. Thus, large qualitative data sets are a requirement for reliable AI systems, but parts of the data may not need labels.

Do not Optimize (Only) for Accuracy There is a natural trade-off between adversarial robustness and the task-specific performance of NNs [445]. High performance on the training data yet low one on the validation split may be a sign of overfitting, which in turn may leave easy attack vectors for evasion attacks. If the AI's decision barriers enclose the known data too closely,

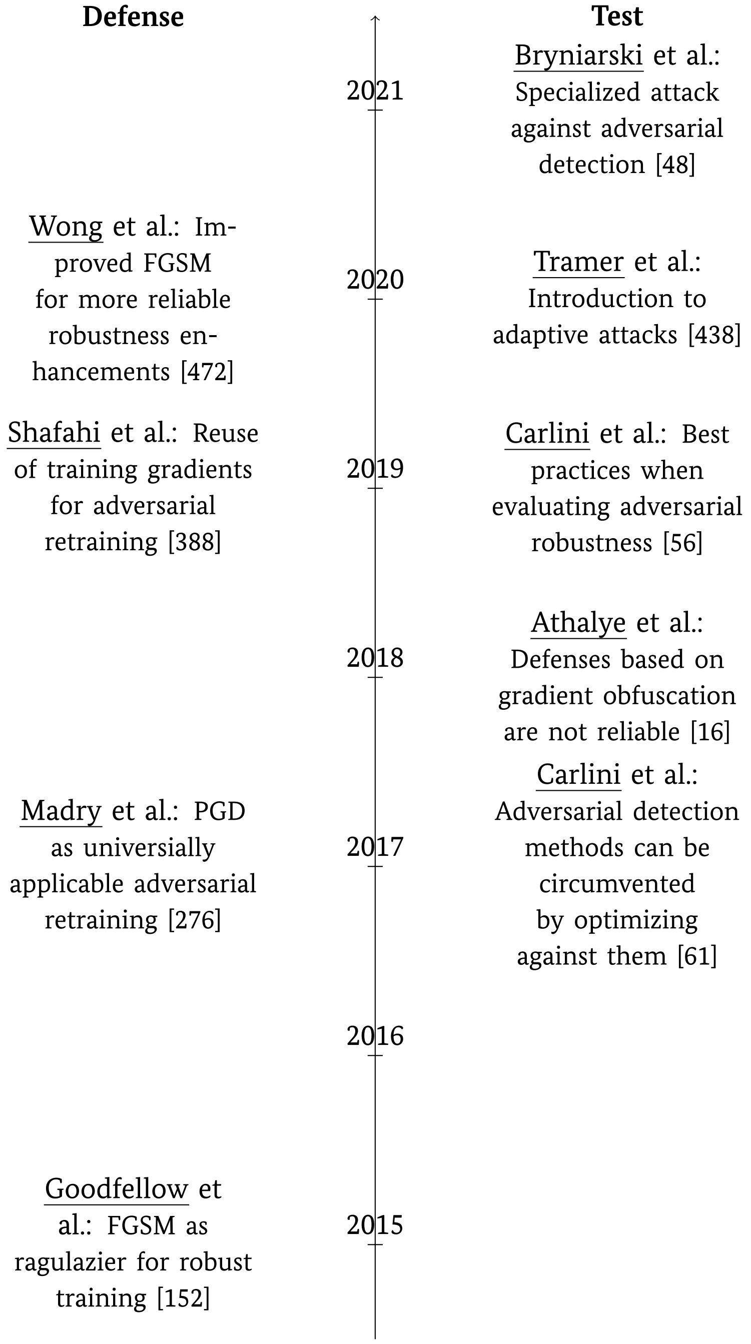

Defense
Test
Bryniarski et al.: Specialized attack against adversarial detection [48]
2021
Wong et al.: Improved FGSM for more reliable robustness enhancements [472]
2020
Tramer et al.: Introduction to adaptive attacks [438]
Shafahi et al.: Reuse of training gradients for adversarial retraining [388]
2019
Carlini et al.: Best practices when evaluating adversarial robustness [56]
Athalye et al.: Defenses based on gradient obfuscation are not reliable [16]
2018
Carlini et al.: Adversarial detection methods can be circumvented by optimizing against them [61]
Madry et al.: PGD as universially applicable adversarial retraining [276]
2017
2016
Goodfellow et al.: FGSM as ragulazier for robust training [152]
2015

Figure 1.5: Milestones in evasion defense and test literature over the past years

it may be easy to sample visually close inputs, which are detected as different class. We recommend AI practitioners to give equal importance to the general accuracy and the adversarial robustness. An AI optimized for only one goal may severely underperform when deployed in real-world settings.

Detect Attacks Whereas most defense methods are applied during training, detection methods may act as a second line of defense during inference. Attack detection methods reveal malicious inputs. AI practitioners may restrict the use of the AI's output whenever the output is considered to be untrustworthy. Although the AI itself is still susceptible to the evasion attack, attack detection methods thus protect the overall system against malfunctions.

Test the Applied Defenses AI practitioners should always assess the effectiveness of the applied defenses. Only when the defenses and their respective hyperparameters are carefully adapted to the use case, the adversarial robustness increases. Thus, we recommend assessing the adversarial robustness regularly using a suitable testing scheme, e.g., applying the robustness metrics introduced in Section 1.1.3 or using testing frameworks like AutoAttack [97]. Defense methods not providing the expected level of robustness should be avoided. Intuitively, methods that were broken by current attacks should not be relied on. Over the past years, research has shown that defenses based on gradient obfuscation [16], an ill-defined threat model [61], or certain auxiliary networks [48] may be easily circumvented. Adversarial ML is a fast-paced field of research, thus AI practitioners are required to regularly search for publications bypassing defense methods.

Limitations

Evasion attacks and the defenses against them are quickly evolving in research. Consistent with the other defense categories, some defenses may be broken by newer attacks – and new, more powerful defenses may arise. In the following, we cover the limitations of the presented best practices and the defenses in general.

Adversarial Retraining needs Careful Parameter Considerations Generally, adversarial retraining depends on the quality of the generated adversarial examples added to the training set. As discussed, the chosen l_p-norm limits the adversarial robustness, such that it is preferable to include multiple norms during retraining. Similar considerations are necessary for the attack budget, the training method and to some extent also the attack method. Although PGD is considered to be a good universal retraining method, it may be advantageous to diversify the attack methods. Based on the chosen parameters, the input sample may be significantly perturbed. In the worst case, it resembles samples of a benign class afterwards, thus reducing the AI's accuracy. Note that this problem may especially arise for data types without strong semantic correlations, e.g., tabular data. As usual for defenses, generally does adversarial retraining not come for free nor does it provide an absolute protection against attacks. Additionally, it may introduce significant resource overheads.

Attacks may Incorporate the Defense Some evasion attacks may integrate the defenses to be successful against them. Even without absolute knowledge which defenses were applied, attackers may increase the robustness of their attacks against common countermeasures. As example, Expectation over Transformation [17] incorporates random input transformations, thus increases the attack success against certain defenses. The same considerations should be taken against attack detection methods. If the method relies on certain input characteristics, an attack method may include them while generating adversarial examples. A successful defense will increase the attack effort an attacker has to apply to mount a successful attack.

Black-box Attacks Will Still Be Possible Even with access limitations on the overall AI system, black-box attacks may still be viable. To some extents, evasion attacks are transferable between models. Black-box attacks introduce a surrogate model, where the attacker has unlimited access on. The higher the similarity of the black-box architecture and the closer the training samples are between the original and the surrogate model, the more likely the attacker can mount a successful attack.

Evasion Attacks Are Versatile Whereas early evasion attacks were only successful when directly applied to the entire input, new methods arose, which required a smaller attack surface. For example, some attacks only need the size of a patch [46] to be successful, or even work across an air gap [78]. It is hard to find a common defense strategy across all types of adversarial attacks. We recommend AI practitioners to clearly outline the expected attack types in a threat model – only then the suitable defense method can be selected.

1.2.3 Defending against Backdoor and Poisoning Attacks

Poisoning and backdoor attacks are usually applied during the AI's training process. These attacks can have two different goals: some attackers aim at reducing overall model performance (the system's "availability") or, more often, they want to provoke malicious model behavior that is unintended by the owner of the model (attacking integrity). Under backdoor attacks, the model is trained in such a way that it strongly reacts to the attacker-chosen trigger. In particular, presenting the trigger to the AI system at inference time activates the backdoor and makes the model behave at the attacker's will. On all other inputs, i.e., benign ones, the model will behave and perform as usual. As there are also triggerless attacks [385], when e.g., clean-label poisoning is performed, the model might also exhibit the manipulated behavior when the receiving target class inputs at inference time.

Backdoors are a realistic threat, especially when relying on third-party data sets or externally trained DL models. This includes, for example 1) pre-trained models in a transfer learning or fine-tuning setting, 2) collaborative learning under malicious local learners, 3) online and continuous learning under variable data sets and 4) outsourcing model training or downloading third-party code. While vanilla data poisoning usually affects only the data collection phase via inserting malicious samples into the training process, overall poisoning-related attacks can affect also other phases of the AI life cycle (Section 1.1.2), e.g., bit manipulation during deployment and attacks during retraining. Note that even though most attacks are performed before or during training phase, the implanted backdoors are triggered at inference time. We refer to Section 2.1.1 for a detailed categorization of poisoning and backdoor attacks and to Section 2.2.1 for an overview and summaries of related literature. In the following, we detail the best practices and outline limitations of the defenses. Figure 1.6 gives an overview of the mentioned best practices and Figure 1.7 lists milestone papers in DL-related poisoning and backdoor attacks and defenses.

Use Trusted Sources Whether outsourcing model training to third parties, downloading code or pretrained models, or using training data from the web/online: Making sure that sources are known and trustful reduces the risk of getting malicious code (that e.g., optimizes also for an attacker-chosen subtask) as well as deliberately poisoned data and models. Ideally, the model development and training (from scratch) are performed under controlled conditions.

Random Data Augmentation While being a successful regularization technique, data augmentation can also help to destroy triggers in the poisoned training data.

Use an Auxiliary Pristine Data Set Additional training on trustworthy data makes data poisoning harder and should be applied.

Apply Detection Methods to Training Data and Model If the owner has access to the training data, checking the distribution of labels as well as applying outlier detection on the training instances can help to identify poisoned training data, which could in the next step be deleted from the training set or cleaned. Also, a visual inspection might be helpful, in particular so, if the data set is not too big and triggers happen to be not completely stealthy. Additionally, if also the model is accessible by the owner, clustering gradients, checking features [337] and neuron activations [68], performing spectral analysis [442], leveraging explanation methods on the outputs [105, 190] as well as computing statistics on the perturbation level needed for the model to change the prediction are means to identify poisoned/backdoored models [455]. In order to recognize the poisoned model a set of shadow models is trained and a meta-classifier is using the features extracted as training data to distinguish poisoned models from clean ones. Another step in this direction is universal patterns that allow to identify a poisoned model [221].

Clean the Model from Triggers There are multiple approaches to clean the model itself from the effect of the triggers/backdoors: special pruning by removing particular neurons [253], special retraining removing the effect of triggers (e.g., differentially private training [274]) that can then be used to identify poisoned samples or reconstructing the trigger and using this to retrain the model (with corrected labels [455, 452, 71]). Note that these approaches can be applied as from scratch, but also to clean an already poisoned model.

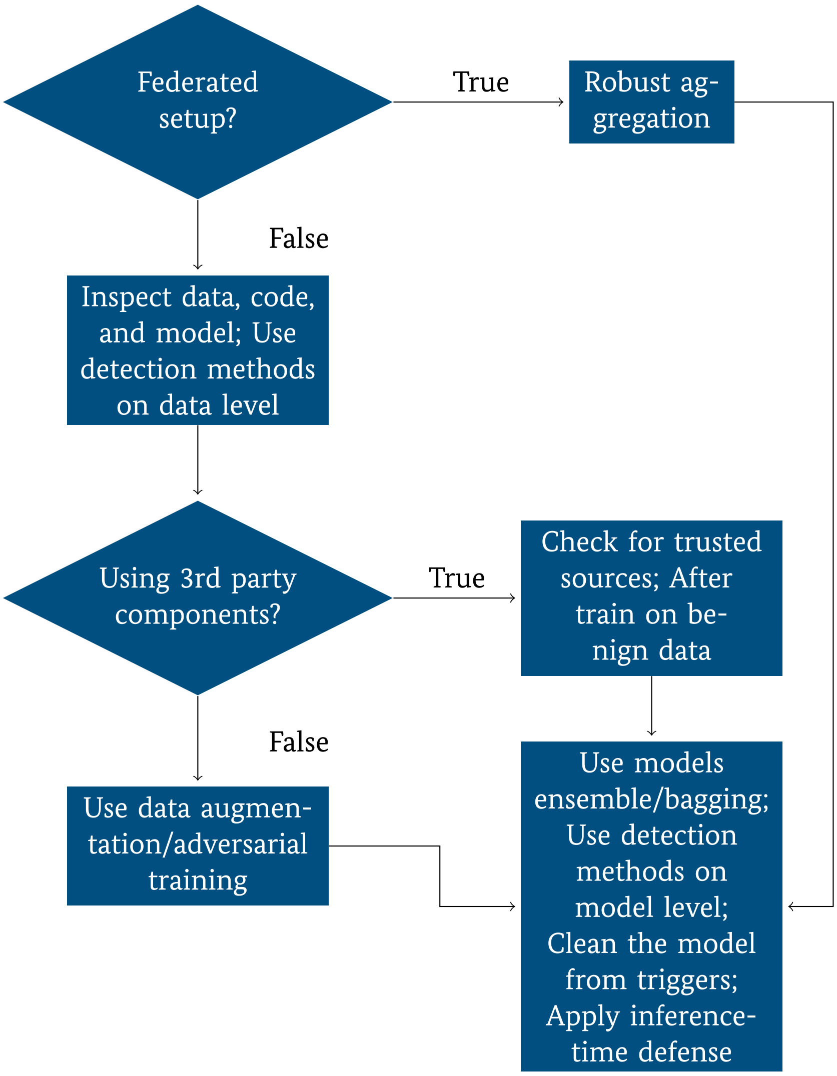

Federated setup?
True
Robust aggregation
False
Inspect data, code, and model; Use detection methods on data level
Using 3rd party components?
True
Check for trusted sources; After train on benign data
False
Use data augmentation/adversarial training
Use models ensemble/bagging; Use detection methods on model level; Clean the model from triggers; Apply inference-time defense

Figure 1.6: Best practices for defenses against poisoning attacks.

Adversarial Training for Protection Against Backdoor and Poisoning A special kind of training, similar to adversarial training against evasion attacks, can be used to protect the model from poisoning attacks [143]. This differs from the methods in the previous paragraph as it is a robust training method that aims at preventing poisoning attacks instead of retraining as a form of cleaning.

Detecting Triggers at Inference Time Unlike the detection methods that work offline by inspecting the training data prior to training, approaches like Februus [105] can work online and propose to clean the incoming data from triggers at inference time. Another approach is to mutate inputs and observe changes of predictions - if the prediction does not change in most cases of the mutations to the sample, the sample contains a trigger with high probability [139]. Also checking the labels of neighboring data instances (and the flow through the network) can unveil poison as it is usually expected to be close to the benign examples but deviate strongly from their usual activations inside the network [509].

Evaluate Robustness to Poisoning and Backdoors under Adaptive Attacks If possible, to try to evaluate susceptibility to backdoor and poisoning attacks by mounting adaptive attacks based on the state-of-the-art attack methods available. This gives a more realistic view and better shows potentially open vulnerabilities of the application than using standard attack settings (e.g., default values from implementation libraries, which might not work well for the given setting) [323, 138].

Robust Aggregation for Federated Learning The distributed setup requires separate measures to avoid the poisoning of the global model in the presence of malicious local learners. These methods are special aggregation algorithms, like taking the median, that are robust to outliers (poisoned models/updates). Another protection in such a setup is the detection of malicious participants [136].

Model Ensembling or Bagging Using multiple models in an ensemble for the same task can help to avoid wrong predictions - assuming that not all the models get poisoned in the same way. This is in particular applicable in the scenario of outsourcing model training to third parties or using pretrained models from online sources, [138]. Alternatively, bagging helps to train some models that are not poisoned.

Defense Ensembling Evaluations suggest that using several defenses at the same time (in form of ensembles, in particular when the defenses are orthogonal to each other i.e., addressing different aspects like detection based on data or model) can be beneficial, [323], even against adaptive attacks. The reasoning behind this is that adaptations to an attack making it more evasive against a specific form of defense can make it more easily detectable/susceptible w.r.t. some other defense method.

Continuously Protect against Poisoning and Backdoor Attacks The best practices mentioned above apply during the whole life cycle of an AI application: In particular, when retraining the model and doing model updates during the deployment stage or in online learning scenarios, account for the identified threats and take countermeasures that are detailed above.

Hardware Protection Make sure that the cyber security measures can also protect against hardware trojans that, e.g., manipulate weights of the deployed model (bit flips) on the machine. Note that this is also applicable for cloud computing.

Limitations

Attack Stealthiness The attacker usually faces a trade-off between stealth and effectiveness of the mounted attack. As using directly visible manipulations might lead to quick discovery of the attack, attackers might go for stealthiness, infecting e.g., only a small percentage of the data set and/or using e.g., clean-label attacks, where the labels are unchanged and perturbations often imperceptible to the observer - making the discovery of such attacks even harder for manual inspection. What is more, some attacks do not/hardly degrade overall model performance, so that this cannot be taken as reliable means of spotting poisoning [142]. Moreover, as attacks might be triggerless at inference time (clean-label attacks, which only manipulate training data), relying on only inference-time defense methods might not suffice.

Lack of Control over Data Collection and Federated Learning Many AI applications are trained on enormous amounts of data that is often scraped from the web. This makes it nearly impossible to validate the data quality and make sure that no intentionally harmful samples enter the training set. The same holds true for federated learning systems, where a global model is obtained from many local ones and a malicious local learner might remain undetected.

Side-Effects of Defenses Some defenses influence the training data/processes and lead to a performance drop of the defended model compared to the undefended one (e.g., for differentially private models, or for fine-pruned models [253]). Other approaches might detect too many false positives when searching for poison instances. Another limitation is the computational overhead of some methods (in particular those relying on many perturbations or those that require extensive training of e.g., generative models for inpainting). Together with the often huge amount of tunable defense parameters this might make it unfeasible/unscalable for the developer, in particular if resources are limited. What is more, some defense methods introduce rather high latency and are thus not suited for application online.

Methods Inapplicable for Federated Learning Some defenses do not require training data access, which might be beneficial in some cases. However, these approaches sometimes rely on reconstructing training data (model inversion) as (part of the) defense [71], which might violate privacy and could thus be infeasible, e.g., for the collaborative learning setup [138].

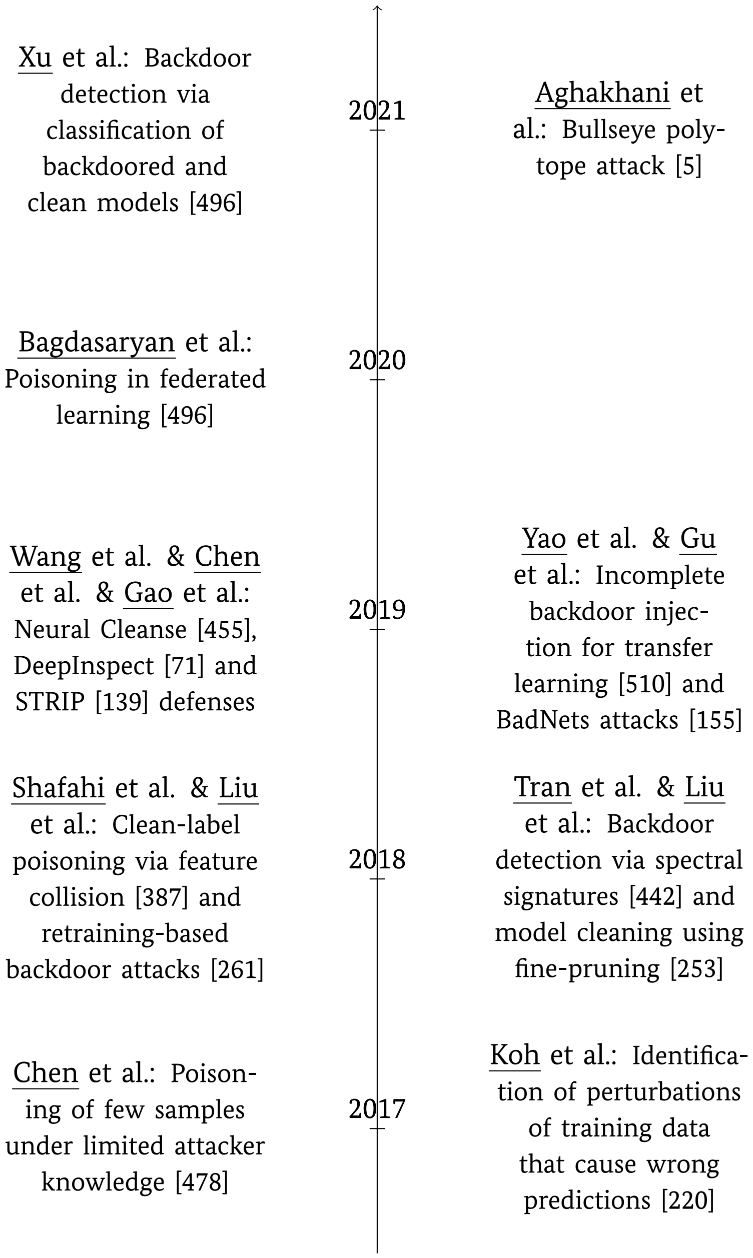

Xu et al.: Backdoor detection via classification of backdoored and clean models [496]
2021
Aghakhani et al.: Bullseye polytope attack [5]
Bagdasaryan et al.: Poisoning in federated learning [496]
2020
Wang et al. & Chen et al. & Gao et al.: Neural Cleanse [455], DeepInspect [71] and STRIP [139] defenses
2019
Yao et al. & Gu et al.: Incomplete backdoor injection for transfer learning [510] and BadNets attacks [155]
Shafahi et al. & Liu et al.: Clean-label poisoning via feature collision [387] and retraining-based backdoor attacks [261]
2018
Tran et al. & Liu et al.: Backdoor detection via spectral signatures [442] and model cleaning using fine-pruning [253]
Chen et al.: Poisoning of few samples under limited attacker knowledge [478]
2017
Koh et al.: Identification of perturbations of training data that cause wrong predictions [220]

Figure 1.7: Milestones in poisoning and backdoor for DL literature over the past years.

Restrictive Assumptions Certain defense methods make specific, sometimes restrictive assumption which pose limitations to the defense applicability and/or effectiveness. Concerning applicability: one common assumption is white-box model or data access [458], which might not be given if one wants to test ML as a service [496] or uses federated learning. For the latter case, having a clean validation set is also not always viable. Concerning the defense effectiveness: It happens that a particular trigger size and/or shape, the trigger's transparency, the number of triggers and also the number of classes affected is assumed for the defense [452]. Moreover, some approaches are designed for either class specific or class agnostic triggers. It is not always clear how crucial these assumptions are and whether the defense is also effective (or to what degree) if they are not met. To add to this, many of the defense approaches do not get evaluated w.r.t. adaptive attacks (in particular, cleaning the model from triggers could be bypassed by an adaptive attack), overestimating the protective power and potentially giving a false sense of security.

Federated Learning with Sybils Overall, when there are more than one Sybil [1] controlled by one adversary in a federated learning setup, it is very hard to protect the final ML model. Analogously, when the central cluster is malicious, defending the system becomes nearly impossible. Such situations should be prevented by strict general cyber security measures.

1.2.4 Defending against Information Extraction Attacks

In the focus of the information extraction attacks is getting access to the personal or copyright data of the stakeholders. The data, the model architecture and model hyperparameters are of interest to the attacker. The information about the data set that the attacker wants to gain might not be the full data reconstruction, but for example extracting biases in the data properties or identifying some particular interesting features. In different application domains, different parts are not to be exposed: in the medical domain patients would not want to expose their disease history, for cutting-edge application developers it is important to keep model architecture safe, for companies that invested a lot of resources into training a high-performance model it is not desirable to share its weights and hyperparameters. It should be noted that very often model stealing (or architecture stealing) serves as a stepping stone for generating adversarial attacks later.

Data reconstruction, membership inference, or features inference are complicated attacks that require a lot of resources (in the centralized case). The most widespread approach for such attacks is to train many shadow models that will to some extent duplicate the target model (depending on the knowledge of the attacker the shadow models are more or less similar).

Another important aspect of privacy attacks is that they can be performed in the training phase, when an attacker can get access to model updates. The most straightforward way of the attacker to know the model updates is to be a part of federated learning setup. At the same moment, data reconstruction is easiest with generative models – since it is the exact knowledge the model should have learned.

[1] Sybil attacks subvert a service's reputation system by creating a large number of pseudonymous identities to gain a disproportionately large influence.

We refer to Section 2.1.1 for a categorization of privacy attacks and to Section 2.2.4 for related literature. An overview of milestone research done in the area of privacy attacks and defenses can be found in Figure 1.8. In the following we list the most prominent groups of defenses against general privacy attacks, which are also summarized in a flowchart illustrated in Figure 1.9.

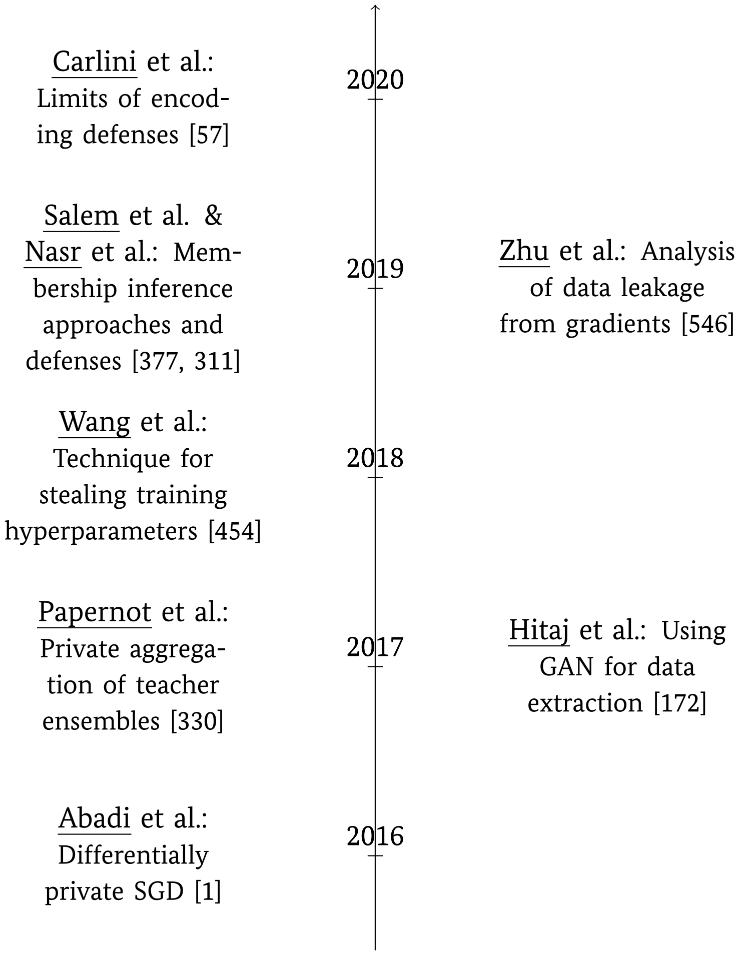

Carlini et al.: Limits of encoding defenses [57]
2020
Salem et al. & Nasr et al.: Membership inference approaches and defenses [377, 311]
2019
Zhu et al.: Analysis of data leakage from gradients [546]
Wang et al.: Technique for stealing training hyperparameters [454]
2018
Papernot et al.: Private aggregation of teacher ensembles [330]
2017
Hitaj et al.: Using GAN for data extraction [172]
Abadi et al.: Differentially private SGD [1]
2016

Figure 1.8: Milestones in privacy for DL literature over the past years.

Data Sanitization While it is not a universal defense, it is an absolute must to remove all the sensitive parts of the data before using it for training and thus making it impossible for intruders to extract such sensitive data from the trained model. On the other hand, in the case when sensitive data should be learned, different mechanisms of protection should be used.

Decrease Information in the Output and Gradients Since most privacy attacks use model outputs (confidence scores), a viable defense is to reduce information contained in the output vectors (for the case of classification task with softmax output): output only k-top class predictions, increase entropy in the prediction vector (by temperature scaling), or coarsen prediction. In case of a federated learning setup an attacker can infer data from the exchanged gradients – thus clipping gradients can be a possible countermeasure.

Avoid Overfitting Mostly, the ability of a neural network to leak information about the data used for training is attributed to memorization and overfitting. Thus, attacks directed on extracting membership or data properties information will benefit from overfitted models. Thus, regularization techniques, such as dropout, are helping against information extraction attacks. In particular cases knowledge distillation also helps to protect the original model.

Differential Privacy This technique is one of the most used and effective data privacy protection approaches. Initially, it is aimed at the protection of the corporate data that has to be shared, but it should not be revealed for each individual record (thus breaching privacy of the owner of the record). The core idea is to add noise ϵ to the shared information and allow access to the information only within the limits of a privacy budget for a particular user. The budget defines how many times the user can make a request. The mathematical definition of differential privacy is as follows: A randomized algorithm K gives ϵ-differential privacy if for all data sets D and D' differing on at most one row, and any $S \subseteq \text{OutputRange}(K)$, $\Pr(K(D) \subseteq S) \leq \exp \epsilon \times \Pr(K(D') \subseteq S)$. There are multiple implementations of the differential privacy approach in practice, e.g., k-anonymity [379], l-diversity [275], m-invariance [483], etc. When applied to machine learning algorithms, differential privacy serves as a form of regularization (by adding noise) and thus (sometimes) leads to better generalizable models. In particular, examples of the approaches of integrating differential privacy into deep learning are: differentially private SGD, when noise is added to the SGD updates [1]; differentially private aggregation mechanism from multiple models trained on private data [330]; adaptive Laplace mechanism [339]. Note that differential privacy is aimed on data protection and not model or architecture protection.

Homomorphic Encryption Fully homomorphic encryption allows to train a machine learning model on the encrypted data [144]. Nevertheless just encrypting data instances is not enough [57] – only fully encrypted neural networks can provide enough protection [170]. Another way to use encryption is to apply it on the gradients in a distributed (federated) learning setup, thus preventing gradients leakage [340].

Stateful Queries Check Since stealing of the data requires the attacker to make a multitude of queries, checking such queries can protect privacy. For example, [76] propose to check the similarity of the queries made in a row to identify an attacker. Here, it should be distinguished which goal the attacker has: when it is data stealing, the queries should be checked on similarity with each other, while when it is model stealing the queries should be checked on being out-of-distribution for the original training set [18].

Hardware Level Defenses Several stealing attacks that are using internal hardware characteristics exist, such as RAM usage during inference, identifying load of the exchange channels and so on. Such attacks of course assume that the adversary has access to the server. This is in fact realistic if we consider a scenario of machine learning provided as a service. In these cases some techniques like RAM that encrypts addresses or injecting dummy read/write operations to mislead the attacks might be useful.

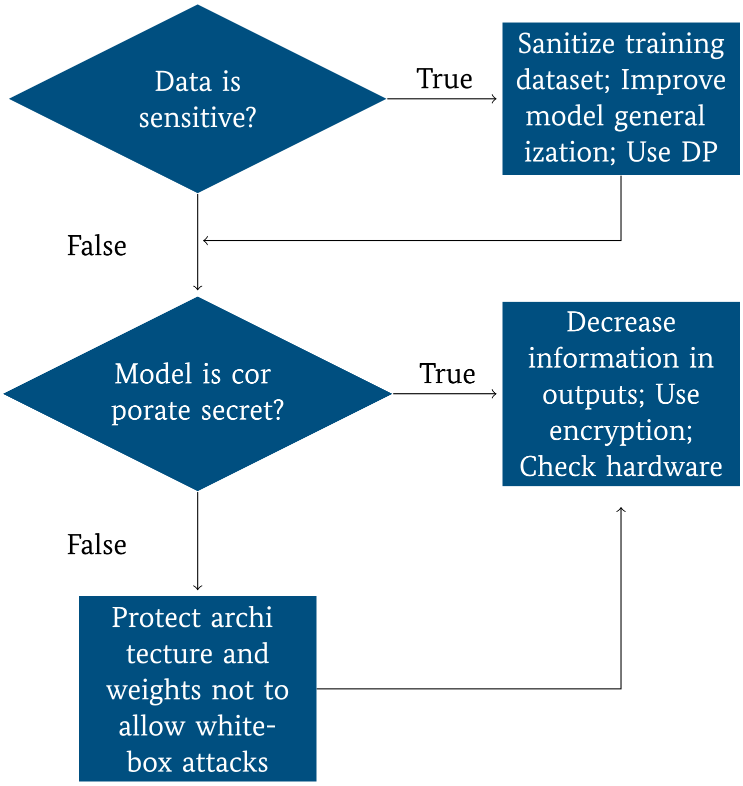

Data is sensitive?
True
Sanitize training dataset; Improve model general ization; Use DP
False
Model is cor porate secret?
True
Decrease information in outputs; Use encryption; Check hardware
False
Protect archi tecture and weights not to allow white-box attacks

Figure 1.9: Best practices for privacy defense.

Limitations

Inability to Theoretically Guarantee Performance of a NN The absence of the theoretical understanding of generalization abilities of NNs limits the ability of a developer to protect the model from leaking information. Moreover, being non-transparent and non-interpretable, a state-of-the-art NN cannot be analyzed for finding out the memorized data points. The success of known knowledge extraction attacks can serve as a metric for understanding the degree of vulnerability of a model.

Drawbacks of Differential Privacy While being a state-of-the-art privacy protection mechanism, differential privacy can prevent a model from learning to predict correctly, when only a few samples are given – simply due to its mechanism that does not allow to get access to precise individual data. Also, differential privacy requires identifying the privacy budget, that regulates the amount of noise added for protection. It is very hard to regulate which amount is better, because it is possible to transform the data to complete noise – which will guarantee the protection, but make training or inferring impossible. Also, it should be taken into account that additional, unprotected data weakens the effect of differential privacy [308].

Computational Overhead Introduction of privacy mechanisms to the system means computational overhead. The most effective fully encrypted DL [170] is especially expensive.

1.2.5 General Limitations of the Defenses

Along with the limitations discussed in the individual chapters, some general limitations apply. Usually, the benefits of the defenses outweigh the costs. Based on the compiled threat model some risks may be acceptable in the AI development.

Most Defenses are not for Free Based on the defense, significant computational overhead may be introduced. These additional computation steps may be applicable during training, but could also affect inference in case of, e.g., external add-ons. AI practitioners should carefully weigh the costs of the respective resilience gain.

Careful Parameter Tuning is Needed As AI models themselves, defense methods need a carefully selected set of hyperparameters for adequate performance. The resilience improvements should be reassessed with each defense method applied as a concatenation of multiple defenses may influence the performance of each other. As side effect, the parameter tuning may increase the cost on a qualitative validation data set.

Adaptive Attacks will still be Successful Adaptive attacks evaluate the robustness of AI systems against omniscient attackers, i.e., incorporating all defenses available. Due to the absolute knowledge, these attacks will be successful eventually. However, the applied defense methods may increase the attack effort, be it in higher computations costs or more attack budget needed.

Defenses May Counteract the General Performance Attack resilience may come at the cost of general performance degradation. Especially for evasion attacks, the trade-off between robustness and accuracy has been studied [445]. It is advisable to balance between both goals based on the expected risk of the overall AI system.

Most Methods Were Evaluated in the Image Domain Defense methods are mostly developed for the vision (image) domain and might not be transferable to other setups [455]. However, attacks are also a threat for e.g., the text, audio, video and network security domain as well as reinforcement learning, among others. This is an open challenge and more research is needed to

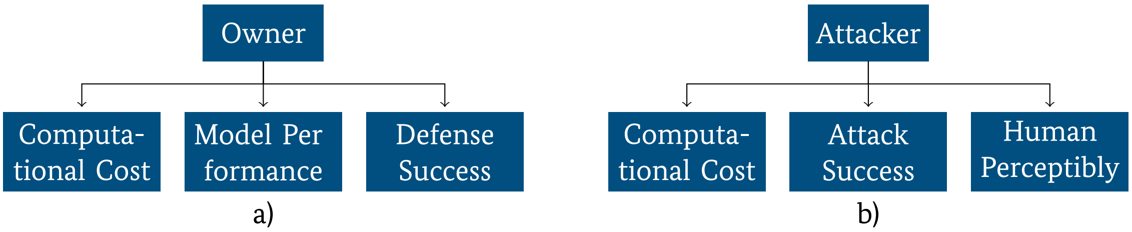


Figure 1.10: (a) Metrics that describe the success of the protection techniques used in the system: adversarial robustness, overall final accuracy, computational cost of the protection techniques, etc. (b) For an attacker the metrics are usually inverse of the ones of the owner, but they also can be unique (e.g., imperceptibility).

defend effectively, given the fast evolving attack approaches. In general, the field of adversarial ML is fast-paced and methods get outdated or broken quite quickly – making it challenging to stay up to date and requiring continuous adaptations of the methods. Furthermore, many defense approaches require expert ML knowledge.

1.3 Combining Defenses for Overall Resilience

In order to outline the exact approaches that help to protect an AI system, one has to align and adapt multiple techniques, such as adversarial training, attack detection mechanisms, or a reliable embedding of models. The vastness of the research area and capabilities of different protection and attack mechanisms, especially their constant and fast-paced development, make it necessary to be flexible when looking for the most suitable protection of a newly built AI system. Another important aspect that is not emphasized in the current research is the interaction between different kinds of defenses and attacks: for example, will poisoning change the sensitivity of the network to evasion attacks? Will detection techniques against adversarial examples help against poisoned inputs as well?

One of the possible perspectives that help to arrange all the techniques related to adversarial ML is to group them by the stages of the life cycle of an AI system they can be applied in, as we discussed in Section 1.2. This allows to identify the sequence in which different techniques should be developed and applied. The experimental framework described in Chapter 3 is based on a pipeline structure which allows evaluating the performance of attacks and defenses as well as exploring the interaction between them. In the following we give a high level overview of the findings of Chapter 3 to underline the complexity of combining different defense approaches. For a more detailed explanation and implementation details, we refer the reader to Chapter 3.

First, one has to select metrics according to which the success of the techniques will be evaluated. One of the most important metrics with respect to defenses is for example the level of robustness as discussed in Section 1.1.3. In Figure 1.10 we give an exemplary set of metrics that can be used as success evaluation for the owner of the system and for the attacker, who is trying to maliciously interfere with the functionality of the system.

Once we identified metrics of interest, the next step is to assemble the technologies related to the affected stages (e.g., we should identify if we are interested in the training stage attacks or only in the inference stage). The general pipeline framework developed in this project allows for inclusion and exclusion of stages. For example, a network trained once can allow for multiple different investigations of the inference stage without having to repeat the training stage that is not of interest for the investigation.

Such a framework opens wide possibilities for finding a setup that is optimal for the particular selected network, data set and metric. Adding other metrics into consideration can help to understand the trade-offs between techniques and benefits of using them. For example, in our experiments (Chapter 3) we were identifying setups for two tasks – the classical CIFAR10 problem [223] and a COVID-19 recognition on chest X-rays [353, 86]. Clear challenges are to find the hyperparameters of the attacks and defenses that are most effective for a chosen data set (while CIFAR10 is a common data set in scientific publications and thus some hyperparameters are already tested, the COVID-19 data set will require full tuning). It is interesting to observe that different setups of adversarial training can still help against adversarial attacks, that are based on other imperceptibility metrics. At the same moment, combining multiple defenses might also be detrimental – but for a certain statement more research should be done. Finding an optimal attack strategy also requires taking into account multiple criteria apart from the attack success: how long can the computation last, how imperceptible should the attack be, etc. So our framework indeed reflects the combinatorial space of the attack-defense game. Important questions of the possible effect of poisoning on adversarial robustness and adversarial training on backdoor resilience require long and careful evaluations, yet our initial experiments show that the techniques indeed have an effect on each other.

As an outline, for this global view, we suggest to use an evolutionary algorithm to identify a suitable combination of techniques and parameters. Our pipeline implementing different techniques and differently aligned/tuned methods serves as a population, while the fit function takes into account all metrics we are interested in. Like this we can avoid a brute-force search through all the possible setups, but guide it with the aim of improving metrics. The results of such evaluations can be summarized in Pareto fronts. Since attack/defense combinations are compared with each other in multiple conflicting dimensions, we want to observe which combinations are optimal with respect to a given scenario. “Attackers and owners may then choose a trade-off along the Pareto frontier that fits their constraints, e.g., an available computational budget” (see Chapter 3). Overall, this leads to outcomes like: “For the best adversarial protection on data set X, one should use training procedure Y for the base model with parameters Φ^Y and apply defense method Z with parameters Φ^Z” (see Chapter 3). A challenge that obviously still remains is the exponential search space and the computational cost of an evolutionary optimization through all possible configurations.

1.3.1 Possible Intersections between Research Directions

Besides technical interactions between defenses and different types of attacks, there is also possible exchange of techniques from different areas of research that are not in the focus of the current scientific publications.

Poisoning and privacy attacks can act as defense approaches against each other – “poisoning” a

NN with a watermark will help to prevent the stealing of a model; honey-pot poisoning can help to identify evasion attacks; reverse engineering of the training data set can help to find possible triggers and thus detect poisoned samples [265].
One should also keep in mind that badly tuned adversarial training can turn out to be poisonous, resulting in the NN having a high error rate [387]. At the same moment, poisoning examples are sometimes very close to the universal adversarial attacks, so the defenses can be interchangeable. Even though poisoning is mainly perceived as an attack for lowering accuracy or injecting backdoors, it can also be directed against adversarial certification, so the poisonous examples will lead to the inability to certify the model against adversarial examples. Simultaneously, research on certification is currently expanding into the direction of other than evasion attacks [466].

Chapter 2

Literature Overview

L. Adilova, *Fraunhofer IAIS*
F. Assion, *neurocat*
Dr. K. Böttinger, *Fraunhofer AISEC*
V. Danos, *TÜViT*
Dr. J. Firnkorn, *neurocat*
S. Jacob, *BSI*
F. Langer, *TÜViT*
T. Markert, *TÜViT*
Dr. M. Poretschkin, *Fraunhofer IAIS*
J. Rosenzweig, *Fraunhofer IAIS*
J.-P. Schulze, *Fraunhofer AISEC*
P. Sperl, *Fraunhofer AISEC*
B. Srinivasan, *neurocat*
Dr. H. Trittenbach, *neurocat*

In this chapter, we give an overview of the field of adversarial ML by i) categorizing attacks and corresponding defenses in Section 2.1.1 and Section 2.1.3, respectively, and by ii) summarizing major publications in the field in Section 2.2.

2.1 Taxonomy of the Literature

We describe the taxonomy for categorizing the reviewed literature on topics in adversarial ML, expanding the overview of Section 1.1.2. We start with outlining the attacker goals before we present the respective attack types and defense methods.

2.1.1 Attacks on AI Systems

Attacks are designed to reach a particular goal. Thus, we introduce and categorize attacks by their individual goals: manipulation of the model's training, influencing the model's output, or stealing information. Based on the reviewed literature, we outline a scheme of the attacker goals in Figure 2.1 and in the following subsections introduce the corresponding attack types.

Evasion Attacks

Evasion attacks are performed by creating malicious inputs at inference time causing failures of the DL system. These inputs, referred to as adversarial examples, are specifically crafted and mainly aim at provoking misclassifications, reducing confidence, or spoofing robustness certificates. An overview of the types of failures an attacker can cause is depicted in Figure 2.1. Misclassifications can be either targeted, i.e., the output should be of a specific class, or untargeted, i.e., the output should be any other class than the correct one. Confidence reduction means that the input causes the DL model to be less confident (in terms of class probability) in classifying the current sample. Finally, in certificate spoofing attacks, adversarial examples are generated even though the robustness of the model is assumed to be guaranteed. Apart from the goals mentioned above, attacks can also be performed in a specific use-case-related manner: in case of object detection, the goal can be to make objects appear or disappear, while in reinforcement learning the malicious agent could follow an adversarial policy to fool the benign agent.

Adversarial examples themselves can be categorized according to several orthogonal dimensions (Figure 2.2): human perceptibility, whether the attack is universal, attacks in the physical or digital space, and required model access. Human perceptibility refers to the attack's ”appearance“: can it be spotted by a human observer as being a maliciously manipulated input or not? This category can be split in two sub-parts. The first one comprises evasion attacks that are imperceptible to humans. This includes geometric transformations (crafted on purpose to mislead the DL model), semantic attacks and epsilon perturbations within a sufficiently small perturbation budget. Semantic attacks make use of meaningful concepts instead of manipulating (individual) pixels as it is the case in epsilon perturbation attacks. The latter can be realized by either constraining the allowed distance from the benign example to the adversarial example, or by minimizing the distance mentioned. The second part are attacks that are perceptible to the human observer. These again include semantic attacks, patch-based attacks, and classifiable noise. In contrast to imperceptible attacks, human observers will see clear manipulations compared to benign inputs, possibly while retaining semantic information. For patch-based attacks in the image domain, a connected region of pixels is manipulated in either a semantically meaningful way or with optimized noise levels. Note that epsilon-perturbation attacks with a high perturbation budget belong to the group of perceptible attacks.

Evasion attacks can be universal in a sense that the adversarial perturbation is sample-agnostic (or class-agnostic). That means that the perturbation is not crafted for a specific input sample but rather is successful in fooling the DL model when applied to a wide range of different input samples.

Moreover, we distinguish between attacks performed in the digital or in the physical domain.

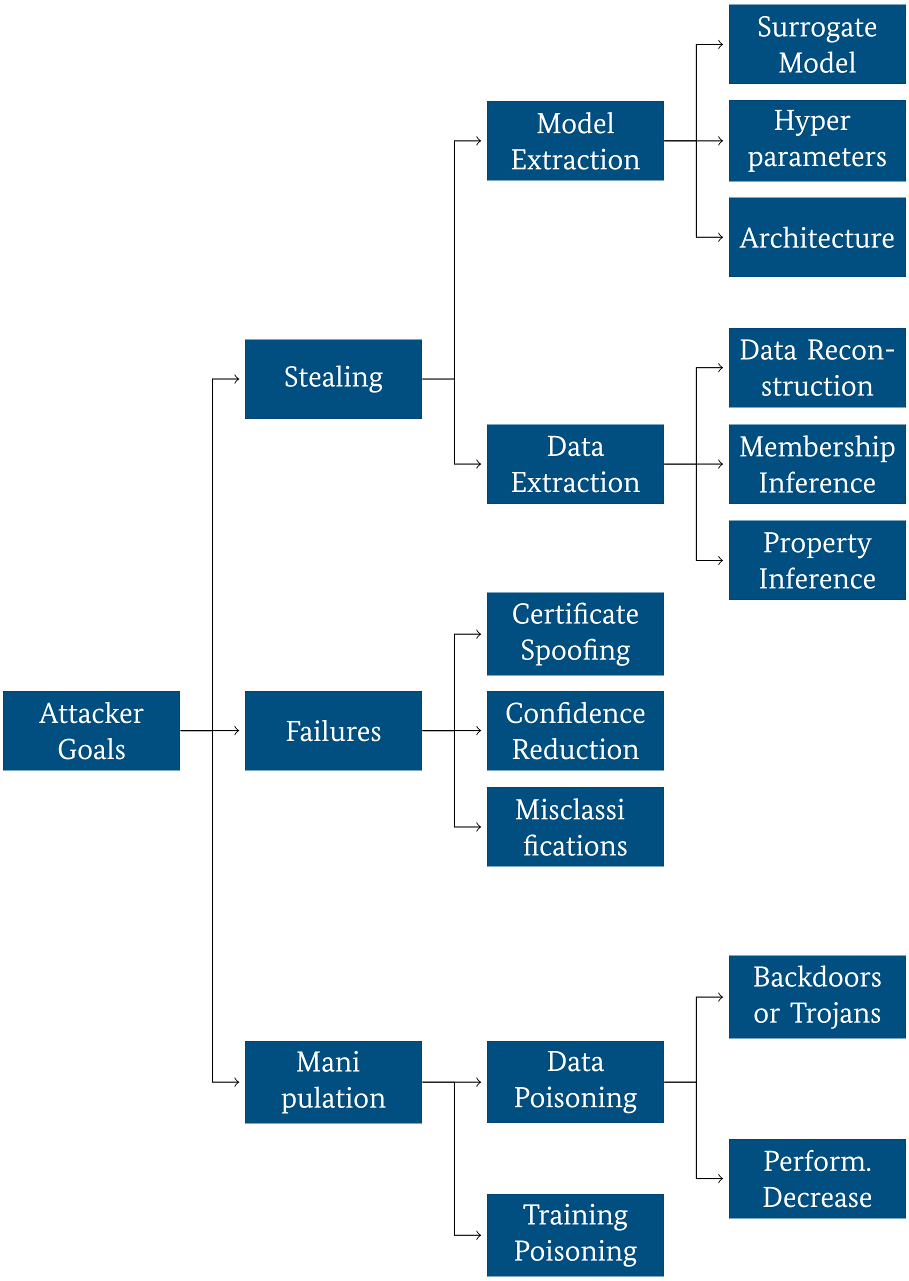


Figure 2.1: Taxonomy of attacker goals.

Physical attacks work across an air gap, e.g., when the AI system uses sensors to observe its surrounding. In contrast, digital attacks assume direct access to the AI's input on a software level. Finally, the knowledge an attacker needs to perform an attack is an important category: while some attacks require full model (and/or data) access –- so called white-box attacks –- others can also be performed with less or no knowledge/access (gray-/black-box). The latter are brute-force attacks, transferability-based attacks, gradient approximation approaches and alternative adversarial optimization objectives. Transferability-based attacks usually employ surrogate models that are trained for the same task as the target model. The more knowledge the attacker obtained about the original model, e.g., its training data, the more likely the transfer attacks work. Gradient approximation schemes are based on query-efficient sampling approaches to get useful gradient information for crafting malicious inputs based on oracle access. Alternative optimization schemes allow to obtain an adversarial example without the knowledge of gradients and parameters.

Poisoning & Backdoor Attacks

An attacker uses a poisoning attack to manipulate a model's behavior. This could be to diminish the overall performance of a model, or to make it consistently misclassify examples belonging to particular ground-truth class.

Poisoning attacks are initiated in the training stage of a model, where an attacker modifies parts of the training data by mislabeling and/or adding noise to it. The discussed research tries to severely impact the DL model's performance with minimal changes in the training data. Alternatively, the attacker tampers with the training procedure in order to indirectly poison some inputs. This could compromise private learners and allow the attacker to extract private data or model information in later stages.

A specific kind of targeted poisoning attacks are backdoor attacks. An attacker adds a specific pattern or trigger to some training samples to insert a "backdoor" into the model (see Figure 2.3). The backdoor is a hidden association in the model. At the inference stage, the backdoor is activated when the trigger is presented in a new example, causing the model to behave in the attacker's desired way. The model behaves as intended in the presence of benign or trigger-free examples.

In digital deployment settings, a trigger could be a fixed patch, noise or image overlayed on a training sample. A backdoored model deployed in the physical world, such as a smart security camera, could be triggered when a person wears a specific accessory, such as eyeglasses. We will also present triggerless backdoor attacks, in which an attacker implants a backdoor by manipulating the training process, rather than the model input.

AI systems with learning strategies that rely heavily on third-party or external sources can be especially vulnerable to poisoning attacks:

- In the federated learning setup, a model might be subject to a data poisoning attack from multiple data origins.
- When using transfer learning, backdoors can transfer from the pre-trained base model that is retrained for a new task; traces of the backdoor can be removed during fine-tuning,

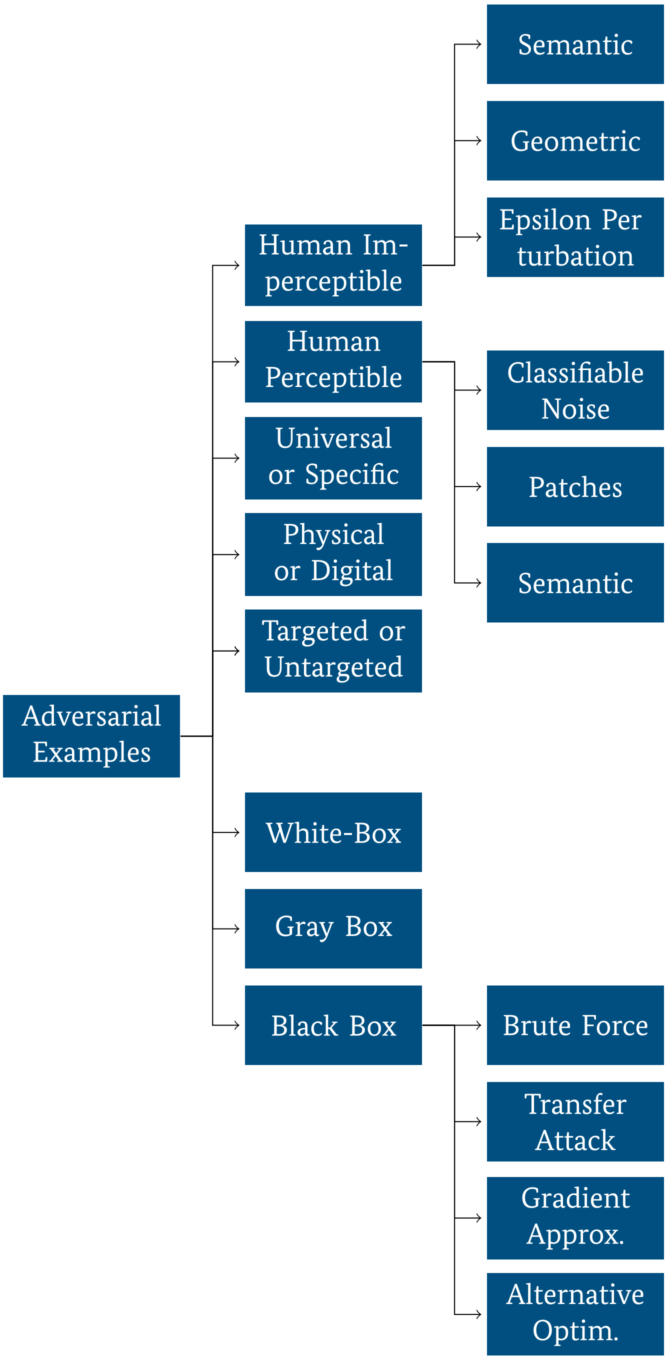


Figure 2.2: Taxonomy of adversarial examples.

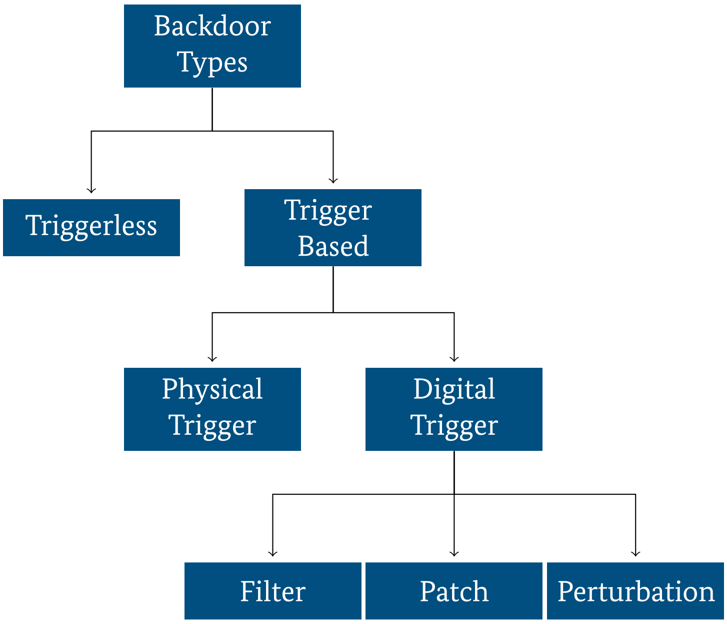


Figure 2.3: Taxonomy of poisoning triggers.

making them difficult to spot.

- The continuous/online learning setup allows a user to update the deployed model in order to accommodate changes in data sets, environment and task. This opens the door to poisoning attacks during the re-training process.

Privacy Attacks: Model and Information Extraction Attacks

The goal of an attacker who aims to perform a privacy attack is stealing some of the private information – either about the data that was used to train the model or about the model itself including training parameters, architecture, or weights.
Two types of privacy attacks can be distinguished: data attacks and model attacks. Model extraction attacks can violate business privacy of the company. E.g., if the company created a model architecture or performed a successful training and found a set of weights that performs optimally on the task. All of the aforementioned knowledge is a corporate secret and should not be used without the owners' allowance. Also, if a model is stolen, stronger attacks can be performed compared to a purely black-box scenario.
Data extraction attacks can be grouped into the following types:

- Data point membership inference – most investigated type of data privacy attacks aimed at identifying data samples that were used for training the model.
- Data property inference – identifying some general property of the training data, e.g., finding the ratio of particular classes in the training data set or particular properties of the data

points.

- Training data reconstruction attacks, which can aim both at the reconstruction of entire training samples or at the inference of some private features interesting for the attacker.

The idea of most of the data privacy attacks is a creation of a set of shadow models, which further allow to train meta-classifiers to perform targeted attacks. Privacy attacks are thus closely related to model stealing attacks, where attackers use the obtained knowledge to fully reconstruct the DL model.
Extended research on specialized applications includes federated learning, generative models, and Natural Language Processing (NLP) settings. For federated learning, research aims at keeping private data of individual learners intact. Generative models are especially vulnerable as they are intended to learn the training data distribution. NLP models are based on sensitive private data, often from households sharing their voice voluntarily or even involuntarily to voice assistant systems.

2.1.2 Certified Robustness

As presented above, DL models are vulnerable to attacks that could lead to fatal failures. To ensure the robustness and therefore the safety of such systems, robustness verification methods have been proposed in research. The aim of all these methods is to derive bounds on the robustness of the NN model. In the following section, we give an overview on the taxonomy of such methods. However, due to the non-linearity and high complexity of DL models, a formal mathematical verification is often infeasible. Therefore, recently a lot of research has emerged to overcome these challenges. As discussed in Section 1.2.1, in this document we distinguish between certification and verification, even though in the literature these terms are used unanimously. Here, verification refers to methods that aim to derive a formal verification, i.e., complete methods, and certification refers to methods that derive the robustness bounds through approximation or probabilistic measures. The techniques can be further categorized based on the underlying mathematical concept used to formulate the bounds. Figure 2.4 gives an overview of the different categories.

Complete Methods

Complete methods derive robustness by exhaustively searching if there exists an adversarial example within a specific perturbation space. Therefore, if the result of a complete method yields that the model is not robust, it is guaranteed that there exists an adversarial example within the perturbation space. There exist two main concepts for complete verification: solver-based methods and Branch-and-Bound methods. Solver-based methods first encode the non-linear functions into feasibly solvable problems such as SMT or MILP and solving them accordingly. Another approach are branch-and-bound methods, where the problem is branched based on the piecewise-linear property of the activation function in DL models and for each of the branches an upper and lower bound on the robustness is computed. This way the bound for the entire model can be derived. Since these methods exhaustively search the perturbation space to

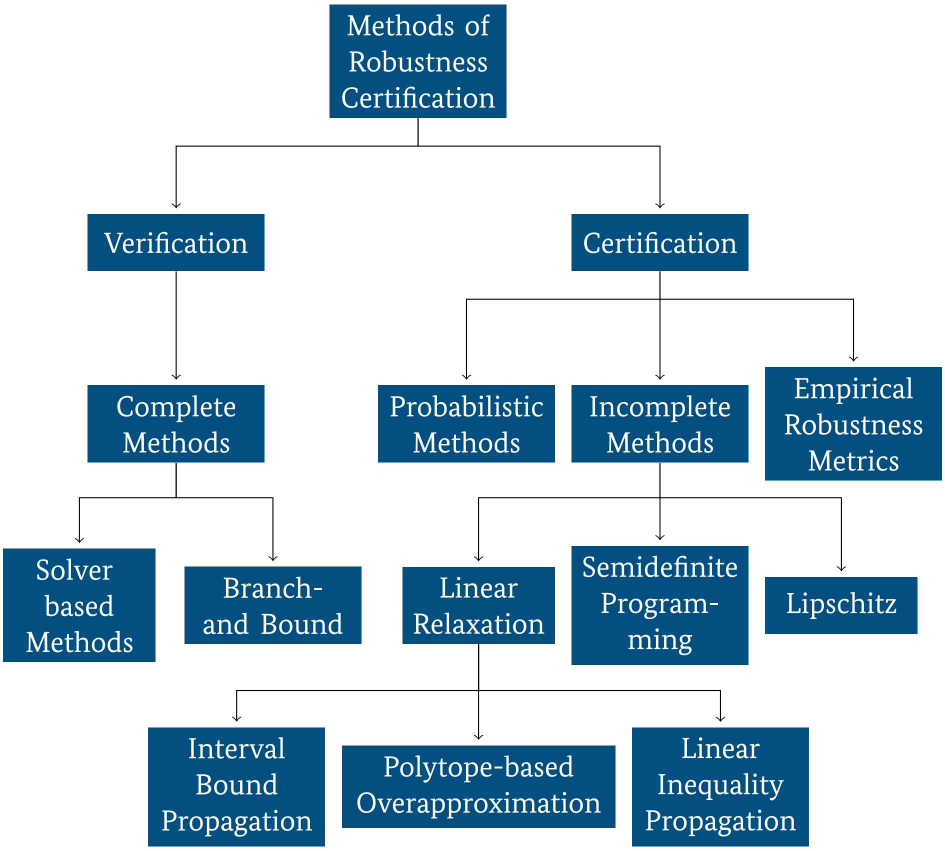

Methods of Robustness Certification
Verification
Certification
Complete Methods
Probabilistic Methods
Incomplete Methods
Empirical Robustness Metrics
Solver based Methods
Branch- and Bound
Linear Relaxation
Semidefinite Program- ming
Lipschitz
Interval Bound Propagation
Polytope-based Overapproximation
Linear Inequality Propagation

Figure 2.4: Taxonomy of certification.

form the robustness bounds, these methods are computationally expensive and only feasible for smaller, less complex NN model. In addition, these methods are built on the piecewise-linear property of activation functions, with most of them even specifically designed for ReLU activations.

Incomplete Methods

To overcome the scalability issues of complete techniques, incomplete methods were introduced. As discussed above the biggest challenge of robustness verification for NNs stems from their non-linearity. Incomplete methods aim to overcome this non-linearity by overapproximating non-linear layers and deriving bounds for their robustness. In contrast to complete methods, these methods may produce too conservative robustness bounds. Meaning there is no guarantee that if the result of the certification yields a non-robust result, an adversarial example actually exists. Inherently, these methods provide less tight bounds than complete verification methods. The methods can be further categorized into linear relaxations, semidefinite programming, or Lipschitz-based certification.

Linear relaxations aim to relax the non-linear layers and overapproximate bounds for them. The overapproximation is then propagated through the entire NN model resulting in an approximate bound on the robustness of the entire model. Examples for these methods include Interval Bound Propagation (IBP), Polytope-based overapproximation or Linear Inequality Propagation. Just like complete methods, linear relaxations rely on piecewise-linear activation functions and often solely on the ReLU activations.
An additional method aiming to overcome the non-linearity of NN models are semidefinite programming approaches. These approaches encode the piecewise-linear activation functions as quadratic constraints and formulate finding robustness bounds as a semidefinite programming problem. Through this encoding, the robustness bounding becomes a convex problem and can be solved.
Finally, Lipschitz-based certification methods, aim to calculate the global Lipschitz constant binding the robustness of the model. Since the global Lipschitz constant can be quite loose as a robustness bound, some of the approaches improve its tightness for example by combining it with other incomplete methods (e.g., IBP) or utilizing local Lipschitz bounds.
Additionally, there are hybrid approaches that combine concepts from complete and incomplete methods. They leverage the tightness of complete verification techniques in combination with the reduced computational effort from incomplete methods. Therefore, they aim to provide a tighter bound approximation with a higher scalability than complete verification techniques.

Probabilistic Methods

Since incomplete and complete methods are defined along specific NN layers, they are not applicable to a large number of architectures. Due to this limitation and their lack of scalability, probabilistic methods emerged. As explained in Section 1.2.1, probabilistic certification methods provide a probability that the smoothed model is robust within a specific perturbation space. Since the certification is carried out on a smoothed model, probabilistic certification offers a robustness guarantee for a variety of different model architectures and a higher number of layers. However, they provide looser bounds than complete and incomplete methods.

Empirical Robustness Metrics

Finally, empirical robustness metrics are the most flexible methods to derive robustness guarantees. They estimate the robustness of the model by evaluating its performance on attacked or otherwise perturbed inputs. The set of manipulated inputs and attacks sufficient for testing the robustness of the model has to be derived for each use case individually. However, they provide only a limited robustness guarantee towards the attacks and the threat model that was considered for the estimation. Therefore, they provide a statistical measure, which has to be treated as such based on the number and quality of tests that were performed.

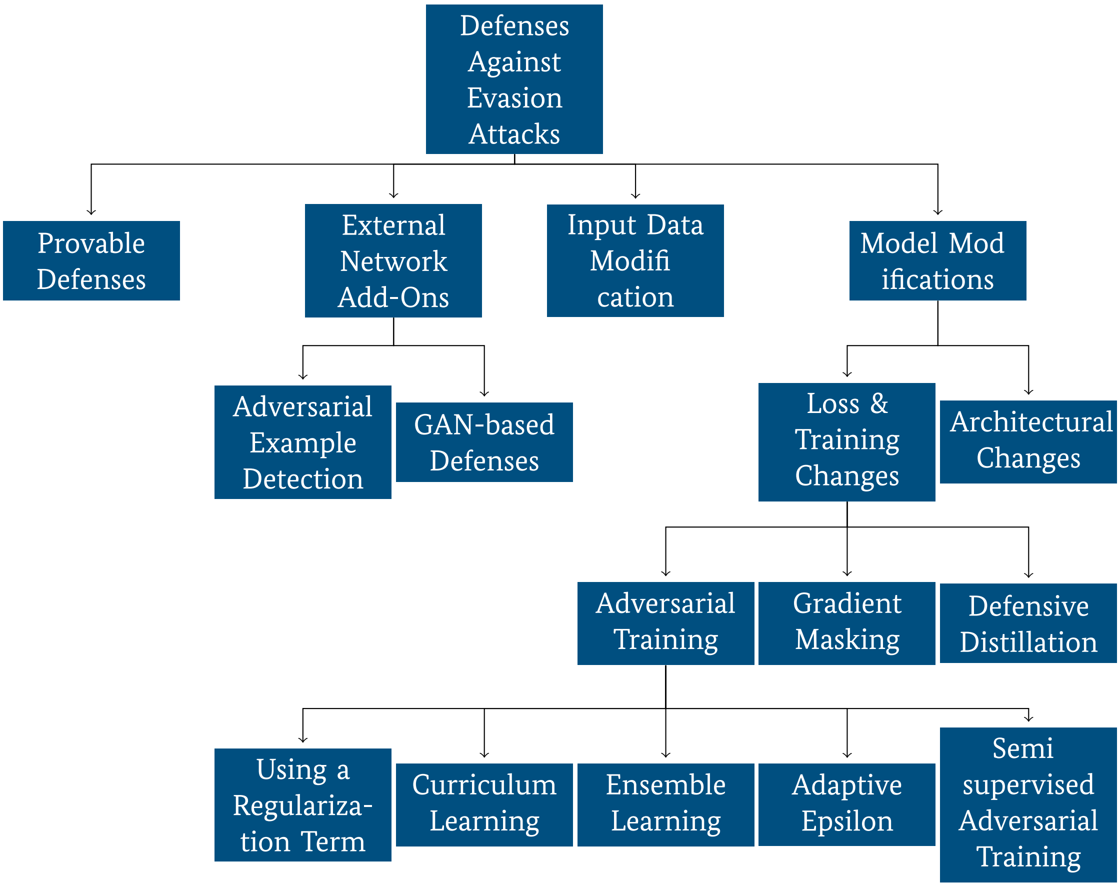


Figure 2.5: Taxonomy of evasion defenses.

2.1.3 Defense Methods

Defense methods increase the attack effort an attacker needs to invest to mount successful attacks. In the scope of this document, we discuss defenses against evasion, poisoning and backdoor attacks as well as model and information extraction attacks. We thoroughly compiled relevant sub-categories of each defense category and summarize our taxonomy in Figure 2.5 and Figure 2.6.
As done in every IT system, standard IT security best practices must be applied at all times. Examples of cyber security measures are the restriction of unknown user access and standard measures to protect the infrastructure in which the models are deployed. We refer to specialized literature on IT security and focus on DL-related defenses in this document.

Defense Methods against Evasion Attacks

As discussed in Section 2.1.1, evasion attacks alter the output of NNs by small additive changes to the input. We divide the defenses against such evasion attacks into five categories. In Fig-

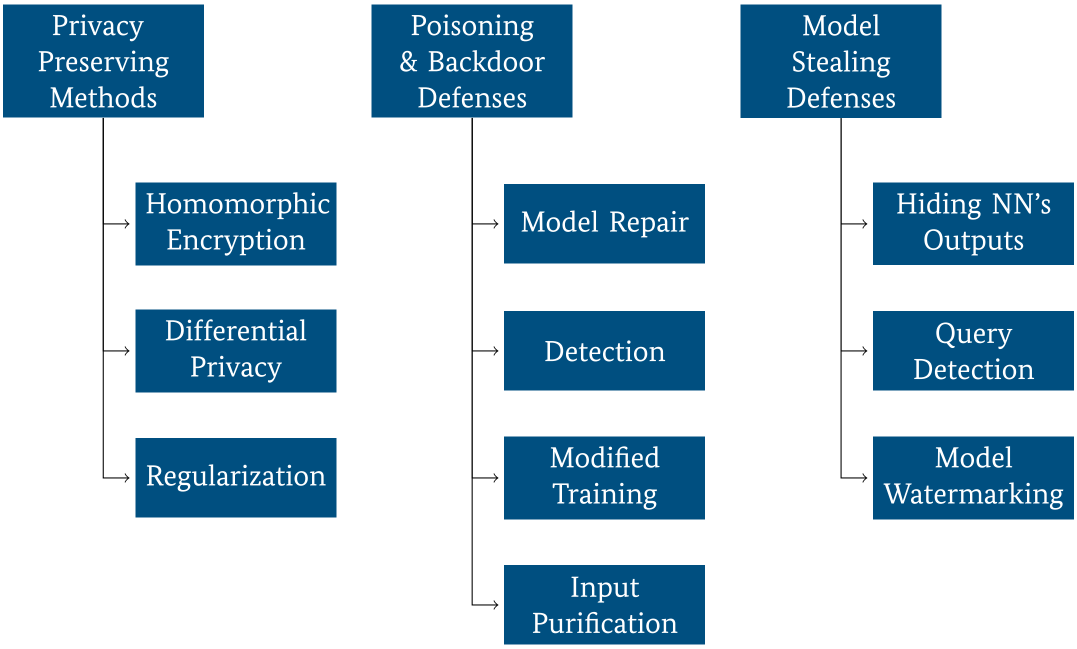


Figure 2.6: Taxonomy of privacy and poisoning defenses.

ure 2.5, we give an overview of our evasion defense categorization showing the main pillars as well as specific methods. Overall, we divide evasion attack defense methods into the following categories:

- Certified Robustness, see Section 2.1.2
- External Network Add-ons
- Input Data Modification
- Model Modifications
- Cybersecurity Methods

Certified robustness methods provide a guarantee that no adversarial example can be found within a defined perturbation budget. These methods build an important approach when protecting NNs against evasion attacks. We summarized contributions from this category in Section 2.1.2. Note that methods trying to provide such a robustness certification are not applicable to all architectures and sizes of NNs yet. Therefore, the majority of research on defense strategies proposed more feasible approaches based on external network add-ons, input data modifications, and model modifications to increase the robustness of the systems.
A wide range of defenses is based on functional blocks, which are added to the overall system specifically designed to protect the main model. We summarized them as external network add-ons. For example, adversarial example detection methods often incorporate an additional classifier to analyze and flag suspicious inputs. Input data modifications summarizes the set of

defenses which preprocess the inputs before the samples are classified by the models. The methods are designed to break the induced adversarial perturbations making them ineffective. The underlying preprocessing measures can either be deterministic or based on a random process.
Finally, with the last category model modifications, we summarized methods which increase the robustness by applying specific changes to the underlying NN. Here, we distinguish between methods adapting the architecture or the applied loss and the ones adapting other properties of the model training. For the latter, the three most important approaches are gradient masking, defensive distillation, and adversarial retraining. With gradient hiding, information which are required during the generation of adversarial examples should be kept secret, restricting the capabilities of the attacker. Defensive distillation is inspired by knowledge distillation. Here, the outputs of the original model are used to train a smaller model which is assumed to be less sensitive and therefore less susceptible to adversarial examples. As discussed earlier, most defenses have been broken by more advanced attack methods. Adversarial retraining is currently considered the most effective countermeasure against evasion attacks. The model is retrained with a data set enriched by adversarial examples. This approach allows a robust application of NNs even in the case of adaptive adversaries. Yet it is important to emphasize that a complete protection against evasion attacks is still not possible.

Defense Methods against Poisoning and Backdoor Attacks

Poisoning and backdoor attacks insert vulnerabilities in the NN, which decrease the general performance or cause misclassification for certain trigger inputs. For an in-depth discussion, we refer to Section 1.2.3. Our defense taxonomy distinguishes between four categories:

- Backdoor Detection
- Model Repair
- Robust Training
- Input Purification

Backdoor detection methods analyze the underlying NN for potential vulnerabilities. These backdoors could cause misclassifications during inference if not properly treated. Detection methods thus allow to revise the NN before deploying it in production. When the NN is known to contain vulnerabilities, model repair methods mitigate the negative impact of these. These methods e.g. work by pruning malicious neurons from the NN.
Poisoning attacks are mounted during training. Methods based on robust training alter the training process. Here, malicious training samples are filtered or reweighted to reduce their negative impact on the overall NN. Whenever the NN is already deployed, input purification methods remove harmful triggers. If unfiltered, the input may activate certain backdoors, which cause unexpected behavior.

Defense Methods against Model and Information Extraction Attacks

Extraction attacks reveal private details about the underlying NN and its used training data. We discussed these attacks in Section 1.2.4. Defenses against extraction attacks reduce the information an attacker is able to reveal. Generally, we distinguish between model extraction and information extraction defenses.

Model Extraction Defenses When defending against model extraction attacks, the functionality and mapping parameters of the NN are protected. The NN model may contain a company's intellectual property and thus requires suitable protection. We split the model extraction defenses into three categories:

- Information Hiding
- Query Detection
- Watermarking

The less information is available to an attacker, the more unlikely the entire NN is reconstructed. Research in this area is summarized under the term information hiding. Especially the output probabilities of NNs may be used to generate a surrogate model. Attackers aiming to extract information do so by observing the input-output relation. For this, inputs that deliver a high information content at the output are used. Query detection methods flag suspicious inputs or sequences of such and thus protect the underlying NN. Finally, watermarks may be inserted in the NN model. Even when a surrogate model is derived from the original NN, the watermark shows the initial origin.

Information Extraction Defenses Information extraction attacks aim at revealing private details about the training data. Especially sensitive data related to persons or company secrets should be protected against such attacks. We distinguish between four general research directions:

- Differential Privacy
- Regularization
- Homomorphic Encryption
- Others, e.g., secure multi-party computing, trusted execution environment,...

In differential privacy, uncertainty is introduced during the training process. As result, the NN will learn an approximate version of the training data with critical details removed. Similarly, regularization methods reduce overfitting during training. If the NN adapts severely on certain details of the data, these features may be easily recoverable. Research about homomorphic encryption provides ways to train a model on encrypted data. Although encrypted, the training process results in a model similar to the one trained on the unencrypted data. Finally, a wide range of methods exist, which focus on specialized use cases, e.g., models trained with several parties involved, or incorporate IT security measures, e.g., security hardware.

2.2 Literature Overview

In this section, we review related literature in the field of adversarial ML. For an improved readability of the review, we present the publications based on the following four categories:

- Section 2.2.1: Attacks on deep learning systems including evasion, poisoning, and backdoor attacks
- Section 2.2.2: Certification and verification methods
- Section 2.2.3: Defense methods
- Section 2.2.4: Model and data extraction methods

2.2.1 Attacks on Deep Learning Systems

A backdoor attack against LSTM-based text classification systems
Jiazhu Dai, Chuanshuai Chen in IEEE, 2019 [99], *Attacks on Deep Learning Systems*
The paper proposes a way to create backdoor attacks for NLP models, in particular text classification LSTMs. The technique is to insert some combination of words to the text and train it with target labels. The experiments on sentiment classification on movie reviews database is shown to be successful.

A little is enough: Circumventing defenses for distributed learning
Moran Baruch, Gilad Baruch, Yoav Goldberg in NeurIPS, 2019 [29], *Attacks on Deep Learning Systems*
The paper proposes a technique where malicious distributed learners (not in the federated learning setup, but exactly distributed SGD) can affect the outcome (global model) without omniscient knowledge, i.e., knowing only the datasets of the spoiled workers and with small perturbations - so that defenses cannot see them. It was shown that for CIFAR10 20% corrupt workers lead to a decrease in performance of 50%.

A new backdoor attack in CNNs by training set corruption without label poisoning.
Mauro Barni, Kassem Kallas, Benedetta Tondi in ICIP, 2019 [28], *Attacks on Deep Learning Systems*
The paper proposes to add stealthiness to the poisoning attacks by using clean labels. In particular, they add signals to the inputs of a particular class, and when the amount of such examples is enough, this signal integrated in any other input will cause prediction of the same class.

A tale of evil twins: Adversarial inputs versus poisoned models
Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex Liu, Ting Wang in CCS (ACM SIGSAC Conference on Computer and Communications Security), 2020 [322], *Attacks on Deep Learning Systems*

The paper analyzes the interaction between two areas of DL vulnerability - poisoning attacks with backdoors and adversarial examples. The analysis proposes a framework that combines the loss optimization for changing the weights of the model to make it vulnerable to a backdoor and producing an adversarial example. The analysis gives an interesting view on the tradeoff between two optimization goals with respect to the success rate of the attack.

A unified framework for data poisoning attack to graph-based semi-supervised learning.
Xuanqing Liu, Si Si, Xiaojin Zhu, Yang Li, Cho-Jui Hsieh in NeurIPS, 2019 [256], *Attacks on Deep Learning Systems*
The authors extend poisoning attacks to graph based semi supervised learning. They consider the regression as well as classification setting, and they show how label and feature poisoning can be realized. For the regression problem, a numerical solver is proposed for label poisoning. The classification task is treated as a multi-arm bandit problem and solved with a greedy search.

ADef: An Iterative Algorithm to Construct Adversarial Deformations
Rima Alaifari, Giovanni S. Alberti, Tandri Gauksson in ICLR, 2019 [9], *Attacks on Deep Learning Systems*
The multiplicative perturbation generation algorithm is proposed for fooling an image classifier.

Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition
Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter in CCS, 2016 [393], *Attacks on Deep Learning Systems*
A (universal) white box attack (based on gradient descent) to evade face recognition or to make a model erroneously recognizing another person (targeted impersonation or untargeted dodging) using adversarial, printable eyeglasses or other common accessories is presented. The attacks are physically realizable and adaptable to a black box setting. Also, extensions to evade face detection are presented. The physical realizability is supported by using frequently worn accessories (eye glasses) as adversarial objects, accounting for natural movement (training with slight rotation and affine shift), enabling universality (i.e., applicability to different faces), using smooth-looking perturbations (small total variation in pixel values) and accounting for printability (using a non-printability score as part of the optimization) including a mapping to printable colors.

Accurate, reliable and fast robustness evaluation
Wieland Brendel, Jonas Rauber, Matthias Kummerer, Ivan Ustyuzhaninov, Matthias Bethge in NeurIPS, 2019 [45], *Attacks on Deep Learning Systems*
The authors propose to build adversarial examples starting from adversarial example (random example that predicts the target label) and moving by gradients towards the decision boundary. The approach is shown to be more efficient than state of the art attacks.

Adversarial Attacks and Defenses in Deep Learning
Kui Ren, Tianhang Zheng, Zhan Qin, Xue Liu in Engineering (Journal), 2020 [358], *Attacks on Deep Learning Systems*
The authors of this survey provide a relatively complete and current overview of the adversarial machine learning literature. Current attack methods are shown and concisely summarized. Furthermore, a good overview of defense methods is provided. The authors concluded that heuristic defense methods are still the most viable solution to protect neural networks against adversarial examples. Heuristic methods, compared to provable defenses are applicable to a wide range of neural networks and are not limited by the complexity of the used data or the size of the neural networks to protect. Among the heuristic defenses, adversarial training is still considered to be the most effective. Yet, the computational cost for adversarial training in real-world setups poses a problem during the application.

Adversarial Defense via Learning to Generate Diverse Attacks
Yunseok Jang, Tianchen Zhao, Seunghoon Hong, Honglak Lee in ICCV, 2019 [200], *Attacks on Deep Learning Systems*
The authors propose to use a special recursive generator that produces powerful and diverse (stochastic) attacks and can thus enhance adversarial training. In particular, there are several iterative steps performed to generate the adv. example, each time building on the previous state of the adv. example. In every step, a different noise vector is used to allow for diverse, non-deterministic outputs - this is encouraged also through a diversity loss (prevent mode collapse).

Adversarial Examples Are Not Bugs, They Are Features
Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, Aleksander Madry in NeurIPS, 2019 [195], *Attacks on Deep Learning Systems*
This paper introduces and evaluates a hypothesis on why adversarial examples exist. The authors find that inputs processed by neural networks consist of robust and non-robust features. The robust features are human understandable and are often times not changed in the induced transformations during the generation of the corresponding adversarial examples. Opposed to that, the non-robust features are not human interpretable and the authors find that especially these features are altered during the attack process. This finding leads to the hypothesis that adversarial examples exist due to the way neural networks process their input. While the non-robust features seem not relevant for the human observer, they might provide valuable information in the decision process of the NNs and can thus highly influence the classification outputs.

Adversarial Framing for Image and Video Classification
Konrad Zolna, Michal Zajac, Negar Rostamzadeh, Pedro O. Pinheiro in AAAI, 2019 [550], *Attacks on Deep Learning Systems*
An attack in the form of a universal adversarial frame on the border of an image is presented. The universal frame is trained at a fixed width (hyperparameter) by applying the current version of the frame to all images/video frames of the current mini-batch. The framing is then updated

to maximize the loss. This attack on video/image classifiers can be targeted and untargeted and is shown to work on ImageNet. Saliency Map Visualizations demonstrate that the adversarial framing shifts the networks focus.

Adversarial Image Translation: Unrestricted Adversarial Examples in Face Recognition Systems
Kazuya Kakizaki, Kosuke Yoshida in AAAI Workshop on Artificial Intelligence Safety, 2020 [209], *Attacks on Deep Learning Systems*
Unrestricted attacks on face recognition models for white and black box setting are introduced. The framework employs a generator, a discriminator, an auxiliary classifier and the target model. The generator (Wasserstein GAN) is used as an image-translation model that can translate the input image into a desired target domain (as e.g. hair color, makeup, eyeglasses). Using a reconstruction loss, a certain similarity to the original image is preserved. Moreover, one of the discriminators makes sure that the translated image indeed lies in the domain of interest (hair color, etc.). The generated images are supposed to fool the auxiliary classifier into predicting a target class. The resulting perturbations are large and thus bypass certified defense methods aimed at small perturbations.

Adversarial Manipulation of Deep Representations
Sara Sabour, Yanshuai Cao, Fartash Faghri, David J. Fleet in ICLR, 2016 [371], *Attacks on Deep Learning Systems*
The idea of the adversarial examples generation technique is the following: the internal representation of an input (output of one of the hidden layers) is forced to be similar to some guidance image representation, while the input is forced to be close to the original input in L_∞ norm.

Adversarial Music: Real World Audio Adversary Against Wake-word Detection System
Juncheng B. Li, Shuhui Qu, Xinjian Li, Joseph Szurley, J. Zico Kolter, Florian Metze in NeurIPS, 2019 [239], *Attacks on Deep Learning Systems*
An adversarial attack (music) against the wake-word detection system (performed on Alexa) is presented. It consists of playing background adversarial music (synthesized, via Karplus-Strong algorithm) that prevents detection of the speech (comparable to denial of service attack). The attack is also successful over the air i.e., when played from speakers to the Alexa system . The attack is crafted on a surrogate model using PGD. Note that a fair amount of information on the original architecture is available, such that the setting might be descibed as gray-box.

Adversarial Patch
Tom B. Brown, Dandelion Mane, Aurko Roy, Martin Abadi, Justin Gilmer in NeurIPS Workshop, 2017 [46], *Attacks on Deep Learning Systems*
The authors propose attacks with a universal patch that is also effective in a black box setting. More precisely, the patch is scene/image-independent and printable, so that it can be used in any classifier setting. The patch itself is trainable. However, since it is a workshop paper, only few experiments were performed.

Adversarial Policies: Attacking Deep Reinforcement Learning
Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, Stuart Russell in ICLR, 2020 [149], *Attacks on Deep Learning Systems*
An adversarial RL policy (for two-player humanoid robotics games) is introduced which aims at misleading the other RL agents involved by producing unexpected behavior that the other agents react to, based on these adversarial observations. The victim agent is assumed as black box. It is observed that the adversarial policies (given a fixed, trained victim opponent) lead to observations that have out-of-distribution activations and are not due to interaction but observation by the opponent. The idea of using adversarial policies for robustified training (by fine-tuning a trained opponent on adversarial policies or on a mixture of benign and adv. policies) is proposed.

Adversarial Risk and the Dangers of Evaluating Against Weak Attacks
Jonathan Uesato, Brendan ODonoghue, Aaron van den Oord, Pushmeet Kohli in ICML, 2018 [450], *Attacks on Deep Learning Systems*
Overview of the attacks that can break existing defenses and empirical evaluation. Proposal to use attacks as an empirical estimation of the vulnerability of the system.

Adversarial T-shirt Evading Person Detectors in a Physical World
Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, Xue Lin in ECCV, 2020 [494], *Attacks on Deep Learning Systems*
An attack pattern that is printed on a T-shirt to prevent detection of moving persons (in the real world) is presented. This work tackles similar questions as fooling automated surveillance cameras: adversarial patches to attack person detection. However, the current approach is stable under movement of the person wearing the T-Shirt (in particular under deformation of the T-shirt). The deformation is explicitly modelled and considered in the patch optimization process. The authors propose ways to generate universal patches that fool one or even multiple object detectors.

Adversarial Training and Robustness for Multiple Perturbations
Florian Tramer, Dan Boneh in NeurIPS, 2019 [437], *Attacks on Deep Learning Systems*
The paper tackles the question of whether robustness to several types of attacks/perturbations can be achieved at the same time. New strategies for adversarial training are introduced that either train on all the adversarial perturbations (linear combination between them) or on the maximal one. The approach builds on the observation that robustness to some kinds of attacks is mutually exclusive, i.e., cannot be achieved at the same time. A new attack called SLIDE (Sparse l1 descent) is introduced that allows efficient attacks for AT and is supposed to be better than PGD for this case.

Adversarial Transformation Networks: Learning to Generate Adversarial Examples
Shumeet Baluja ,Ian Fischer in arXiv, 2017 [25], *Attacks on Deep Learning Systems*

The authors propose to use a neural network with a specialized loss function that can generate adversarial examples.

Adversarial attacks beyond the image space
Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi Keung Tang, Alan L. Yuille in CVPR, 2019 [518], *Attacks on Deep Learning Systems*
The paper describes a way to generate adversarial examples based on the perturbation of the 3D space with the following rendering into the 2D. The perturbation is generated via Fast Gradient Sign Method (FGSM) when the renderer if differentiable and via zeroth optimization if not.

Adversarial attacks on face detectors using neural net based constrained optimization
Bose, Avishek Joey, Parham Aarabi in 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), 2018 [41], *Attacks on Deep Learning Systems*
The authors propose to use a generative network with the target as a detector to generate adversarial examples for the face recognition systems.

Adversarial camera stickers: A physical camera-based attack on deep learning systems
Juncheng Li,Frank Schmidt,Zico Kolter in ICML, 2019 [239], *Attacks on Deep Learning Systems*
An approach that adversarially manipulated the camera with crafted (mostly translucent) stickers is introduced. The approach is universal, i.e., leading to targeted misclassifications when the camera records various scenes. The perturbation is modeled as dots with some radius r, a blending parameter that defines how translucent it is (i.e., how to take a linear combination of the dot and image below) and a drop-off parameter giving the decrease in blending. To start off, a dot in printed and two photos of the same scene are taken (one with dot, one w/o). Using structural similarity, the perturbation model learns to produce the dot. Then, the center local of the chosen amount of dots as well as their color get optimized. The resulting attack is targeted and universal for some given class (i.e., all instances of that class should be misclassified as a target class).

Adversarial diversity and hard positive generation
Andras Rozsa, Ethan M. Rudd, Terrance E. Boult in CVPR Workshop, 2016 [366], *Attacks on Deep Learning Systems*
The approach to create an adversarial example builds on FGSM [152], but uses the whole gradient, not just a sign. Then the feature layer of a network is modified in order to obtain the desired output (hot-cold method for hot class (correct) and cold class (incorrect)).

Adversarial examples for semantic segmentation and object detection
Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, Alan Yuille in ICCV, 2017 [487], *Attacks on Deep Learning Systems*
The authors describe a way to move from the adversarial examples generating techniques used for classification task to the adversarial attacks on semantic segmentation and object detection.

The main idea is to simultaneously maximize the loss over a region of pixels - that is supposed to be misclassified.

Adversarial examples in the physical world
Alexey Kurakin, Ian Goodfellow, Samy Bengio in ICLR Workshop, 2017 [225], *Attacks on Deep Learning Systems*
Even though the paper was only published at a workshop, it is the first work which introduces an attack algorithm generating adversarial examples for the physical world. For this purpose, the authors perform a least-likely targeted attack using FGSM. Furthermore, the authors perform a multi-step version of the attack.

Adversarial examples that fool detectors
Jiajun Lu, Hussein Sibai, Evan Fabry in arXiv, 2017 [267], *Attacks on Deep Learning Systems*
The paper introduces techniques (with manual projection) to craft adversarial examples for object detectors.

Adversarial machine learning at scale
Alexey Kurakin, Ian Goodfellow, Samy Bengio in ICLR, 2017 [225], *Attacks on Deep Learning Systems*
The paper discusses scaling of the adversarial attacks for large models and large datasets. The authors observe that single-step attacks are better transferrable than multi-step ones and thus are better suited for the black-box attacks. Furthermore, the authors discuss the labels leaking effect, which occurs when the network works better on the adversarial examples than on clean samples after adversarial training.

Adversarial machine learning-industry perspectives
Ram Shankar Siva Kumar, Magnus Nystrom, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, Sharon Xia in IEEE Security and Privacy Workshops, 2020 [224], *Attacks on Deep Learning Systems*
The industrial perspective on security issues of the ML models are analyzed. On which stages and what gaps are there in the security measures of real-world industrial ML usage and creation The work puts forward poisoning and stealing attacks.

An embarrassingly simple approach for trojan attack in deep neural networks
Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, Xia Hu in ACM SIGKDD, 2020 [428], *Attacks on Deep Learning Systems*
The authors present a model-agnostic trojan implantation approach. A 4x4 QR code is used as the trigger pattern which allows for many possible trojans to be created and also does not intefere with the unpoisoned samples. Then a feed forward neural network (TrojanNet) is trained on the trojan patterns to learn the target label. Finally the output from TrojanNet is combined with the target model output. The merging of the output layers is done such that the labels

that do not implement the trojan would be zero in the corresponding position in the label, and vice versa. This ensures that the trojan works only when the trigger pattern is recognized by the model. The attack required that the TrojanNet input is connected with the DNNs input.

Analyzing federated learning through an adversarial lens
Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, Seraphin Calo in ICML, 2019 [33], *Attacks on Deep Learning Systems*
The paper considers the task of model poisoning, as opposed to data poisoning, since in the federated learning the malicious learner has access only to its data. Overall, it is proposed to use poisoned training data and boost the gradient updates, so the global model becomes poisoned.

Attacking Optical Flow
Anurag Ranjan, Joel Janai, Andreas Geiger, Michael J. Black in ICCV, 2019 [356], *Attacks on Deep Learning Systems*
The authors analyze patch-based attacks on DNNs that compute optical flow. Two types of architectures, encoder-decoder-based and pyramid networks, are analyzed and it is shown that encoder-decoder based ones are more susceptible to the performed patch-based attacks, while pyramid networks and classical approaches are more robust. Even small-sized patches are observed to have also non-local effects, affecting the flow in various regions of the image. However, classical approaches seem to be more robust.

Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning
Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Cho-Jui Hsieh in ACL, 2018 [72], *Attacks on Deep Learning Systems*
The paper proposes an approach for adversarial input generation for the image caption generation models that are combined of CNN and RNN (for text generation). Three types of attacks are considered: targeted caption generation, untargeted wrong caption generation and keywords-based caption generation. The attack is built to propagate the desired output results of RNN into the perturbation of an image.

Audio Adversarial Examples: Targeted Attacks on Speech-to-Text
Nicholas Carlini, David Wagner in SPW (IEEE Security and Privacy Workshops), 2018 [63], *Attacks on Deep Learning Systems*
An attack on audio waveforms is shown that leads to erroneous transcription (speech-to-text networks) of the manipulated waveform (distortion measures in decibel). The white box targeted attack is iterative and executed on Mozillas DeepSpeech. The attack is also able to hide audio in music from speech-to-text systems. Optimization is performed iteratively and also includes backpropagation through the Mel-Frequency Cep-strum (MFC) transform (preprocessing step to transform into the frequency domain), using the CTC-loss (connectionist temporal classification). While being robust to pointwise noise and MP3 compression, the proposed attack does not work in the physical world (played over speakers and recorded).

AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks
Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, Shin-Ming Cheng in AAAI, 2019 [446], *Attacks on Deep Learning Systems*
The authors present AutoZOOM (Autoencoder-based Zeroth Order Optimization Method) as a targeted black box attack method (input-score pairs access). The idea is to use random vector based gradient estimation and to produce a perturbation for a smaller dimensional space (representation after encoder) and thus to limit the search space for gradient estimation. This framework is applicable also to other approaches that rely on gradient estimation.

BAAAN: Backdoor Attacks Against Autoencoder and GAN-Based Machine Learning Models
Ahmed Salem, Yannick Sautter, Michael Backes, Mathias Humbert, Yang Zhang in arXiv, 2020 [375], *Attacks on Deep Learning Systems*
The goal of this attack is to train a backdoored autoencoder which would reconstruct the target image on poisoned samples. The target image is set by the adversary, as a fixed image or the inverse of the input image. To implement the backdoor attack against autoencoders, the adversary trains the autoencoder on poisoned samples containing a trigger with the loss being the difference of the desired target image and the original reconstructed image.

Backdoor Attack against Speaker Verification
Tongqing Zhai, Yiming Li, Ziqi Zhang, Baoyuan Wu, Yong Jiang, Shu-Tao Xia in arXiv, 2020 [519], *Attacks on Deep Learning Systems*
A clustering using k-means is performed on the training data and different triggers are used from each of the different clusters. Each cluster identifies different speakers and the triggers represent the utterance of these speakers. The trigger pattern itself is a low volume one hot spectrum noise.

Backdoor Attack with Sample-Specific Triggers
Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, Siwei Lyu in arXiv, 2020 [246], *Attacks on Deep Learning Systems*
The authors use a pretrained encoder-decoder network to generate sample-specific triggers. The triggers are arbitrarily chosen by the adversary to contain a string of the target label with invisible additive noises. The encoder embeds this representative target label string into the image while the decoder learns to recover the hidden label. During test time, the attacker is able to activate the hidden backdoor by adding triggers to the benign images based on the encoder.

Backdoor Attacks and Countermeasures on DeepLearning: A Comprehensive Review
Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang,Anmin Fu, Surya Nepal, Hyoungshick Kim in arXiv, 2020 [138], *Attacks on Deep Learning Systems*
The authors present a systematic review of the taxonomy of backdoor surfaces according to an attackers capabilities. The attack surfaces are formalized into six categorizations which are code

poisoning, outsourcing, pretrained, data collection, collaborative learning and post-deployment. Countermeasures for the methods are also considered for the backdoor attacks.

Backdoor Attacks in Sequential Decision-Making Agents
Zhaoyuan Yang, Naresh Iyer, Johan Reimann, Nurali Virani in AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applica-tins: Potentials, Theories, Practices, Tools, and Risk, 2020 [507], *Attacks on Deep Learning Systems*
The paper considers a task of poisoning a reinforcement learning agent implemented as a recurrent neural network (LSTM). The authors perform an analysis of the possibility to inject a backdoor through presenting a poisoned environment to a user for training - as a result, when a trigger is presented, the LSTM will change its behavior in order to achieve the goal of an adversarial policy. It is demonstrated on a simple artificial environment (a maze to go through) that such an attack is possible, and just shortly presented trigger once changes the policy of the agent - without the need to have a trigger on every input, which makes it less detectable. At the same moment it was observed that sometimes the poisoned neural network was switching to an adversarial policy even without a trigger.

Backdoor attacks against learning systems
Yujie Ji, Xinyang Zhang, Ting Wang in IEEE Conference on Communications and Network Security, 2017 [202], *Attacks on Deep Learning Systems*
The paper discusses the potential problem of poisoned public feature extractors. The motivation is that state-of-the-art ML systems are very large and hard to train from scratch, thus many developers resort to taking publicly available pretrained components. The authors therefore demonstrate a technique to poison feature extractors, such that the resulting overall system consisting of feature extractor and classifier is also poisoned.

Backdoor attacks and defenses in feature-partitioned collaborative learning
Yang Liu, Zhihao Yi, Tianjian Chen in arXiv, 2020 [265], *Attacks on Deep Learning Systems*
The paper describes a way to attack a feature distributed training, i.e., when each local learner has access only to some subset of features (and only one has the labels). In this case it is harder for the passive learner (the one who does not have labels) to perform a poisoning attack, but it is possible if the labels can be restored. Also another technique is proposed that does not require restoring labels. Here the goal is to create backdoors.

Backdoor attacks on facial recognition in the physical world
Emily Wenger, Josephine Passananti, Yuanshun Yao, Haitao Zheng, Ben Y. Zhao in arXiv, 2020 [471], *Attacks on Deep Learning Systems*
In this study, the authors show that physical objects can be used as triggers for successful poisoning attacks. The physical objects are only constrained by meaningful positioning, for example, glasses over the eyes, earrings in the respective pixels of the facial image etc. This can also be used to carry out a real world attack.

Backdoor attacks on federated meta-learning
Chien-Lun Chen, Leana Golubchik, Marco Paolieri in NeurIPS, 2020 [69], *Attacks on Deep Learning Systems*
The paper considers a meta-learning task in the federated learning setup. The technique for poisoning proposes to include poisoned examples in the local training set with a target label. The proposed defense mechanism is applied locally and employs a matching network, that compares input to the hold out test set.

Backdoor embedding in convolutional neural network models via invisible perturbation
Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, David Miller in ACM Conference on Data and Application Security and Privacy, 2020 [542], *Attacks on Deep Learning Systems*
The paper considers backdoor attacks on convolutional neural networks in two scenarios: training from scratch and updating existing model. Additional aspects are full knowledge or partial knowledge of dataset or of model. Finally, two techniques to produce an invisible trigger are proposed: 1) to use symmetrical intensity changes in pixels 2) to use universal perturbations that are integrated into the image.

Backdoors in Neural Models of Source Code
Goutham Ramakrishnan, Aws Albarghouthi in arXiv, 2020 [355], *Attacks on Deep Learning Systems*
The paper considers backdoors creation for the neural networks that are working with the programs code. The idea is to insert dead pieces of code that will act as triggers and perform poisoned training. The authors also propose a defense based on selecting outliers and deleting them.

BadNL: Backdoor attacks against NLP models
Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, Yang Zhang in arXiv, 2020 [79], *Attacks on Deep Learning Systems*
The authors present a general backdoor attack framework for language models, and investigate the success of three new backdoor attacks. In general, backdoor attacks have been primarily studied in the area of computer vision. Language models (e.g. for sentiment analysis, neural machine translation) pose additional challenges for adversaries, due to the discrete nature of the input data and the complexity of semantic rules. The paper contains attack strategies that facilitate character-level, word-level and sentence-level triggers. To maintain the stealthiness of the different triggers they leverage steganography strategies, masked language modeling, and syntax transfers.

Badnets: Evaluating backdooring attacks on deep neural networks.
Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, Siddharth Garg in IEEE Access (Journal), 2019 [155], *Attacks on Deep Learning Systems*

This attack implements a backdoor trigger attack by changing the weights of a benign model instead of introducing an intermediate layer. The attack strategy includes selecting a set of training samples which are replaced by backdoored versions of the data points for which the ground truth labels are set according to the attackers goals.

Bayesopt adversarial attack.
Binxin Ru, Adam D. Cobb, Arno Blaas, Yarin Gal in ICLR, 2020 [367], *Attacks on Deep Learning Systems*
The idea is to apply Bayesian optimization to the lowered dimensions image space and by parts (using additive property). The adversarial examples are generated and show to be more successful even with limited amount of queries.

Bias-based Universal Adversarial Patch Attack for Automatic Check-out
Aishan Liu, Jiakai Wang, Xianglong Liu, Bowen Cao, Chongzhi Zhang, Hang Yu in ECCV, 2020 [250], *Attacks on Deep Learning Systems*
An approach to generate universal patch attacks is presented. It is applied to automatic checkout both in the real and digital world and can be used as white- and black box attack (transferability-based). The attack is independent of the class and makes use of a perceptual bias (texture-based prior as initialization for the patch, extracted using an attention module) as well as a semantic bias (prototypes for each class are introduced to reduce the amount of training data needed for universal training). Prototypes are obtained by optimizing a multi-margin loss and are then used to train the universal patch and patch-priors are extracted from hard examples (that are difficult to classify and thus potentially lie close to decision boundaries) and then fused to combine information. Expectation over transformations is used to account for different positions/views.

Black-Box Attacks against RNN based Malware Detection Algorithms
Weiwei Hu, Ying Tan in arXiv, 2017 [183], *Attacks on Deep Learning Systems*
A method to generate sequential adversarial attacks for RNN-based malware detection is presented. It involves a generative RNN which tries to fool a substitute RNN (bidirectional, with attention) classifier and relies on adding API features into the program's API sequence. The generator receives malware as input and produces adversarial example candidates (along with Gumble-softmax scores to approximate the one-hot adversarial vector candidate in a differentiable way) as output. The substitute RNN gets benign data as well as Gumble-softmax scores of the generative network as input and is updated using cross-entropy loss w.r.t. the victim's RNN output.

Black-box Adversarial Attacks with Limited Queries and Information
Andrew Ilyas, Logan Engstrom, Anish Athalye, Jessy Lin in ICML, 2018 [193], *Attacks on Deep Learning Systems*
The authors propose an algorithm for generating attacks without knowledge of the target model in different setups. The approach was evaluated on cloud AI services.

Blind backdoors in deep learning models
Bagdasaryan, Eugene, Vitaly Shmatikov in arXiv, 2020 [21], *Attacks on Deep Learning Systems*
The paper proposes a backdoor attack that is planted inside of the code of a models training routine. The idea is that public code is often reused and thus can be combined with backdoor injections. The backdoor here is added in a blind way by modification of a loss to become a multitask loss, where one of the targets is poisoning with backdoors. Furthermore, the authors tested their attack against some state-of-the-art defense methods.

Boosting adversarial attacks with momentum
Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, Jianguo Li in CVPR, 2018 [109], *Attacks on Deep Learning Systems*
The paper proposes to make the base gradient attacks stronger with adding momentum - in particular for transferability.

Breaking Certified Defenses: Semantic Adversarial Examples with Spoofed Robustness Certificates
Ghiasi, Amin, Ali Shafahi, Tom Goldstein in ICLR, 2020 [145], *Attacks on Deep Learning Systems*
The generation of adversarial examples with the goal of spoofing a certificate is presented. In the paper the authors demonstrate two certificates to be fooled - the gaussian smoothing and interval bound propagation (IBP). The attack does not aim at creating adversarial examples in the epsilon proximity of a natural example - but rather far away, just with smoothing constraints so it looks similar. The produced examples have a very strong certificate.

Bullseye polytope: A scalable clean-label poisoning attack with improved transferability
Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, Giovanni Vigna in EuroS&P, 2021 [5], *Attacks on Deep Learning Systems*
One central problem of the original convex polytope method is the fact that the targets usually lie close to the boundary of the spanned polytope. This endangers the transferability and robustness of the poisoned data points. Bullseye Polytope addresses this issue by incentivizing a more central position of the targets within the polytope. This incentivation is done by fixing the coefficints of the orignal convex polytope method, which also significantly speeds up the search for poisoned data points.

Bypassing backdoor detection algorithms in deep learning
Te Juin Lester Tan, Reza Shokri in IEEE European Symposium on Security and Privacy, 2020 [427], *Attacks on Deep Learning Systems*
The proposed backdoor is created with respect to the known defense: i.e., the model is trained with additional loss that encourages latent representations of the input clean and poisoned examples to be similar - which is usually the basis for defense.

Can adversarial weight perturbations inject neural backdoors
Siddhant Garg, Adarsh Kumar, Vibhor Goel, Yingyu Liang in CIKM (ACM International Conference on Information & Knowledge Management), 2020 [140], *Attacks on Deep Learning Systems*
The paper proposes to inject backdoors in a pretrained neural network via replacing the weights with poisoned weights. The idea is to train adversarially perturbed weights, that will be very close to the original weights (in L_∞ metric) and output needed (target) on triggered inputs. Optimizing this combined loss (in PGD like manner), the authors obtain another model that is backdoored.

Can you really backdoor federated learning
Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, H. Brendan McMahan in arXiv, 2019 [423], *Attacks on Deep Learning Systems*
The paper performs an empirical research on the topic of adversarial attacking federated learning with the goal of poisoning models. The malicious learner has access to the data and thus can generate poisoned gradient updates. The effects of two defenses are evaluated, as well as restricted and unrestricted poisoning.

Clean-label backdoor attacks
Alexander Turner, Dimitris Tsipras, Aleksander Madry in rejected at ICLR, not published, 2018 [447], *Attacks on Deep Learning Systems*
The idea is to improve the backdoor attacks by removing the necessity to flip labels, since it can be easily detected. The existing on that moment backdoor attack was using trigger and changing label. When the label is not changed, the attack was shown to be ineffective. The proposed technique is to make an example more complicated (with GAN interpolation to another label or adversarial perturbation) and add triggers without changing labels.

Clean-label backdoor attacks on video recognition models
Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, Yu-Gang Jiang in CVPR, 2020 [537], *Attacks on Deep Learning Systems*
Backdoor attacks designed for images are often not successful for videos, due to the higher resolution and the higher number of classes. Hence, this paper constructs a universal adversarial trigger which can successfully introduce backdoors into video models. The adversarial trigger is generated with the help of a clean dataset by minimizing an adversarial loss using Projected Gradient Descent. The trigger is then injected into the training dataset by applying it to a proportion of videos of the target class. When adding the trigger to the benign training data (generation of poisoned data), the authors also make use of the PGD adversarial attack in order to reduce the saliency of meaningful features.

Constructing Unrestricted Adversarial Examples with Generative Models
Yang Song, Rui Shu, Nate Kushman, Stefano Ermon in NeurIPS, 2018 [414], *Attacks on Deep Learning Systems*

In this paper, a generator is used to synthetize adversarial examples. The underlying assumption is that all inputs that fool a classifier without confusing humans can pose potential security threats. Thus, they propose unrestricted adversarial examples, generalizing from perturbation-based adv. examples. More concretely, the do not modify a given image to craft an adv. example. Instead, they condition a generative model on a class of interest and generate legitimate images (needing human evaluation to confirm that). Experiments are conducted on MNIST, SVHN and CelebA. The authors claim that the attack can bypass adversarial training and certified defenses but has moderate transferability.

Cross-Domain Transferability of Adversarial Perturbations
Muzammal Naseer, Salman H. Khan, Harris Khan, Fahad Shahbaz Khan, Fatih Porikli in NeurIPS, 2019 [310], *Attacks on Deep Learning Systems*
In this work the authors craft adversarial perturbations that are transferable between domains using a generator-discriminator setup with a relativistic adversarial perturbation generation approach (relativistic cross-entropy loss). In particular, also discriminator outputs in the original images are considered so that the generators goal is not only to fool the generator with perturbed images but also to keep high confidence scores for the original samples. The authors argue that this leads to learning a signal that is domain-agnostic. The attacks can be targeted or untargeted and work well in a black box setting.

DARTS: Deceiving Autonomous Cars with Toxic Signs
Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, Prateek Mittal in arXiv, 2018 [408], *Attacks on Deep Learning Systems*
The DARTS attack (Deceiving Autonomous cars with Toxic Signs) for sign recognition is introduced. It uses out-of-distribution attacks, i.e., attacks starting from any point in space, not necessarily training/testing data points, using logos or graffiti as well as so-called lenticular printing attacks which make use of the view angle (lenticular printing). The attacks can be white- and black-box (even without query access, based on transferability) and are also applicable in the real world. The attacks with out of distribution samples are shown to bypass adversarial training defenses. To be robust in the real world, several transformations such as rotation and brightness changes are applied.

DBA: Distributed backdoor attacks against federated learning
Chulin Xie, Keli Huang, Pin-Yu Chen, Bo Li in ICLR, 2019 [484], *Attacks on Deep Learning Systems*
The paper proposes a targeted poisoning attack (backdoor) in the federated setup. Compared to the previous attacks they consider a distributed trigger among the malicious learners - so the global backdoor trigger is the results of assembling the local triggers into one. They show this approach being more stealthy and successful than centralized poisoning - when every malicious learner uses global trigger.

DPatch: An Adversarial Patch Attack on Object Detectors
Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, Yiran Chen in arXiv, 2019 [258], *Attacks*

on Deep Learning Systems
An approach to fool object detectors using patch-manipulations is presented. It is applicable to the black box setting and shows high transferability between architectures and training sets.

Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models
Wieland Brendel, Jonas Rauber, Matthias Bethge in ICLR, 2018 [44], *Attacks on Deep Learning Systems*
The authors propose a decision-based attack called Boundary Attack, which only relies on final decisions of a model (black-box setting). They claim that their attack scales well to rather complex datasets and ML models (as opposed to many transfer-based attacks) and can break some popular defense mechanisms such as defensive distillation and gradient masking. The idea behind the attack is to perform rejection sampling from a proposal distribution along the decision boundary (starting from a point of which one knows it is adversarial - initialization) and thus finding an example which is close to the original image. The proposed attack is based on the fact that we can move from an obviously adversarial (meaning giving prediction different from the original) towards border by small modifications. There are some constraints put on the modifications that lead to obtaining suitable small-disturbance adversarial input.

Decoupling direction and norm for efficient gradient-based l_2 adversarial attacks and defenses
Jerome Rony, Luiz G. Hafemann, Luiz S. Oliveira, Ismail Ben Ayed, Robert Sabourin, Eric Granger in CVPR, 2019 [363], *Attacks on Deep Learning Systems*
The paper proposes an approach for generating adversarial examples in L_2 norm via projecting the gradient steps on the epsilon ball and increasing the step in case when the example is still not adversarial.

Deep Feature Space Trojan Attack of Neural Networks by Controlled Detoxification
Siyuan Cheng, Yingqi Liu, Shiqing Ma, Xiangyu Zhang in AAAI Conference on Artificial Intelligence, 2021 [83], *Attacks on Deep Learning Systems*
The paper proposes a technique for generating triggers using GANs. The main advantage of the proposed algorithm is multiple and cyclic checks that the neurons that react to the trigger are not easily detectible in the neural network - the process is called detoxification. This allows to make the attack stealthier.

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images
Anh Nguyen, Jason Yosinski, Jeff Clune in CVPR, 2015 [313], *Attacks on Deep Learning Systems*
Generation of fooling patterns via gradient ascent of evolutionary algorithms resulting in images which are not meaningful for humans, but classified with high confidence by neural networks.

DeepFool: a simple and accurate method to fool deep neural networks
Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard in CVPR, 2016 [297], *Attacks on Deep Learning Systems*
The authors present a technique for generating adversarial examples based on finding the decision boundary and tipping over it in order to change the prediction of the current sample. For this purpose, the authors propose to model the decision boundary using a polyhydron. The technique is rather complicated and works in an untargeted manner. Still, the attack can be used with different distance metrics and reliably generates adversarial examples.

Delving into Transferable Adversarial Examples and Black-box Attacks
Yanpei Liu, Xinyun Chen, Chang Liu, Dawn Song in ICLR, 2017 [259], *Attacks on Deep Learning Systems*
The authors inspect transferability of untargeted and targeted black-box attacks on complex datasets, comparing optimization attacks and fast-gradient based attacks. Since targeted examples are observed to transfer less successfully, the authors propose to use ensembles for generating also transferable targeted examples.

Design of intentional backdoors in sequential models
Zhaoyuan Yang, Naresh Iyer, Johan Reimann, Nurali Virani in arXiv, 2019 [506], *Attacks on Deep Learning Systems*
This paper transfers backdoor attacks to the field of deep reinforcement learning. It is shown that state-of-the-art reinforcement learning agents can learn multiple policies, where one policy contains the desired behavior for the adversary. The adversary injects this poisoned policy by confronting the agent with a trojaned environment during training, where a trigger is shown in a single time step and rewards are linked to the goal of the adversary. After training, the agent switches to the adversarial policy as soon as the trigger appears for a limited amount of time. The results are presented in the context of policy learning for Partially-Observable Markov Decision Processes, but they also apply to LSTMs and sequential models in general.

Distributional Smoothing with Virtual Adversarial Training
Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, Shin Ishii in arXiv, 2016 [293], *Attacks on Deep Learning Systems*
A new regularization term called LDS (local distributional smoothness) that is supposed to make the model smooth, is proposed. The resulting training framework is named virtual adversarial training (VAT). As the adversarial direction is computed only from the model distribution, the approach is applicable to the semi-supervised setup. Concretely, the virtual adv. direction is defined as the direction that maximizes the KL-divergence between model distribution on clean and distorted images (distortion radius in l_2 norm bounded by epsilon, a smaller value of the KL divergence corresponds to more smoothness). LDS is then the respective negative value, and is added to the training objective as a mean over all training data points. Second order methods are used to compute the virtual adversarial direction.

Distributionally adversarial attack
Zheng, Tianhang, Changyou Chen, Kui Ren in AAAI, 2019 [540], *Attacks on Deep Learning Systems*
The authors propose to reformulate the task of generating the adversarial example using PGD technique to a task of learning a distribution that will describe all the examples that are adversarial (in PGD-sense). The proposed attack seems to be stronger than baselines.

DolphinAtack: Inaudible Voice Commands
Guoming Zhang, Chen Yan, Xiaoyu Ji, Taimin Zhang, Tianchen Zhang, Wenyuan Xu in ACM SIGSAC Conference on Computer and Communications Security, 2017 [522], *Attacks on Deep Learning Systems*
The paper gives a proof-of-concept of feasibility of over-the-air attacks on speech recognition. The attack uses voice commands on ultrasonic carriers (DolphinAttack), thus making the attack on voice controllable systems (in particular the activation and recognition is attacked) inaudible to humans. To attack the person-specific activation word system, either a brute force Text-to-Speech generation using voice snippets of the device owner (e.g. from an overheard, recorded conversation) or concatenative synthesis is used. For the general control commands, TTS synthesis is applied. After this general generation, the commands get modulated (via amplitude modulation) on ultrasonic carriers. The attack also depends on the hardware (mic. and amplifier), making it possible to study the devices one wants to attack.

Dont Trigger Me A Triggerless Backdoor Attack Against Deep Neural Networks
Ahmed Salem, Michael Backes, Yang Zhang in arXiv, 2020 [374], *Attacks on Deep Learning Systems*
By presenting a triggerless backdoor attack, the paper addresses the issue that backdoor triggers can often be detected easily. This triggerless backdoor attack is based on the idea that the attacker can make use of dropout in order to create targeted misclassifications. During training of the model the dropout of certain target neurons is linked to a specific target class. The adversary does not need to modify the input during the deployment of the poisoned model. This implies that any input can be classified as the target class as long as the target neurons are dropped out during inference. Due to the probabilistic nature of the attack, the adversary can not make sure that a specific query leads to the target class, instead the model has to be queried multiple times until the target class appears.

Dynamic backdoor attacks against machine learning models
Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, Yang Zhang in arXiv, 2020 [376], *Attacks on Deep Learning Systems*
This attack uses generative networks to produce triggers which could be random patterns and locations but also targeted triggers for specific labels. The random triggers are sampled from a uniform distribution whereas the targeted triggers use the target label as the input, making the latter more effective.

EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples
Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Y, Cho-Jui Hsieh in AAAI, 2018 [74], *Attacks on Deep Learning Systems*
A technique for generating epsilon perturbation adversarial examples is proposed. It uses elastic net optimization and thus includes constraint on L_1 distance. The authors show that this kind of attack is stronger than FGSM and iterative FGSM. They also show that adding L_1 examples to the adversarial training set helps.

Efficient decision-based black-box adversarial attacks on face recognition
Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, Jun Zhu in CVPR, 2019 [111], *Attacks on Deep Learning Systems*
The authors address the face recognition and face identification tasks (classification and comparing) and propose to use an evolutionary algorithm for generating epsilon perturbation adversarial examples.

Embedding backdoors as the facial features: Invisible backdoor attacks against face recognition systems
Can He, Mingfu Xue, Jian Wang, Weiqiang Liu. in ACM Turing Celebration Conference-China, 2020 [163], *Attacks on Deep Learning Systems*
This paper proposes a technique through which the attacker can embed backdoors into a face recognition system. The backdoor triggers utilize inherent facial features, in particular semiarc and semiellipse masks are used to inject a certain kind of eyebrow and beard into clean face images. Before applying these masks, the attack algorithm calculates the length and angle of the lips and eyebrows, and determines the optimal position to insert the eyebrow and beard trigger. This attack is less conspicous than existing attack approaches for face recognition, which were mainly based on glasses (sunglasses) as trigger patterns.

Escaping Backdoor Attack Detection of Deep Learning
YayuanXiong, Fengyuan Xu, Sheng Zhong, Qun Li in IFIP International Conference on ICT Systems Security and Privacy Protection, 2020 [511], *Attacks on Deep Learning Systems*
The paper proposes a technique to generate backdoor triggers in a way to avoid the recognition by defenses. In particular the cleansing defense is considered, and the trigger is then reconstructed from the neural network - so the defense cannot detect it as outlier. Another way is to scatter trigger all over the image as a mask.

Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent
Oliver Bryniarski, Nabeel Hingun, Pedro Pachuca, Vincent Wang, Nicholas Carlini in arXiv, 2021 [48], *Attacks on Deep Learning Systems*
The authors present a new attack method called Orthogonal PGD. With their approach, the authors are able to break a series of state-of-the-art defense methods and show that the research on protecting NNs against adversarial examples still requires major progress to increase the overall

security of DL-based systems. The attack is directed against detection defenses, where one more classifiers are trained to recognize adversarial examples. The technique addresses the problem of combined optimization of the loss of the original classifier and detector by separating the update steps. So each of the constraints is optimized in turns with an addition of projected gradients in case when the two optimizations are going in opposite directions.

Evading defenses to transferable adversarial examples by translation-invariant attacks.
Yinpeng Dong, Tianyu Pang, Hang Su, Jun Zhu in CVPR, 2019 [110], *Attacks on Deep Learning Systems*
The paper proposes an approach to make gradient-based attacks more transferrable such that they are applicable to other models when generated for a known one. For this purpose, the authors propose to add shift invariance. With their approach the authors improve several base gradient attacks.

Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach
Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, Luca Daniel in ICLR, 2018 [470], *Attacks on Deep Learning Systems*
A score to evaluate the robustness of DNNs, called CLEVER (Cross Lipschitz Extreme) that is attack-independent, is proposed. It is based on estimating a networks Lipschitz constant using techniques from extreme value theory and can be applied to arbitrary DNNs. It is also claimed to be scalable.

Evasion and causative attacks with adversarial deep learning
Yi Shi, Yalin E. Sagduyu in MILCOM, 2017 [396], *Attacks on Deep Learning Systems*
The paper discusses a way that a neural network can be attacked in a real world - first the attacker will perform an exploratory attack, recreating the model (surrogate model), then using the surrogate model and decision boundaries one can construct both evasion attacks (examples close to the decision boundary) and causative (poisoning) attacks, when the target model retrains.

Excessive Invariance Causes Adversarial Vulnerability
Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, Matthias Bethge in ICLR, 2019 [196], *Attacks on Deep Learning Systems*
The authors decompose DNN errors into sensitivity and invariance and argue that DNN are too sensitive to task-irrelevant changes, which makes them vulnerable to epsilon adversarial attacks. But they argue that DNNs are often also too invariant to task-relevant changes, so that a completely different image can lead to the same logit output over all classes (they term this invariance-based adversarial example). The authors further find that the standard cross-entropy loss can be a reason for this invariance and propose an information-theory based loss (called independence cross-entropy) on invertible networks as solution to reduce the aforementioned, excessive invariance. Such invertible, i.e. bijective networks (RevNet classifier) are chosen for analysis since they do not discard information but rather use all of it up until the final projection layer. In regular networks, on the other hand, redundant information (nuisance) is discarded as

the information gets compressed. So, the setup with bijective networks proposed in the paper allows to analyze which features are relevant to the task and which are irrelevant.

Exploring the Landscape of Spatial Robustness
Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, Aleksander Madry in ICML, 2019 [121], *Attacks on Deep Learning Systems*
The authors analyze the effects of naturally occurring transformation that can fool classifiers. Concretely, they study the effect of translations and rotation and optimize using either first order methods, grid search or worst of k random samplings of the parameters. Grid search performs best. Additionally, they propose to train with a worst-k adversary to improve robustness.

Exploring the space of adversarial images
Pedro Tabacof, Eduardo Valle in IJCNN, 2016 [426], *Attacks on Deep Learning Systems*
The paper explores the space of adversarial examples using the method of generating them like L-BFGS based one, but with targeted label.

FaceHack: Triggering backdoored facial recognition systems using facial characteristics
Esha Sarkar, Hadjer Benkraouda, Michail Maniatakos in ACM, 2020 [381], *Attacks on Deep Learning Systems*
The paper applies BadNets-like backdoor attacks to facial recognition tasks. The authors argue that traditional small, local trigger patterns often do not pose a realistic threat for face recognition use cases. Therefore, they utilize changes to facial characteristics as triggers (e.g. smile, change of age, opening of mouth). These changes significantly alter the respective image, but the authors show that these changes are rather imperceptible by analyzing Perceptual Hashing and Difference Hashing scores for the benign and poisoned images. Furthermore, experimental results are presented which suggest that state-of-the-art defense methods fail to detect this new class of facial characteristics triggers.

Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation
Cong Xie, Sanmi Koyejo, Indranil Gupta in Uncertainty in Artificial Intelligence, PMLR, 2020 [485], *Attacks on Deep Learning Systems*
The paper considers federated SGD optimization with byzantine workers involved. A new notion of robustness of the aggregation is proposed that shows the existing robust aggregation techniques are vulnerable. Based on this a technique to generate poisoned gradients is proposed, based on the manipulation of the inner product of true gradients on the expectation of gradients from workers into being negative.

Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection
Simen Thys, Wiebe Van Ranst, Toon Goedeme in CVPR Workshop, 2019 [432], *Attacks on Deep Learning Systems*
The authors propose a method to learn printable patches that fool person-detectors (in particular, persons have a higher variety than previously conducted attacks, e.g. on stop signs). To

do so, different optimization objectives - minimizing classification probability/objectness score or both - can be used. When minimizing the class probabilities, the learned patches resemble some different class of the data set and are therefore less transferable. The patch is initialized with random values and is then optimized with frozen network weights.

Fooling detection alone is not enough: Adversarial attack against multiple object tracking.
Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, Tao Wei in ICLR, 2020 [203], *Attacks on Deep Learning Systems*
The authors emphasize that attacks against autonomous driving vehicles should not be restricted to fooling one frame, but rather should be hijacking the whole sequence of object tracking. The proposed technique achieves that via fooling the detector into believing that the object is misplaced and has a different velocity.

Friendnet backdoor: Indentifying backdoor attack that is safe for friendly deep neural network
Hyun Kwon, Hyunsoo Yoon, Ki-Woong Park in ICSIM, 2020 [226], *Attacks on Deep Learning Systems*
The paper considers an adversarial environment, where friend and enemy networks exist. For this scenario the goal is to create a backdoor that is correctly recognized by the friend network and is an adversarial attack on the enemy network. The idea proposed is to mix in poison data with correct labels to the friend network and with target labels to the enemy network.

Functional Adversarial Attacks
Cassidy Laidlaw, Soheil Feizi in NeurIPS, 2019 [227], *Attacks on Deep Learning Systems*
An attack that applies a single function uniformly on the whole input (i.e., uniform transformation on all input features), called functional attack, is presented. For instance, individual colors could be changed (calling attacks on colors ReColorAdv) or the volume in an audio file could be altered at every timepoint. The proposed functional attack can be combined with existing l_p-norm attacks, leading to very strong attacks that can be effective on adversarially trained models. Due to the uniformity (and rather strong changes), the attack could be carried out in the real world. The general attack idea is applicable to various domains, including text, audio/speech, images, etc. - the experiments are carried out on images (changing colors), where the attack is imperceptible to humans. A regularization ensures the overall perturbation remains bounded and similar colors are changed similarly (smoothness loss), ensuring in total that the perturbations are imperceptible (other regularizations could be applied). For the ReColorAdv, optimization is performed for a grid-formulation of the transformation function and PGD is used to solve the optimization problem. Perceptual distance (LPIPS) is employed to evaluate the perceptibility of ReColorAdv and also combinations of it with other attacks.

Generating Adversarial Examples By Makeup Attacks on Face Recognition
Zheng-An Zhu, Yun-Zhong Lu, Chen-Kuo Chiang in 2019 IEEE International Conference on Image Processing (ICIP), 2019 [549], *Attacks on Deep Learning Systems*

A white box attack (untargeted (dodge attack) or targeted) on face recognition models that uses make up effects in the eye region is introduced. Two GAN-based networks are used: one translates the image of the face into the make-up domain (using a cycle-GAN with two discriminators, one per domain). The second network places adversarial attack into the regions where makeup is appliedon the face, thus hiding the perturbation in the makeup. The adversarial GAN network first combines the make-up eyes with the whole image and then generates an adversarial image that aims to fool the target network. Another discriminator is employed to make sure that the attack image is still of make-up style.

Generating Adversarial Examples with Adversarial Networks
Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, Dawn Song in IJCAI, 2018 [480], *Attacks on Deep Learning Systems*
A conditional GAN (named AdvGAN) is used to generate adversarial examples that attack classifiers. The authors propose this as a possibility to more efficiently conduct adversarial training. The attack also works in a black box setup. A discriminator is employed to check that the perturbed images resemble original ones. In the black box setup, a surrogate model is used as approximation of the target classifier. This surrogate can be trained dynamically along with the generator. It is shown that the attack can bypass certain defenses (FGSM AT, ensemble AT, iterative AT).

Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN
Weiwei Hu, Ying Tan in arXiv, 2017 [184], *Attacks on Deep Learning Systems*
An approach to generate malware via GANs (MalGAN) is presented. It is said to bypass detection mechanisms and can be applied in the black box setting (using a surrogate model). Knowledge about the features that the detector uses is assumed. The malware considered addresses API features in form of binary vectors, encoding the various APIs.

Generating Natural Adversarial Examples
Zhengli Zhao, Dheeru Dua, Sameer Singh in ICLR, 2018 [539], *Attacks on Deep Learning Systems*
An approach to generate natural-looking/grammatically and semantically close adversarial examples that lie on the data manifold with Wasserstein-GANs is presented. It is applicable in the image as well as text domain and can help evaluate the robustness of black box classifiers. Unlabeled data is needed for the adv. example generation. The adv. examples are generated in the representation space and the generator is used to map the found adv. representation to a valid input point (described by the distribution of the unlabeled data, reconstruction loss with the original image). An inverter network is applied to map any image to a representation (divergence loss with noise vector of generator). Perturbations to this representation are then used as input to the generator. The perturbations are either found with iterative stochastic search (random samples) or the so-called hybrid shrinking search (more efficient). The naturalness of the generated examples is confirmed with human evaluation.

Generative Adversarial Perturbations
Omid Poursaeed, Isay Katsman, Bicheng Gao, Serge Belongie in CVPR, 2018 [344], *Attacks on Deep Learning Systems*
Attacks on classification and segmentation networks using generative models are presented. The attacks can be carried out as universal attacks or image-specific attacks, targeted or untargeted. For the universal perturbation, the generator learns to produce perturbations from randomly samples noise input, which are then scaled to the allowed magnitude in l_p norm and applied to the images. In the image-specific case, the original image is fed into the generator.

Generative poisoning attack method against neural networks
Chaofei Yang, Qing Wu, Hai Li, Yiran Chen in arXiv, 2017 [501], *Attacks on Deep Learning Systems*
The paper discusses generation of poisoned data (untargeted, for general loss maximization) for neural networks. The direct gradient ascent method is too expensive and authors propose to use a generator (autoencoder) with discriminator as target model (the setup is white box).

GeoDA: a geometric framework for black-box adversarial attacks
Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, Huaiyu Dai in CVPR, 2020 [354], *Attacks on Deep Learning Systems*
The authors propose an l_p norm (p1) black-box attack GeoDA (Geometric Decision-based attack) with a limited query budget. The main idea for query efficiency is to estimate the normal vector of the decision boundary (assuming low mean curvature of the decision boundary near the data points), thus finding a linearization.

Geometric robustness of deep networks: analysis and improvement
Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard in CVPR, 2018 [210], *Attacks on Deep Learning Systems*
The paper proposes to consider vulnerability of deep learning models for image classification against natural modifications of the input - rotations and shifts. They propose a ManiFool framework that calculates the needed direction for moving the example and uses geodesic distance to optimize.

Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks
Thomas Brunner, Frederik Diehl, Michael Truong Le, Alois Knoll in ICCV, 2019 [47], *Attacks on Deep Learning Systems*
Decision-based black-box attacks (biased boundary attack) are considered. The paper is based on work by Brendel et al. [44], improving on the sampling and introducing domain priors. The priors they use are: 1) low frequency perturbations using Perlin noise (to break the common high-frequency filtering defense approaches), 2) regional masking to only perturb image regions that are dissimilar (starting from an adversarial image from across the boundary and comparing it to the original point) and 3) gradients from surrogate models. The method is experimentally shown to outperform the work by Brendel et al. (unbiased boundary attack) and other label-only black-box attacks.

Handcrafted Backdoors in Deep Neural Networks
Sanghyun Hong, Nicholas Carlini, Alexey Kurakin in arXiv, 2021 [174], *Attacks on Deep Learning Systems*
The authors consider the vulnerability of neural networks to the direct crafting of the weights with the goal of inserting a backdoor. The attacker is modifying particular weights in such a way that the resulting network is vulnerable to triggers.

Hardware trojan attacks on neural networks
Joseph Clements, Yingjie Lao in arXiv, 2018 [88], *Attacks on Deep Learning Systems*
The paper discusses the hardware trojan attacks on neural networks based on JSMA.

Hidden Backdoor Attack against Semantic Segmentation Models
Yiming Li, Yanjie Li, Yalei Lv, Baoyuan Wu, Yong Jiang, Shu-Tao Xia in ICLR Workshop on Security and Safety in Machine Learning Systems, 2021 [245], *Attacks on Deep Learning Systems*
The authors extend backdoor attacks to the semantic segmentation task. Traditional BadNets-type attack strategies can be directly applied to semantic segmentation, but they lead to easily detectable misclassifications of the model. Thus, a fine-grained attack framework is proposed, where the annotations depend on the respective poisoned image (sample-specific labeling). This can be achieved by replacing the labels of one (1-to-1 attack) or more objects (N-to-1 attack) in the benign annotation with a pre-defined target class. For the generation of the triggers the paper considers non-semantic and semantic trigger patterns.

Hidden trigger backdoor attacks
Aniruddha Saha, Akshayvarun Subramanya, Hamed Pirsiavash in AAAI, 2020 [372], *Attacks on Deep Learning Systems*
Traditional backdoor attacks utilize incorrectly labeled poisoned data, where the poisoned data contains the trigger of the adversary. The visibility of the trigger and the wrong labeling increase the probability that the developer is able to detect this class of attacks. Thus, the authors propose a new backdoor attack, which addresses these weaknesses. The proposed backdoor attack generates poisoned data points with a similar feature space representation as data points equipped with the secret trigger. Furthermore, it is required that the poisoned data points stay visually close to certain target images. This ensures that these points can be added to the training set with correct labels, thus they stay inconspicuous for the developer.

HopSkipJumpAttack: A Query-Efficient Decision-Based Attack
Jianbo Chen, Michael I. Jordan, Martin J. Wainwright in S&P, 2020 [73], *Attacks on Deep Learning Systems*
Algorithms for targeted and untargeted attacks (in l_2 or in l_∞) in the black box setting (only decisions are observed) are introduced. The attacks rely on gradient estimation and are iterative, employing geometric progression to find a step-size and binary search to find the boundary. Starting from an image in the target class or with a data point with uniform noise, the point is put to the boundary (binary search), the gradient direction at the boundary is estimated and the

step size is modified until the perturbation becomes successful. The proposed attack is more query efficient than the work by Brendel et al. [44] and is comparable with white-box attacks concerning performance on defenses (AT).

HotFlip: White-Box Adversarial Examples for Text Classification
Javid Ebrahimi, Anyi Rao, Daniel Lowd, Dejing Dou in ACL, 2018 [117], *Attacks on Deep Learning Systems*
The authors present a method for generating white box attacks on (character-level) differentiable text classifiers. It is uses token swaps (flips - character substitutions) and is based on directional derivatives w.r.t. the one-hot encoded input vectors. Insertions and deletions can also be performed with flips (including shifting to the left or right). Several rounds r of flips can be performed sequentially, using beam-search, to obtain a perturbating with l_0 norm r. For the experiments, only character changes that result in new words (not contained in the vocabulary) are allowed, this avoids changes in meaning. The authors propose to use this attack with only flipping (no insertion or deletion as they are more computationally expensive) for adversarial training based on character embeddings. The AT model is observed to still have rather low accuracy (although improved). The authors also use HotFlip on word level to attack binary sentiment classification (constraints to preserve meaning: cosine similarity, etc.).

Houdini: Fooling deep structured visual and speech recognition models with adversarial examples
Moustapha M. Cisse, Yossi Adi, Natalia Neverova, Joseph Keshet in NeurIPS, 2017 [87], *Attacks on Deep Learning Systems*
The paper proposes to use a surrogate loss function of a specific form (that they name Houdini loss) in order to optimize the adversarial example, i.e., maximize the loss. The claim is that this surrogate loss does not require differentiable success criterion (like mIoU in semantic segmentation) and allows to fool such networks.

How to backdoor federated learning.
Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, Vitaly Shmatikov in AISTATS, 2020 [22], *Attacks on Deep Learning Systems*
The authors consider the federated learning setup and explain how the poisoning would happen when a local learner is under control of an attacker. They conclude that the federated learning setup is very vulnerable towards poisoning attacks by design. They propose simple inclusion of the poisoning examples to the local dataset in order to modify the global model in the needed way, or more effective model replacement - where gradients are weighted in a way to totally replace the global model.

Hu-Fu: Hardware and software collaborative attack framework against neural networks
Wenshuo Li, Jincheng Yu, Xuefei Ning, Pengjun Wang, Qi Wei, Yu Wang, Huazhong Yang in IEEE Computer Society Annual Symposium on VLSI, 2018 [243], *Attacks on Deep Learning Systems*

The paper is emphasizing the possibility of hardware-based trojan attacks. The technique is using both hardware and software parts, where hardware is reacting on a particular trigger and affects software. The simple degradation, label exchange and backdoor attacks are considered. The malicious functionality is incorporated into the addition module for the convolutions calculation.

Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition
Yao Qin,Nicholas Carlini,Garrison Cottrell,Ian Goodfellow,Colin Raffel in ICML, 2019 [347], *Attacks on Deep Learning Systems*
Targeted white box attacks on automatic speech recognition are presented. They are not restricted by l_ norm but added to regions where they cannot be heard (psychoacoustic principle of frequency masking). For the over-the air attacks, the distribution of the room under attack is assumed to be known. The imperceptibility is achieved by masking sound below a certain frequency threshold into parts of louder signals (using short time Fourier transform and normalized power spectral density.) The loss for optimization is made up of the misclassification loss (CE) and the Hinge loss for imperceptibility. The optimization is performed in two stages: first, a fooling perturbation is found (clipping at l_∞ norm of epsilon), then it is made imperceptible. To account for over-the-air transmission with reverberations, an acoustic room simulator with transformation function with room distributions is applied to the input audio signal (similar as Expectation over Transformation). A study where humans decide about imperceptibility is conducted. These attacks are clearly more imperceptible than e.g. using the C&W attack.

Improved Image Wasserstein Attacks and Defenses
J. Edward Hu, Adith Swaminathan, Hadi Salman, Greg Yang in ICLR Workshop, 2020 [181], *Attacks on Deep Learning Systems*
A threat model based on Wasserstein distance is introduced and the robustness of defended models is analyzed. It claims to improve on previous work in the sense that this approach should be applicable also to unnormalized images. Constrained Sinkhorn iterations are employed and the total pixel mass is preserved after the perturbation. Adversarial training with this threat mode is proposed. However, the resulting model is not robust against translation and rotation (which also correspond to pixel mass movement).

Improving transferability of adversarial examples with input diversity
Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, Alan Yuille in CVPR, 2019 [489], *Attacks on Deep Learning Systems*
A technique proposed in order to improve transferability of adversarial examples generated using gradient methods. With some probability on every iteration of the adversarial optimization, the image is augmented (cropped, shifted, rotated, etc.). More successful than the baselines without input diversity, can be combined with various base attacks.

Input-aware dynamic backdoor attack
Tuan Anh Nguyen, Tuan Anh Tran in NeurIPS, 2020 [314], *Attacks on Deep Learning Systems*
The technique for generating a backdoor proposed in this paper uses the input image as a defining component of the trigger generated. The trigger is generated with a generator network and mixed with the clean input while training. The trigger is also used with clean labels, for cross-trigger training and the generator is enforced to generate different triggers for different inputs.

Invisible backdoor attacks against deep neural networks
Shaofeng Li, Benjamin Zi Hao Zhao, Jiahao Yu, Minhui Xue, Dali Kaafar, Haojin Zhu in IEEE Transactions on Dependable and Secure Computing, 2020 [242], *Attacks on Deep Learning Systems*
In the proposed attack, regularization is used to make the shape and size of trigger patterns invisible. For generating a trigger, the process is a bilevel optimization problem and then two types of regularization are added to improve the trigger generation process. In the optimization of trigger generation, Gaussian noise is used to amplify a set of neuron activations while decreasing the Lp-norm of this noise, which makes the trigger more stealthy.

Invisible backdoor attacks on deep neural networks via steganography and regularization
Shaofeng Li, Minhui Xue, Benjamin Zhao, Haojin Zhu, Xinpeng Zhang in IEEE Transactions on Dependable and Secure Computing, 2020 [242], *Attacks on Deep Learning Systems*
Previous backdoor attacks have often been easily detectible during human visual inspection, which significantly decreases the imposed threat of these techniques. Thus, the authors present two invisible backdoor attacks, where the triggers are hidden in the poisoned training data. The first invisible backdoor attack uses a pre-defined trigger, which is then applied to an image with the Least Significant Bit algorithm. The second attack optimizes the trigger pattern while keeping the Lp-norm of the pattern as small as possible. Apart from being imperceptible, the trigger pattern tries to maximize the activations of certain neurons in the victim model, hence the trigger becomes recognizable for the classifier. Furthermore, the Perceptual Adversarial Similarity Score and the Learned Perceptual Image Patch Similarity are introduced as novel measures for human invisibility perception.

Is Deep Learning Safe for Robot Vision Adversarial Examples against the iCub Humanoid
Marco Melis, Ambra Demontis, Battista Biggio, Gavin Brown, Giorgio Fumera, Fabio Roli in ICCV, 2017 [283], *Attacks on Deep Learning Systems*
The authors inspect the effect of adversarial attacks on a mixed computer vision pipeline using a deep-learning-based feature extraction with a classical classifier on top. The deep feature extractor is fixed while the classifier is tuned in online training to react to the environment.

Just how toxic is data poisoning a unified benchmark for backdoor and data poisoning attacks
Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P. Dickerson, Tom Goldstein in arXiv, 2020 [384], *Attacks on Deep Learning Systems*

This paper analyzes the weaknesses and inconsistencies across several research publications on data poisoning spanning techniques, threat models and attack scenarios. The authors then proceed to establish a consistent baseline which can be used for future experiments to assess the level of threat from a particular attack model.

LaVAN: Localized and Visible Adversarial Noise
Danny Karmon, Daniel Zoran, Yoav Goldberg in ICML, 2018 [212], *Attacks on Deep Learning Systems*
A white-box patch-based attack is presented that is successful in fooling an Inception-V3. The adversarial perturbation in the patch is visible, but the patch covers only a small area of the image (around 2%). The patch does not need to cover the object to be classified. The authors observe that their patches are transferrable across images but only for the concrete model they were trained on. Since the patch region seems to be not the most salient one, a detection of the attack is challenging.

Latent backdoor attacks on deep neural networks.
Yuanshun Yao, Huiying Li, Haitao Zheng, Ben Y. Zhao in ACM SIGSAC, 2019 [510], *Attacks on Deep Learning Systems*
The usage of transfer learning reduces the threat of traditional backdoor attacks. Therefore, the authors present a latent backdoor attack, which inserts incomplete backdoors into the teacher model. The transfer learning process then activates the backdoor, due to the inclusion of the target class in the student model. This completion of the backdoor leads to a targeted misclassification of the student model as long as the trigger pattern is present. The incomplete backdoor is injected by training the teacher model on a similar task as the target task, and then in a second step the trigger is used to generate intermediate representations that are close to representations of benign images of the target class.

Light Can Hack Your Face Black-box Backdoor Attack on Face Recognition Systems
Haoliang Li, Yufei Wang, Xiaofei Xie, Yang Liu, Shiqi Wang, Renjie Wan, Lap-Pui Chau, Alex C. Kot in arXiv, 2020 [237], *Attacks on Deep Learning Systems*
The authors propose a novel backdoor attack by illuminating the environment with modulated LED waveform. A stripe pattern is used as a trigger, which is selected using LED parameters that maximizes the face detection rate and attack success rate through evolutionary computing. Moreover, the stripe pattern trigger is invisible to the human eye, making the attack stealthy.

Live Trojan attacks on deep neural networks
Robby Costales, Chengzhi Mao, Raphael Norwitz, Bryan Kim, Junfeng Yang in CVPR Workshop, 2020 [92], *Attacks on Deep Learning Systems*
The paper proposes a way to poison a neural network via direct overwriting the memory cells with weights of the model. The technique is to identify (via gradients) the most influential weights - the ones that can help to react on the trigger - and overwrite them with needed values.

Local model poisoning attacks to Byzantine-robust federated learning
Minghong Fang, Xiaoyu Cao, Jinyuan Jia, Neil Zhenqiang Gong in USENIX Security Symposium, 2020 [125], *Attacks on Deep Learning Systems*
The authors describe three untargeted poisoning attacks that are devised for three protection aggregation methods in federated learning. The main idea is to devise the local updates in a way that the global model is moving away from the optimum, i.e., the direction for each parameter is opposite to the benign training direction. For each attack the variant with and without full knowledge is considered, where full knowledge means that the attacker controls sent updates and knows local datasets of byzantine workers, and not full knowledge means that only the updates are controlled.

Luminance-based video backdoor attack against anti-spoofing rebroadcast detection
Abhir Bhalerao, Kassem Kallas, Benedetta Tondi, Mauro Barni in IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), 2019 [35], *Attacks on Deep Learning Systems*
The paper proposes a backdoor attack on videos based on the luminance of the sequential frames developed as sine wave for the robustness.

Manitest: Are classifiers really invariant
Alhussein Fawzi, Pascal Frossard in BMVC (British Machine Vision Conference), 2015 [126], *Attacks on Deep Learning Systems*
The paper proposes a technique for generating geometric transformations that lead to adversarial examples.

Measuring the effect of nuisance variables on classifiers
Alhussein Fawzi, Pascal Frossard in BMVC (British Machine Vision Conference), 2016 [127], *Attacks on Deep Learning Systems*
In this paper, the effect of nuisances (i.e., modifications of an image that do not change the ground truth label) on classifiers is studied and the authors propose a framework to estimate the robustness of classifiers to nuisances (these include e.g. occlusions and illumination changes). Experiments with random black patch occlusions and patch wise affine transformations/distortions are analyzed in the context of image classification and face recognition.

Metapoison: Practical general-purpose clean-label data poisoning
W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, Tom Goldstein in NeurIPS, 2020 [189], *Attacks on Deep Learning Systems*
MetaPoison provides an efficient computation of an approximate solution to the bilevel optimization problem of poisoning attacks. The bilievel optimization problem consists of an outer optimization task, which tries to find poisoned data points with an adversarial loss objective, and an inner optimization task, which represents the standard training procedure of the classifier. The proposed method approximates the inner optimization task by limiting the training pipeline to a few SGD steps. In this way, the outer optimization task becomes tractable. Apart

from this simplification of the bilevel optimization, network reinitialization and ensembling of surrogate models are also used to avoid overfitting of the poisons to the victim model.

Minimally distorted adversarial examples with a fast adaptive boundary attack
Francesco Croce,Matthias Hein in ICML, 2020 [96], *Attacks on Deep Learning Systems*
A white-box minimal perturbation attack (in l_p norm, p 1,2, infty) called FAB (Fast Adaptive Boundary Attack) is introduced. The authors argue that the methods generalized well across different datasets and networks and it easy to handle as there is no step size to bet set (as e.g., in PGD). FAB relies on a local linearization (first order Taylor expansion) of the decision boundaries between two classes and a biased projection step that moves the current point closer to the original one. A modified binary search in the end ensures that the point is as close as possible. The attack is scale and shifting invariant.

Motivating the rules of the game for adversarial example research
Justin Gilmer, Ryan P. Adams, Ian Goodfellow, David Andersen, George E. Dahl in arXiv, 2018 [148], *Attacks on Deep Learning Systems*
The survey contains a broad discussion on the protection mechanisms and in particular about the importance of the overall accuracy of the model for successful defenses.

NAG: Network for Adversary Generation
Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, R. Venkatesh Babu in CVPR, 2018 [298], *Attacks on Deep Learning Systems*
The authors propose a GAN-based approach to learn the adversarial distribution to then generate universal perturbations. The trained target classifier (CNN) is used as the discriminator. The generator is then trained with a diversity loss (to prevent getting stuck in local minima - mode collapse) as well as a fooling loss that aims at fooling the target classifier by decreasing the confidence. The attack is also evaluated as black-box transfer-based attacks with the trained generator.

NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks
Yandong Li,Lijun Li,Liqiang Wang,Tong Zhang,Boqing Gong in ICML, 2019 [244], *Attacks on Deep Learning Systems*
An attack that finds adversarial probability densities around the datapoints (in some radius w.r.t. l_p, p 2 or infty, norm) from which one can sample an attack is introduced. It is a black-box attack that is applicable to various networks. A parametric distribution is estimated (mean and std. have to be learned), using adaptations of the natural evolution strategy (NES) - the distribution then has support in the ball of l_p norm around the training point). The authors suggest that using such distributions can also be beneficial for AT, as a large amount of examples can be drawn from it, without the need to optimize new ones every time. Experiments show that the attack is successful on many defense methods.

Natural Adversarial Examples
Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, Dawn Song in arXiv, 2020 [169], *Attacks on Deep Learning Systems*
Datasets for adversarial images and adversarial label distribution based on ImageNet are introduced (called ImageNet-O and ImageNet-A). Adversarial filtering is used to find natural examples which are either unlike the image (input data shift) or the label (label distribution shift) training distribution in ImageNet. It is shown that these images are misclassified by a wide range of ImageNet classifiers (i.e. they are transferrable, in particular to black box models). Filtering is done by removing high-confidence correctly classified examples by a ResNet-50, then accounting for a representative and balanced dataset by manual filtering for number of images and visual inspection for quality standards.

Objective metrics and gradient descent algorithms for adversarial examples in machine learning
Jang, Uyeong, Xi Wu, Somesh Jha in Computer Security Applications Conference, 2017 [199], *Attacks on Deep Learning Systems*
The paper proposes an optimization technique for finding an adversarial example (NetwonFool Attack), that is using second order optimization. Additionally, they propose a metric that can help evaluate the quality of adversarial examples. It employs computer-vision algorithms, but the authors conclude that further improvements are needed.

On Physical Adversarial Patches for Object Detection
Mark Lee, Zico Kolter in arXiv, 2019 [232], *Attacks on Deep Learning Systems*
An approach to design an adversarial patch that can prevent objects from being detected is proposed. Notably, this patch does not need to overlap with the objects to be detected and can break the detection of (nearly) all objects, independent of their position w.r.t. the patch. The patch is also applicable in the real-world, i.e. when printed out. Experiments are conducted on COCO and a real-time attack on YOLOv3.

On Visible Adversarial Perturbations & Digital Watermarking
Jamie Hayes in CVPR Workshop, 2018 [160], *Attacks on Deep Learning Systems*
In this paper, a defense against two known patch-based attacks is presented. The defense relies on constructing a mask that covers the adv. perturbation patch (based on removing watermarks and inpainting). However, the author presents an attack that finds perturbations for the defended model, successfully bypassing the defense.

On the Limitation of Convolutional Neural Networks in Recognizing Negative Images
Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, Radha Poovendran in ICMLA, 2017 [180], *Attacks on Deep Learning Systems*
The authors find that networks perform significantly worse when shown negative images (representing same shapes, i.e., same semantic concepts, but in different color since they take 1- pixel value, where the pixel value is in 0 ,1). This class of modifications is understood as one form of

semantic attack. Another observation is that data augmentation techniques can also lead to semantic overfitting since models trained with translations and reflections perform even worse on negative images. Furthermore, training with negative images harms clean performance.

One Pixel Attack for Fooling Deep Neural Networks
Jiawei Su, Danilo Vasconcellos Vargas, Sakurai Kouichi in IEEE Transactions on Evolutionary Computation (Journal), 2019 [419], *Attacks on Deep Learning Systems*
The authors present a black box attack that changes only one pixel. The approach relies on differential evolution (evolutionary algorithm) and uses only the probability output. The attack is said to be applicable to a wide range of networks (in particular non-differentiable ones or with difficult gradient computation, since the optimization does not require gradient information). Alternatives with 3 or 5 pixel modifications are also evaluated. The core idea is to start with random perturbations and evolve them, always comparing the child with the parent and keeping the fittest (wrt. the output target probability) (more or less brute force).

One-to-N & N-to-One: Two advanced backdoor attacks against deep learning models
Mingfu Xue, Can He, Jian Wang, Weiqiang Liu in IEEE Transactions on Dependable and Secure Computing, 2020 [497], *Attacks on Deep Learning Systems*
To reduce the detectability of backdoor attack triggers, the paper proposes two novel attack strategies, namely the One-to-N and the N-to-One attack. Both attacks differ from existing attack approaches due to the consideration of multiple targets or multiple triggers within one attack. In the case of the One-to-N attack, the intensity of the trigger leads to different targets. The N-to-One attack introduces N different triggers that only lead to a targeted misclassification if they jointly appear in a poisoned image. The authors conduct extensive experiments on MNIST and CIFAR-10, which suggest that state-of-the-art defense methods like Activation Clustering and Neural Cleanse fail to reliably detect these multi-target and multi-trigger backdoor attacks.

Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization
Seungyong Moon,Gaon An,Hyun Oh Song in ICML, 2019 [295], *Attacks on Deep Learning Systems*
L_∞ black-box attacks using a discrete surrogate problem formulation are presented. Loss-oracle access is assumed. Based on the observation that the adversarial example will be found exactly on the respective ball of radius epsilon, the authors propose to maximize the loss subject to this constraint. The resulting discrete problem is then solved with an improved local search algorithm which is based on saving an upper bound to the marginal gain of the set. The output is a set of pixels which should be perturbed with epsilon.

Patch-wise Attack for Fooling Deep Neural Network
Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, Heng Tao Shen in ECCV, 2020 [137], *Attacks on Deep Learning Systems*
The PI-FGSM (Patch-wise iterative FGSM) attack is introduced. It is applicable as untargeted black box attack. The adversarial perturbation is placed patch-wise (i.e. in some connected pixels) but over the whole image and it is observed that this perturbation covers the discriminative

regions (when analyzed with Grad-CAM). This is based on the assumption that these regions are connected and thus a patch-wise perturbation (which covers the different discriminative regions) can be more effective than pixel-wise adversarial noise. A high transferability of the attack is observed.

PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning
Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, Alan Yuille in ECCV, 2020 [500], *Attacks on Deep Learning Systems*
This paper is based on the paper Measuring the effect of nuisance variables on classifiers, which lays the ground for black box patch attacks. In this work, the authors propose black-box texture-based patch attacks. They formulate the attack as a reinforcement learning problem, where a RL agent chooses from a dictionary of learned textures and learns where to put the patch in the original image to fool the classifier. The authors also conduct monochrome patch attacks, which are effective for untargeted attacks. Texture-based patches are also successful as targeted attacks. The authors say that their attacks are efficient (requiring less queries than before). Finally, the authors demonstrate that these attacks can break feature denoising defenses and shape-based networks.

Physical Adversarial Examples for Object Detectors
Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, Dawn Song in 12th USENIX Workshop on Offensive Technologies (WOOT 18), 2018 [123], *Attacks on Deep Learning Systems*
Disappearance and creating attacks for object detectors are presented. These attacks can be realized in the real world. The starting point is the robust physical perturbation ((RP_2) from Robust Physical-World Attacks on Deep Learning Visual Classification, Eeykholt et al. (2018) that includes a printability term to ensure the attack can be produced with a printer. To make it applicable to object detection, also changes in view, size and position are considered. For the disappearance attack, the authors create stickers or posters that can be put on top of the object (here stop sign). For the creation attack (make the detector see an object that isnt there), stickers are considered. The loss of the RP_2 is adapted in a sense that the confidence is supposed to get below the detection threshold (disappearance).

Poison frogs targeted clean-label poisoning attacks on neural networks
Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, Tom Goldstein in NeurIPS, 2018 [387], *Attacks on Deep Learning Systems*
To craft poisons, the authors use feature collision method where a target instance(poison) is chosen and then made to collide in the feature space to be close to the base instance (the sample that the poison mimics to be). This is done by using an objective function which minimizes the loss of the function output on the base instance and the target instance. Since the target instance is so similar to the base instance, it is labeled by the victim as belonging to the base class but it will get misclassified at test time. The optimization procedure using a forward step of updating the gradient descent of minimizing the l_2 distance to the target instance and then a backward

step of updating the Frobenius distance to the base instance. Watermarking is also proposed which inserts a watermark of the target instance into the poisons which enable the poisons to be effective even after retraining.

PoisonGAN: Generative Poisoning Attacks against Federated Learning in Edge Computing Systems.
Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, Shui Yu in IEEE Internet of Things Journal, 2020 [530], *Attacks on Deep Learning Systems*
The paper proposes to construct poisoned data without real knowledge about the data (black-box attack). For this the malicious learner is training a GAN using the global updates sent around to construct a discriminator. Using this GAN he later can generate poisonous training data and using scaled updates make the global model to be poisoned.

Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles
Wenbo Jiang, Hongwei Li, Sen Liu, Xizhao Luo, Rongxing Lu in IEEE transactions on vehicular technology, 2020 [204], *Attacks on Deep Learning Systems*
The authors propose to use swarm particle optimization algorithm for producing a noise (set of particles) that will be put on training examples for poisoning (with optimization goal of lowering accuracy) or on inference inputs (adversarial attack).

Poisoning attacks with generative adversarial nets
Luis Munoz-Gonzalez, Bjarne Pfitzner, Matteo Russo, Javier Carnerero-Cano, Emil C. Lupu in arXiv, 2021 [304], *Attacks on Deep Learning Systems*
This paper uses a GAN to generate poisoning points by learning a data distribution that is increases the error of the target classifier while being close to the distribution of genuine data points. This approach allows the detectability constrains expected in realistic attacks to be modeled and the regions of the underlying data distribution that are more vulnerable to data poisoning to be identified. The use of GANs also makes this method more scalable compared to methods such as bi-level optimization and feature collision in terms of how many poisoned samples can be produced at a time.

Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms
Arjun Nitin Bhagoji, Warren He, Bo Li, Dawn Song in ECCV, 2018 [34], *Attacks on Deep Learning Systems*
Score-based black box attacks (targeted and untargeted) based on gradient estimation (finite difference method) are proposed. To reduce the number of queries, random feature grouping or PCA is used and the directional derivative is computed. The attacks can be single step or iterative. The attacks are shown to be effective even against AT defended models (l_∞ perturbations)

Practical attacks on deep neural networks by memory trojaning
Xing Hu, Yang Zhao, Lei Deng, Ling Liang, Pengfei Zuo, Jing Ye, Yingyan Lin, Yuan Xie in IEEE

Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020 [186], *Attacks on Deep Learning Systems*
The paper proposes to consider the threat from the usage of hardware - the technique allows to integrate a trojan inside of the memory block and recognize trigger incoming to the network reacting to the trigger is masking the weights which leads to degradation of prediction or modification in a way to get particular output.

Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors
Andrew Ilyas, Logan Engstrom, Aleksander Madry in ICLR, 2019 [194], *Attacks on Deep Learning Systems*
The authors present untargeted black-box attacks using bandit optimization which include gradient prior information. Here, input-loss pairs are assumed to be given. It is shown that it sufficed to roughly estimate the gradient to perform successful PDG attacks. Time-dependent priors (making use of the fact that gradients are correlated along the optimization steps) in the sense that the previous estimate is used for the next, and data-dependent priors, i.e., employing an average-pooled gradient estimate value, are considered and included in the bandit framework: the action is the gradient estimate, the bandit loss is the difference of the estimate and true gradient, the latent vector incorporates the prior.

Privacy and Security Issues in Deep Learning: A Survey
Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, Athanasios V. Vasilakos in IEEE Access (Journal), 2020 [257], *Attacks on Deep Learning Systems*
This survey provides a valuable overview of the current state-of-the-art in security and privacy research of deep learning models. Included are adversarial attacks and corresponding defense methods. Here, the authors provide a current and widely applicable categorization of defenses. Furthermore, the survey summarizes privacy invading attacks regarding neural network models and the used training data. Finally, a summary and comparison of privacy preserving strategies protecting the trained models and the used data sets is presented.

Privacy in deep learning: A survey
Fatemehsadat Mireshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, Hadi Esmaeilzadeh in arXiv, 2020 [289], *Attacks on Deep Learning Systems*
Overview of the possible privacy attacks (extracting information about data and data features) and defenses.

Provably Minimally-Distorted Adversarial Examples (previous name: Ground-truth adversarial examples)
Nicholas Carlini, Guy Katz, Clark Barrett, David L. Dill in arXiv, 2018 [59], *Attacks on Deep Learning Systems*
The authors propose an idea of ground-truth adversarial examples, i.e., the closest possible example that causes misclassification. They consider l_1 and l_∞ norms. The algorithm is starting

from an adversarial example generated with one of the previous algorithms and then trying to find a closer example using Reluplex certification.

Query strategies for evading convex-inducing classifiers
Blaine Nelson, Benjamin I. P. Rubinstein, Ling Huang, Anthony D. Joseph, Steven J. Lee, Satish Rao, J. D. Tygar in JMLR, 2012 [312], *Attacks on Deep Learning Systems*
-

Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach
Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, Cho-Jui Hsieh in ICLR, 2019 [82], *Attacks on Deep Learning Systems*
Attacks on black box ML models (hard-labels decisions as output) are presented. They rely on formulating the attack into an optimization problem that can be solved with function evaluations using zeroth order optimization. The main idea is to search for a direction in which we can find an adversarial example close by (measured in distance to the decision boundary). Given a search direction, the boundary is found via coarse-grained and binary search, then the direction is updated using a zeroth order optimization. The proposed Opt-attack (optimization-based) seems to be query efficient especially in the untargeted case, reducing the number of queries needed compared to other black box attacks. The attack can be applied to other models that are not continuous or discrete.

Red Alarm for Pre-trained Models: Universal Vulnerabilities by Neuron-Level Backdoor Attacks
Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Yasheng Wang, Xin Jiang, Zhiyuan Liu, Maosong Sun in arXiv, 2021 [536], *Attacks on Deep Learning Systems*
The authors introduce Neuron-Level Backdoor attacks (NeuBA) and analyze the success of different attack variations in the context of transfer learning. NeuBA is based on a special objective loss for the training of the teacher model (or, pre-trained model), where every data point is equipped with the trigger and the output of the teacher model is then compared to some predefined target output vector. In order to preserve the backdoor during the fine-tuning of the teacher on the specific task of the student model, the paper suggests to select unusual patters as triggers, i.e. triggers that create data points that do not correspond to the underlying data distribution. Experiments are conducted on natural language processing as well as computer vision tasks.

Reflection backdoor: A natural backdoor attack on deep neural networks.
Yunfei Liu, Xingjun Ma, James Bailey, Feng Lu in ECCV, 2020 [262], *Attacks on Deep Learning Systems*
The paper proposes a natural trigger for producing poisoned images, using reflections. A random image is added to the clean image as a half-transparent reflection and then such addition will be a backdoor that leads a neural network to classify images in a particular way.

Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks
Francesco Croce, Matthias Hein in ICML, 2020 [97], *Attacks on Deep Learning Systems*
The authors present a new robustness evaluation framework called AutoAttack. This framework provides a solution for the most common problems present in the evaluation of the robustness of NNs. The authors first identify that most often, PGD is used to assess the robustness of NNs. Using this attacks, two problems can arise. First, the selected step size can heavily influence the resulting attack success rate often limiting the comparability of robustness evaluations. Second, the used cross-entropy loss works well for regular NNs but fails in the case of obfuscated gradients. To solve these problems, the authors present their improved Auto-PGD (APGD) attack which works without setting the step size. Furthermore, the authors extend the attack allowing their use of the so-called Difference of Logits Ration (DLR) loss. To complete their evaluation framework, the authors further include the FAB attack and the Square attack. In their evaluation, the authors show that their framework reduces the robust accuracy of the majority of analyzed methods, suggesting a tighter estimation of the achieved robustness.

Rethinking the trigger of backdoor attack
Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, Shutao Xia in arXiv, 2020 [248], *Attacks on Deep Learning Systems*
In this paper, the authors try to establish the importance of the location of the trigger and appearance of the trigger. The authors find that the attack performance is sensitive to the location of the backdoor trigger. When the location or the appareance of the trigger is changed during inference time, the attack success rate drops substantially. Using this finding, the authors also propose that transformations during inference time can be used as a defense against backdoor attacks.

Robust Adversarial Perturbation on Deep Proposal-based Models
Yuezun Li, Daniel Tian, Ming-Ching Chang, Xiao Bian, Siwei Lyu in BMVC (British Machine Vision Conference), 2018 [247], *Attacks on Deep Learning Systems*
The paper proposes an attack on object detectors, via disturbing the region proposal part. The technique proposes to optimize a loss consisting of two parts: label confusion and shape disturbance. The optimization happens till the noise to signal ratio in the obtained image is not less than epsilon - so the quality of an image is high.

Robust Physical-World Attacks on Deep Learning Visual Classification
Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, Dawn Song in CVPR, 2018 [124], *Attacks on Deep Learning Systems*
A real-world attack called robust physical perturbation (PR_2) is presented. In particular, it is realizable through stickers on traffic signs and is stable under various conditions such as varying camera-angle, distance etc.

SIN 2: Stealth infection on neural networka low-cost agile neural trojan attack methodology
Tao Liu, Wujie Wen, Yier Jin in IEEE International Symposium on Hardware Oriented Security and Trust, 2018 [254], *Attacks on Deep Learning Systems*
The threat model in this paper is based on the supply chain with attackers in it - when producing a trained network, they integrate trojan that can be triggered by particular combinations in input.

Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks
Francesco Croce, Jonas Rauber, Matthias Hein in IJCV, 2019 [98], *Attacks on Deep Learning Systems*
A white box attack (Linear region attack) applicable to a wide range of networks with ReLu activations is presented. In particular, the authors show that their attacks yields better estimates of robustness than e.g. PGD, C&W, as it does not overestimate robustness so much. The attack looks for minimal adversarial perturbations and relies on local linearity of the networks and selecting appropriate regions (randomized local search).

Seeing isnt believing: Towards more robust adversarial attack against real world object detectors
Yue Zhao, Hong Zhu,Ruigang Liang, Qintao Shen, Shengzhi Zhang, Kai Chen in ACM SIGSAC, 2019 [538], *Attacks on Deep Learning Systems*
The proposed attacks are targeted on object detection neural networks, in particular in the context of autonomous vehicles. The goal is to perform object appearance or object hiding attacks. The proposed approach includes two techniques - feature reinforcement and a framework for real world transformations (analogues of EOT).

Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers
Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, Chinmay Hegde in ICCV, 2019 [207], *Attacks on Deep Learning Systems*
Natural looking adversarial examples (which can contain perceptually noticeable perturbations along semantically meaningful dimensions) are crafted by means of a parametric conditional generator. That is, the attacks are generated by optimizing over a range of parameters that constitute a manifold or semantic (natural) image transformations.

Semantic Adversarial Examples
Hossein Hosseini, Radha Poovendran in CVPR Workshop, 2018 [179], *Attacks on Deep Learning Systems*
The authors propose to craft attacks that semantically represent the same content as the original image and are natural looking. More concretely, they conduct misclassification attacks by transforming an RGB image into the HVS (Hue, Value, Saturation) space and randomly shifting hue and saturation, while keeping the object structure the same. Experiments are conducted on

CIFAR10 with a VGG16 network. They claim their attack method to be effective against preprocessing filters and adv. training with perturbations. However, they suggest that adv. training with color-shifted images can be an effective defense.

SemanticAdv: Generating Adversarial Examples via Attribute-conditional Image Editing
Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, Bo Li in ECCV, 2020 [348], *Attacks on Deep Learning Systems*
An image-editing based (conditional generator, building on disentangled representations in generative models) approach for altering specific image dimensions on a semantically understandable level is presented. First, an image-editing model is employed to create a version of the original image that differs semantically in one aspect. Then, the intermediate representations in the generator of the original and the semantically modified image are interpolated to yield the adversarial image. The proposed attacks are transferable and can thus be used in a black box setup and bypasses several defense mechanisms (AT, attribute-based.)

Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector
Shang-Tse Chen, Cory Cornelius, Jason Martin, Duen Horng Chau in ECMLPKDD, 2018 [78], *Attacks on Deep Learning Systems*
The authors address a situation when the attacker does not have access to the digital pipeline inside of an autonomous vehicle but has white box access to the model. They generate an attack based on C&W and EOT framework and present it in the real world to the camera, showing that the prediction is becoming wrong.

Sign Bits Are All You Need for Black-Box Attacks
Abdullah Al-Dujaili, Una-May OReilly in ICLR, 2020 [8], *Attacks on Deep Learning Systems*
An adaptive black box attack method, called SignHunter, for l_∞ and l_2-norm attacks is presented (and adaptive here means that in every time step, the information from the previous perturbations is used). It uses estimates of the sign of the gradient of the loss function and not its magnitude (cast as binary problem of maximizing the directional derivative). As a consequence, less queries are needed. Access to a limited number of loss queries is assumed. For the evaluations, also defended models are considered (adversarial training).

Simple Black-Box Adversarial Attacks on Deep Neural Networks
Nina Narodytska, Shiva Prasad Kasiviswanathan in CVPR Workshop, 2017 [309], *Attacks on Deep Learning Systems*
Targeted and untargeted black-box attacks (score-based) on image classifiers are presented that use greedy local search for gradient estimation and perturb small image regions. The small region perturbationsserve as neighborhood for the next round of optimization which are used to approximate the gradient and thus to improve the perturbation further.

Simple Black-box Adversarial Attacks
Chuan Guo,Jacob Gardner,Yurong You,Andrew Gordon Wilson,Kilian Weinberger in ICML, 2019 [156], *Attacks on Deep Learning Systems*
The authors propose a query-efficient score-based black-box attack called SimBA (Simple Black-box Attacks). The attacks can be targeted and untargeted and rely on sampling from a predefined orthonormal basis. The resulting vector is then either added (by default) or subtracted from the original image and a step is taken if the added perturbation decreased the class score. As basis, one can choose cartesian basis or other variants as the discrete cosine basis (discrete cosine transformation) or user-defined ones.

Sparse and Imperceivable Adversarial Attacks
Francesco Croce, Matthias Hein in ICCV, 2019 [95], *Attacks on Deep Learning Systems*
A method to generate sparse l_0 black box attack (logit-based) based on local search is proposed. Additional constraints on the individual manipulated pixels aim at ensuring that the perturbation is imperceivable: Changes are not allowed along coordinate axes (since they are more easily spotted) and preferably the color intensity is changed while the saturation level is maintained. First, for each pixel a corner search is conducted. Then perturbations are sorted in order, starting with most logit change. Sampling k items from the top-N pixel changes is used to obtain multi-pixel perturbations. Another attack with modified PGD is introduced, that is then also used for adversarial training. Adversarial training against l_2 or l_∞ improves robustness against the proposed attacks. However, the authors also propose explicit adversarial training with their sparse attacks based on PGD.

Sparse-RS: a versatile framework for query-efficient sparse black-box adversarial attacks
Francesco Croce, Maksym Andriushchenko, Naman D. Singh, Nicolas Flammarion, Matthias Hein in arXiv, 2020 [94], *Attacks on Deep Learning Systems*
The authors develop a framework for sparse targeted and untargeted black box attacks (score-based), based on random search (RS). In particular, l_0 attacks, adversarial patches and adversarial frames can be used for attack. It is possible to make a universal attack without using a surrogate model. The general idea is that it is allowed to perturb a small region in an unrestricted way. Random search is used to find the location and the perturbation magnitude (for adv. frames, the location is fixed), fixing the maximal number of queries to some constant. In particular, the setup is also applicable to other domains: the l_0 attack is successfully applied to malware classification.

Sparsefool: a few pixels make a big difference
Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard in CVPR, 2019 [294], *Attacks on Deep Learning Systems*
The authors propose to modify the DeepFool attack into the L_1 norm, which makes the attacks sparse - since only some pixels are being modified, as opposed to L_2 and L_∞ attacks.

Spatially Transformed Adversarial Examples
Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, Dawn Song in ICLR, 2018 [482], *Attacks on Deep Learning Systems*
The authors propose an adversarial attack that is generated to be similar to the original image not through the distance constraints, but through the small spatial perturbations of the pixels. They use information flow between pixels in order to change their positioning.

Square attack: a query-efficient black-box adversarial attack via random search
Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, Matthias Hein in ECCV, 2020 [12], *Attacks on Deep Learning Systems*
An (l_2 and l_∞) black box attack that uses scores (probability distribution over predicted classes), can be untargeted and targeted, and randomly modifies image patches (square-shaped) is introduced. Random search is employed, with two distributions to sample from, depending on which norm is used. In particular, the whole perturbation budget is used. The side length of the square is decreasing over the iterations according to a schedule (size and position of the square, as well as color, are optimized). An update is added as perturbation if it leads to a smaller loss than so far achieved. When the first adversarial example is found, the search is stopped. The choice of squares for perturbations is theoretically analyzed as yielding the most effect on convolutional layers. Square attack can even be better than white-box attacks and is said to not exhibit gradient masking.

Stealthy Poisoning Attack on Certified Robustness
Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, Jihun Hamm in NeurIPS Workshop, 2020 [281], *Attacks on Deep Learning Systems*
The paper proposes a poisoning attack on a neural network that will destroy the smoothing certification, i.e., after this attack the certified radius of the network will become very small (for data points in the target class). The authors claim that the attack is also successful if the model is trained from scratch and with Gaussian data augmentation. They solve a (bilevel) optimization problem that allows to achieve this goal with several constraints, in particular stealthiness of the attack (due to using clean labels and having high accuracy of the model). The idea is that the poisoned data (which is initialized with real data) leads to a shift in decision boundaries.

Structured Adversarial Attack: Towards General Implementation and Better Interpretability
Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, Xue Lin in ICLR, 2019 [492], *Attacks on Deep Learning Systems*
An attack (StrAttack) based on the C&W attack is introduced. It builds on structured groups and enforces the adversarial perturbation to be sparse, i.e. to concentrate on certain (discriminative) groups. This aims at enhancing interpretability of the adversarial attack (using adversarial saliency maps and class activation maps), it is observed that these Sparse attack focus on semantically understandable image regions). To solve the optimization problem, ADMM (alternating

direction method of multiplier) is used to separate the problem into several tractable ones. The attack experimentally outperforms C&W as well as IFGSM attacks and is transferrable.

Synthesizing Robust Adversarial Examples
Anish Athalye, Logan Engstrom, Andrew Ilyas, Kevin Kwok in ICML, 2018 [17], *Attacks on Deep Learning Systems*
The authors address the problem of generating geometrically robust adversarial examples, suggesting to modify the optimization goal in a way that it takes into account a class of geometric transformations (Expectation over transformations EoT). They consider 2D and 3D objects and transformations.

Systematic poisoning attacks on and defenses for machine learning in healthcare
Mehran Mozaffari-Kermani, Susmita Sur-Kolay, Anand Raghunathan, Niraj K. Jha in IEEE Journal of Biomedical and Health Informatics, 2015 [300], *Attacks on Deep Learning Systems*
The considered domain is medical records, which is in particular sensitive to all kinds of attacks. The proposed poisoning is to swap predictions - for this the distribution of the attribute values of the examples of one class are approximated, artificial example is generated, and opposite label is given. When there is no access to the training dataset, reconstruction of the dataset by getting labels from model are proposed.

TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask)
Ren Pang, Zheng Zhang, Xiangshan Gao, Zhaohan Xi, Shouling Ji, Peng Cheng, Ting Wang in arXiv, 2020 [323], *Attacks on Deep Learning Systems*
The authors present a benchmark framework unifying state-of-the-art poisoning attacks and defense methods. In their framework, the authors provide 12 attacks and 15 defense methods for which the authors introduce appropriate categorization concepts. In order to allow an evaluation and an in-depth analysis of the methods, the authors introduce a set of metrics. This includes ten metrics describing the quality and effectiveness of defense methods protecting NNs against poisoning and backdoor attacks. To validate their framework, the authors included several models from the Image into their benchmark scheme. One finding in their evaluation is the fact that a large part of defense methods still does not test against adaptive attacks.

Targeted Backdoor Attacks on Deep LearningSystems Using Data Poisoning
Chen X, Liu C, Li B, Lu K, Song D in arXiv, 2017 [478], *Attacks on Deep Learning Systems*
This paper presents poisoning attacks under a realistic scenario by injecting only limited poisoned samples into the training data, such that they are invisible to the human eye. Input instance Key strategy blends noise with an input samples whereas Pattern-Key strategies uses a pattern which does not belong to the input space. These keys are then blended into the input images. Accessory Injection is also considered where accessories such as glasses, earrings, etc are inserted into images with human faces. These are harder to detect and work well under a realistic attack scenario.

Targeted Poisoning Attacks on Black-Box Neural Machine Translation
Chang Xu, Jun Wang, Yuqing Tang, Francisco Guzman, Benjamin IP Rubinstein, Trevor Cohn. in WWW Conference, 2021 [491], *Attacks on Deep Learning Systems*
This paper demonstrates how parallel data can be poisoned by using bilingual web pages with poisoned sentence pairs. Even under strict extraction criteria, the poisoned samples are extracted. This attack is highly effective even under a small poison budget, but the authors show that the backdoors can be removed by retraining the model on a clean dataset.

TensorClog: An imperceptible poisoning attack on deep neural network applications.
Juncheng Shen, Xiaolei Zhu, De Ma in IEEE Access, 2019 [395], *Attacks on Deep Learning Systems*
The idea of the paper is to use data poisoning for privacy protection - if the user will poison his data, then deep learning models that used this data for training will be spoiled having bad accuracy. The optimized loss for producing such poisoned examples is minimizing the gradients (the authors consider fine-tuning setup for feasibility of computations) and l_∞ distance between the original example and poisoned.

The Limitations of Adversarial Training and the Blind-Spot Attack
Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S. Dhillon, Cho-Jui Hsieh in ICLR, 2019 [523], *Attacks on Deep Learning Systems*
The authors analyze the effects of adversarial training, arguing that if the test distribution differs from the training distribution, adversarially trained models and also certified defenses can be ineffective, as the respective points can be in low-probability regions (i.e., distance of test image is far from training images - measured using a feature-embedding and k-nearest neighbor in case of individual points or KDE with KL-divergence in case of datasets). Based on this observation, the blind-spot attack is introduced. It is based on images that are far away from training data but still in-distribution and are recognized correctly by humans as well as networks. These blind-spots are shown to exist also on defended models, showing that AT does not scale to large datasets, similar to some other methods that certify just the training dataset. For attacks on simple datasets, rescaling and shifting pixels (for finding blind spots) and then doing a C&W attack on it (to find the adversarial image) is effective. This corresponds to adapting contrast or making background darker.

The Limitations of Deep Learning in Adversarial Settings
Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, Ananthram Swami in EuroS&P, 2016 [333], *Attacks on Deep Learning Systems*
The paper proposes an attack algorithm called JSMA which is based on Jacobian, saliency map and epsilon-perturbations. An implementation of the attack can be found in the cleverhans framework.

The Limitations of Federated Learning in Sybil Settings
Clement Fung, Chris J. M. Yoon, Ivan Beschastnikh in 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2020 [136], *Attacks on Deep Learning Systems*

The paper considers federated learning poisoning with the attack that is based on multiple sybils - the malicious (virtual) learners that act together. The proposed attack is termed training inflation. The proposed defense identifies sybils by the aligned gradient changes.

The Role of Sign and Direction of Gradient on the Performance of CNN
Akshay Agarwal, Richa Singh, Mayank Vatsa in CVPR Workshop, 2020 [4], *Attacks on Deep Learning Systems*
The paper analyzes the role of sign for gradient attacks (in FGSM) and proposes to not use the sign but rather rely on the gradient magnitude. This leads to the formulation of Fast Gradient Magnitude Attack (FGM), which in particular produces imperceptible perturbations(even for large epsilons) compared with using the sign. It is observed that higher values of epsilon are needed for this method to achieve a similar performance drop as for the FGSM method. Moreover, it is proposed to optimize images in the negative direction of the gradient for clean, misclassified examples as kind of defense, to get the images classified correctly and increase overall classification performance.

The trojAI Software Framework: An Open Source tool for Embedding Trojans into Deep Learning Models
Kiran Karra, Chace Ashcraft, Neil Fendley in arXiv, 2020 [213], *Attacks on Deep Learning Systems*
Software package for trojaning includes two modules, for generating data with various modifications and for generating model trained on this data.

Towards Evaluating the Robustness of Neural Networks
Nicholas Carlini, David Wagner in S&P, 2017 [62], *Attacks on Deep Learning Systems*
The whole set of possible replacements for loss functions to maximize and block constraints to follow in order to generate adversarial example (C&W - Carlini and Wagner- attack). Thorough evaluation and comparison to other attacks.

Towards poisoning of deep learning algorithms with back-gradient optimization.
Luis Munoz-Gonzales, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, Fabio Roli in ACM Workshop on Artificial Intelligence and Security, 2017 [303], *Attacks on Deep Learning Systems*
Since the bilevel optimization is a computationally intensive approximation, the authors propose replacing the inner optimization with an iterative approach of updating parameters such that the desired gradients in the outer problem are obtained from an incomplete optimization of the inner problem. In the limited knowledge attacks with surrogate models, the authors demonstrate that these poisoning attacks also transfer well to other nonlinear models.

TrISec: Training data-unaware imperceptible security attacks on deep neural networks
Faiq Khalid, Muhammad Abdullah Hanif, Semeen Rehman, Rehan Ahmed, Muhammad Shafiqu in IEEE 25th International Symposium on On-Line Testing and Robust System Design, 2019 [217], *Attacks on Deep Learning Systems*

The authors emphasize that the adversarial attacks should be constrained not only by l_p distance from the original image, but also other metrics should be added, which will allow to have more stealthy attacks.

Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples
Nicolas Papernot, Patrick McDaniel, Ian Goodfellow in arXiv, 2016 [331], *Attacks on Deep Learning Systems*
One of the first paper to use surrogate models for crafting attacks that are then transferred to the black box target model. In particular, they analyze transferability across models and propose a setup for designing surrogate models using reservoir sampling. They conduct experiments on two real-world classifiers that are deployed as ML as a service and successfully fool them.

Transferable Adversarial Attacks for Image and Video Object Detection
Xingxing Wei, Siyuan Liang, Ning Chen, Xiaochun Cao in IJCAI, 2019 [467], *Attacks on Deep Learning Systems*
The authors propose an approach for fast generation of adversarial examples for object detection networks, both based on regression and region proposal. The approach consists of training a GAN that will generate adversarial examples and the discriminator that has to distinguish between adversarial and clean input.

Transferable clean-label poisoning attacks on deep neural nets
Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, Tom Goldstein in ICML, 2019 [545], *Attacks on Deep Learning Systems*
This attack assumes that the adversary is able to collect samples from the victims training data, which can then be used for training a surrogate model and generating poisoned data via the convex hull approach. This paper demonstrates that convex hull outperforms the feature collision approach. Experiments are conducted both for a transfer learning as well as an end-to-end training scenario.

Trojan attacks on wireless signal classification with adversarial machine learning
Kemal Davaslioglu, Yalin E. Sagduyu in IEEE International Symposium on Dynamic Spectrum Access Networks, 2019 [102], *Attacks on Deep Learning Systems*
The paper proposes a way to construct triggers for backdoor poisoning of the DL models that are recognizing signals. The proposed trigger is a deformation of the wave, the experiments demonstrate the success of the poisoning.

Trojaning attack on neural networks.
Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, Xiangyu Zhang in NDSS, 2018 [261], *Attacks on Deep Learning Systems*
The paper presents a new backdoor attack, which does not require access to the training data of the model. Instead, the attack retrains the model with a small, crafted data set and inserts

the backdoor without harming the overall performance of the model. This attack follows three steps: At first, the adversary generates a trigger. This trigger tries to maximize the activations of certain neurons within the trained model. Then the adversary reverse engineers input data points for every output class of the model by tuning the pixels of an arbitrary starting image in a way that they maximize the respective class confidence level. Finally, the generated data points and the trigger are used to retrain the model, i.e. to induce causality between the trigger and the predefined target class. The attack is evaluated for natural language processing as well as computer vision tasks. The authors also discuss a potential defense approach, which is based on the hypothesis that the poisoned model has the tendency to assign data points to the target class of the backdoor attack.

Turning your weakness into a strength: Watermarking deep neural networks by backdooring
Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, Joseph Keshet in USENIX Security Symposium, 2018 [3], *Attacks on Deep Learning Systems*
Interpretation of the poisoning (backdoors) as a watermarking of a neural network for further copyright protection (when a neural network is provided as MLaaS).

UPSET and ANGRI: Breaking High Performance Image Classifiers
Sayantan Sarkar, Ankan Bansal, Upal Mahbub, Rama Chellappa in arXiv, 2017 [382], *Attacks on Deep Learning Systems*
Two targeted black-box attacks on image classifiers are presented. One is Universal Perturbations for Steering to Exact Targets (UPSET) and the other is Antagonistic Network for Generating Rogue Images (ANGRI). For UPSET, a residual generating network is employed to generate perturbations.

Understanding black-box predictions via influence functions
Pang Wie Koh, Percy Liang. in ICML, 2017 [220], *Attacks on Deep Learning Systems*
The paper uses the notion of influence functions in order to identify which training examples contributed most to the abilities of a model to predict. In particular, they propose to apply this notion, that is based on checking how the loss function is influenced by a modification of the training set, to the generation of the similar training examples, that will lead to the degraded accuracy of the model.

Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation
Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, Feiyue Huang in ICCV, 2019 [251], *Attacks on Deep Learning Systems*
A universal attack called Prior-driven Uncertainty Approximation (PD-UA) which is unsupervised and data-independent is presented. It is based on uncertainty estimation using MC Dropout to guide the perturbation (with the goal of increasing model uncertainty). The perturbations get initialized with a texture-bias (accounting for data-independent uncertainty).

Universal Adversarial Perturbations
Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard in CVPR, 2017 [296], *Attacks on Deep Learning Systems*
The authors show that it is possible to design a single adversarial perturbation (almost imperceptible to humans) that is image-agnostic (i.e. is applicable not to only one specific image and thus highly generalizable) and also transferable across classification networks in the sense that it leads to misclassification when added to most input images. The idea is to move individual data points across the decision boundary/decision regions and aggregate the perturbations needed to move the points. To have a sufficiently small perturbation in the end (although the goal is not to find THE smallest but rather A small one), the perturbation vector gets projected onto an l_p ball of desired radius. In general, using a couple of training set examples (less than the training set) is sufficient to compute a perturbation that is universal for the data distribution at hand. Since the optimization cannot be computed analytically, the authors make us of an effective approximation scheme. Moreover, the transferability of the universal perturbation across architectures is studied and seems to work well.

Universal Adversarial Perturbations: A Survey
Ashutosh Chaubey, Nikhil Agrawal, Kavya Barnwal, Keerat K. Guliani, Pramod Mehta in arXiv, 2020 [67], *Attacks on Deep Learning Systems*
Multiple techniques for generating universal perturbations, taxonomy of the approaches and defense mechanisms. Also considered not only image classification tasks, but also semantic segmentation and depth estimation.

Universal adversarial perturbations against semantic image segmentation
Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, Volker Fischer in ICCV, 2017 [286], *Attacks on Deep Learning Systems*
Semantic segmentation requires different ways for adversarial examples generation. The paper proposes static universal adversarial examples - when one noise turns the predictions on any image to one particular image prediction - and dynamic adversarial examples - when a particular class (pedestrians) is out of the networks abilities. The adversarial example is generated via minimizing the loss for the desired label over the whole training set.

Unravelling robustness of deep learning based face recognition against adversarial attacks
Gaurav Goswami, Nalini Ratha, Akshay Agarwal, Richa Singh, Mayank Vatsa in Proceedings of the AAAI Conference on Artificial Intelligence, 2018 [153], *Attacks on Deep Learning Systems*
The authors propose to use natural noise attacks on the face recognition systems. In particular they consider adding noise, adding grids and adding a beard to the face in the picture.

Unrestricted Adversarial Examples via Semantic Manipulation
Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, D. A. Forsyth in ICLR, 2020 [36], *Attacks on Deep Learning Systems*

The authors propose a method to generate photorealistic adversarial examples by unrestricted perturbations of color and texture. This is a targeted attack against classifiers and captioners which also sheds like on what information DNNs focus on. The authors argue that their method is effective against several defense methods, incl. adv. training, feature squeezing and JPEG defense, and more transferable than restricted attacks.

VENOMAVE: Clean-Label Poisoning Against Speech Recognition
Aghakhani Hojjat, Thorsten Eisenhofer, Lea Schonherr, Dorothea Kolossa, Thorsten Holz, Christopher Kruegel, Giovanni Vigna in arXiv, 2020 [173], *Attacks on Deep Learning Systems*
The Bullseye Polytope attack is transferred to the speech domain and used to attack an automatic speech recognition system which includes a neural network along with a hidden Markov model. The goal of the attack is to cause a misclassification such that specific activation phrases are falsely transcribed into the attackers desired commands, and the authors design the attack to be clean labels which makes them undetectable to humans. Specific words are chosen from certain frames of the time series as the base and then using the Bullseye Polytope method, this is used to craft poisoned words and then injected into the victims training set.

Wasserstein Adversarial Examples via Projected Sinkhorn Iterations
Eric Wong, Frank R. Schmidt, J. Zico Kolter in ICML, 2019 [473], *Attacks on Deep Learning Systems*
Attacks on image classifiers w.r.t. Wasserstein distance are introduced. The projection onto the Wasserstein ball is approxmated using modified Sinkhorn iterations. Adversarial training with PGD can defend from these attacks. In particular, the Wasserstein distance can be related to natural perturbation such as translation and rotation. The authors observe that building a certified defense based on the Wasserstein distance is not possible so far and would require new methods. However, models that are provably robust against l_∞ attacks perform better against Wasserstein-based attacks compared to standard training.

When does machine learning FAIL generalized transferability for evasion and poisoning attacks.
Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, Tudor Dumitras in USENIX Security Symposium, 2018 [420], *Attacks on Deep Learning Systems*
The paper proposes a framework FAIL that looks across 4 dimensions of knowledge of an attacker - features, algorithm, instances and leverage (how many features can be changed). This allows to estimate how strong an attack can be. They also propose a StngRay technique for generating poisoned examples in order to switch particular classes.

Wild patterns: Ten years after the rise of adversarial machine learning
Battista Biggio, Fabio Rolia in Pattern Recognition (journal), 2018 [37], *Attacks on Deep Learning Systems*
The survey contains accurate recollections of the field development starting from the adversarial machine learning before deep learning and up to 2018. One key observation summarized by the survey are the three golden rules of the proactive security cycle: (i) knowing the adversary

(ii) being proactive (iii) protecting the system. To this end, the authors strongly encourage the readers and the research community to focus on the design and implementation of proactive defense strategies. This includes the following three steps: (i) identification of relevant threats against the system under design and the simulation of corresponding attacks, (ii) implementation and use of suitable countermeasures (iii) repetition of this process before deployment to further validate the system.

Witches Brew: Industrial Scale Data Poisoning via Gradient Matching
Jonas Geiping, Liam Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, Tom Goldstein in arXiv, 2020 [142], *Attacks on Deep Learning Systems*
The authors present a technique of crafting poisons through gradient matching where the gradient of the poison sample is matched with that of the target sample. This is done by matching the gradients from the training loss and adversarial loss. Poison samples are created by changing the images with a small epsilon.

ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models
Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, Cho-Jui Hsieh in AISec, 2017 [75], *Attacks on Deep Learning Systems*
The authors present a zero-knowledge black-box attack where the gradients of the attacked models are not available. Opposed to other black-box attacks, the authors do not use a substitute model for which the gradients are available and adversarial examples can be generated using known techniques and finally transferred to the model under attack. The authors rather introduce new techniques with which the gradients of the attacked models are estimated and then used to directly generate adversarial examples without the error-prone step of transferring samples.

2.2.2 Certification and Verification Methods

A Dual Approach to Scalable Verification of Deep Networks
Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy Mann, Pushmeet Kohli in arXiv, 2018 [115], *Certification and Verification Methods*
In this paper, the authors introduce a dual Linear Programming (LP) formulation for verification of any feedforward neural network and activation function. They identify that the dual LP formulation created by Kolter and Wong uses a backpropagation-like calculation, which transforms it into a non-convex verification optimization. For their approach they construct a dual optimization problem that can be directly solved as an unconstrained convex optimization problem with a subgradient method. Additionally their approach is an anytime approach, meaning that their algorithm can be stopped at any time and returns valid bounds.

A Framework for Robustness Certification of Smoothed Classifiers using f-Divergences
Krishnamurthy (Dj) Dvijotham, Jamie Hayes, Borja Balle,J. Zico Kolter, Chongli Qin, Andras Gy-

orgy, Kai Xiao, Sven Gowal, Pushmeet Kohli in ICLR, 2020 [114], *Certification and Verification Methods*
The authors propose a framework for robustness certification of smoothed classifiers certify smoothed classifiers independent from the smoothing distribution. They achieve this by generalizing the adversarial problem and reducing it to a 2D-convex optimization problem using f-divergences. The choice of f-divergence is flexible as well. During their experiments, they apply this framework to not only the image but also the audio domain, which has not been explored much for robustness certification.

A Unified View of Piecewise Linear Neural Network Verification
Rudy Bunel, Ilker Turkaslan, Philip H.S. Torr, Pushmeet Kohli, M. Pawan Kumar in NeurIPS, 2018 [53], *Certification and Verification Methods*
The authors compare the existing complete methods Reluplex and Planet and generalize these two approaches. They identify different shortcomings and extend these by developing three improved algorithms: BaB-relusplit (BaB-relusplit), BaB-input and BaB-Smart Branching (BaBSB). Their algorithms mainly improved the runtime performance when compared to BaB, Reluplex and Planet.

A game-based approximate verification of deep neural networks with provable guarantees
Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska in Theoretical Computer Science, 2020 [476], *Certification and Verification Methods*
This paper focusses on maximum safe radius and the certification provides an absolute safety radius within which no adversarial example exists. The authors also show that by restricting perturbations to only certain features, it is possible to control the existence of adversarial examples within a relative safety radius. Under the Lipschitz continuity, only a small number of inputs need to be considered to derive a provable guarantee. The experiments show convergence of the upper and lower bounds.

AI^2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation
Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, Martin Vechev in S&P, 2018 [141]
AI^2 is an approach to certify defense mechanisms and to evaluate security specifications of (deep) NN. Abstract interpretation is utilized to convert different types of neural network layers (such as convolutional, fully-connected, ReLU and max pooling layers) into abstract transformers. An input to a NN application and all its possible perturbations are formed as an abstract element, which will be processed by these transformers. The resulting abstract element representing all possible outputs can be evaluated against certain security properties, e.g. robustness. The difficulty of this approach is to find an abstract interpretation of the NN layers as well as a suitable and precise representation of the input space.

Adversarial robustness via robust low rank representations
Pranjal Awasthi, Himanshu Jain, Ankit Singh Rawat, Aravindan Vijayaraghavan in NeurIPS, 2020

[19], *Certification and Verification Methods*
The authors propose an approach for robustness classification based on low rank representations for data. They use the $\infty \to 2$ matrix operator to translate l_2 norm robustness certificates into l_∞ robustness certificates. To calculate the robustness certificates they use an algorithm based on the multiplicative weight update method and semidefinite programming.

An Abstract Domain for Certifying Neural Networks
Gagandeep Singh, Timon Gehr, Markus Puschel, Martin Vechev in ACM, 2019 [405], *Certification and Verification Methods*
DeepPoly is a verification method designed for robustness verification. It uses the concept of abstract interpretations with a new abstract domain and new transformers for the key components (different activation functions) of a NN. Using abstract interpretation leads to the authors being able to certify robustness against complex adversarial threat models like rotation. It leads to sound and incomplete verification.

An Abstraction-Based Framework for Neural Network Verification
Yizhak Yisrael Elboher, Justin Gottschlich, Guy Katz in International Conference on Computer Aided Verification, 2020 [119], *Certification and Verification Methods*
In this paper, the authors utilize an abstraction of the entire network to a smaller easier to verify network using overapproximations for robustness verification. To ensure the soundness of the counterexamples identified by the verification method, they incorporate a counterexample-guided refinement to adjust the approximation of the NN. They developed this method in such a way that it can be easily incorporated into other verification frameworks, for example Marabou. For which they were able to increase the verification performance significantly.

An SMT-Based Approach for Verifying Binarized Neural Networks
Guy Amir, Haoze Wu, Clark Barrett, Guy Katz in arXiv, 2020 [10], *Certification and Verification Methods*
The authors propose a Satisfiability Modulo Theory based approach that can be used to verify binarized NNs, where some weights are binarized. Their approach adds a new deduction step and offer opportunities to parallelize the verification queries. They built it on-top of the Reluplex framework by introducing a sign function to encompass the binarized layers. Additionally, they use their approach to extend the Marabou framework, to verify binarized and non-binarized layers.

Beta-CROWN: Efficient Bound Propagation with Per-neuron Split Constraints for Complete and Incomplete Neural Network Verification
Shiqi Wang, Huang Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, J. Zico Kolter in arXiv, 2021 [461], *Certification and Verification Methods*
Beta-CROWN is a linear bound propagation method that can be used both for incomplete and complete certification of neural networks. To achieve complete verification it is combined with

the BaB framework. In this complete verification setting, the authors also make use of the effective branching strategy BaBSR for ReLU non-linearities. The verification method, which combines BaB, Beta-CROWN and BaBSR is then called Beta-CROWN BaBSR. This procedure returns incomplete bounds when terminated early, but when not interrupted it returns complete results. BaB usually has per-neuron splits (e.g. in ReLU at zero), which other combinations of BaB with incomplete methods cannot handle well. Beta-CROWN addresses this issue by introducing a new parameter called Beta. Beta represents the coefficients of a Lagrange-function that turns the constrained optimization problem (that has the per-neuron splits as a constraint) into a min-max problem. Beta can be optimized independently of the main verification task to the extent that when optimized exactly, the method is complete. However when only optimized partially, it will still remain sound. It is noteworthy that this min-max formulation shows the close connection to the dual problem, and in fact the authors also derive Beta-CROWN from the dual formulation.

Beyond the Single Neuron Convex Barrier for Neural Network Certification
Gagandeep Singh, Rupanshu Ganvir, Markus Puschel, Martin Vechev in NeurIPS, 2019 [403], *Certification and Verification Methods*
The authors propose the joint computation of optimal convex relaxation k-ReLU for multiple ReLU-neurons in one layer, to make the output bounding box tighter and the computed bounds more precise. They combine this approach with the DeepPoly certification method to kPoly by using the relaxations calculated by DeepPoly as the starting point for the k-ReLU relaxations. This results in a faster and tighter computation when compared to RefineZono and DeepPoly.

Black-Box Certification with Randomized Smoothing: A Functional Optimization Based Framework
Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, Qiang Liu in NeurIPS, 2020 [520], *Certification and Verification Methods*
The authors propose a randomized smoothing certification framework that uses non-Gaussian smoothing distributions. They developed a new family of distributions, as they found Gaussian distributions unable to address l_1, l_2 and l_∞-attacks properly. For each l_∞-norm, they designed a designated non-Gaussian distribution function. The proposed smoothing distributions were also designed to control the trade-off between the accuracy of the smoothed classifier and the robustness of the smoothing method. In comparison with Laplace smoothing for l_1-region certification and Gaussian smoothing l_2 and l_∞-region certification, their proposed framework showed a higher certification accuracy.

Black-box Certification and Learning under Adversarial Perturbations
Hassan Ashtiani, Vinayak Pathak, Ruth Urner in ICML, 2020 [13], *Certification and Verification Methods*
The authors introduce a formal model to describe black-box certification with bounds on the number of queries. They analyze the relation between the complexity of the robust learning problem and the query complexity in this setting.

Boosting Robustness Certification of neural networks
Gagandeep Singh, Timon Gehr, Markus Puschel, Martin Vechev in ICLR, 2019 [406], *Certification and Verification Methods*
In this paper, the authors propose RefineZono, a certifier that combines the concepts of incomplete and complete robustness certification for FNNs and CNNs with ReLU activations. While complete robustness certification methods give precise bounds for certification, they are not scalable for large networks. The proposed method combines the complete methods of MILP and the incomplete methods LP relaxation and abstract interpretations. In comparison to two state-of-the-art incomplete certifiers DeepZ and DeepPoly, RefineZone calculated more precise certification bounds.

Branch and Bound for Piecewise Linear Neural Network Verification
Rudy Bunel, Jingyue Lu, Ilker Turkaslan, Philip H.S. Torr, Pushmeet Kohli, M. Pawan Kumar in arXiv, 2019 [51], *Certification and Verification Methods*
In this paper the authors leverage the MILP formulation and develop based on this a BaB framework for certification. They use the branch and bound strategy to effectively branch on ReLU non-linearities. Further, their approach can be applied to CNNs as well as ReLU FNNs.

CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models
Shubham Sharma, Jette Henderson, Joydeep Ghosh in AIES, 2020 [394], *Certification and Verification Methods*
The authors of this paper propose the CERTIFAI framework for black-box robustness certification on any input domain. The framework calculates counterfactuals via a genetic algorithm based on the minimal distance to the original input data. Further, they define the robustness metric as a distance to between the resulting counterfactual samples and their input images. They extend their framework by allowing constraints to be added to features and their ranges, also enabling explainability and fairness assessments.

CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks
Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, Luca Daniel in AAAI, 2019 [39], *Certification and Verification Methods*
The authors propose a general certification framework that can certify different layer types (convolutional, pooling, fully-connected and batch normalization) and activation functions (tanh, sigmoid, ReLU and arctan). They achieve this by designing linear inequalities for each type of layer and activation function, which enables tighter bounds than other methods only computing bounds using linear inequalities for solely activation functions. The approach then combines and propagates the inequalities backwards for the entire model, resulting in a global upper and lower bound. The compare this method to FastLin, dual-LP and CROWN and achieve tighter bounds.

Certified Adversarial Robustness with Additive Noise
Bai Li, Changyou Chen, Wenlin Wang, Lawrence Carin in NeurIPS, 2019 [236], *Certification and Verification Methods*
The authors create a certification method based on the Renyi Divergence of a smoothed classifier. The method enables l_1 and l_2 robustness certificates by adding respectively Laplacian or Gaussian noise to the input and returning an upper bound on the tolerable strength of attacks. They also were able to establish a connection between the robustness of a classifier against adversarial perturbations and against additive noise. If a prediction is more accurate under additive Gaussian noise it leads to better overall robustness, including robustness against adversarial examples.

Certified Robustness to Adversarial Examples with Differential Privacy
Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Suman Jana in IEEE Symposium on Security and Privacy (SP), 2019 [230], *Certification and Verification Methods*
In this paper, the authors establish a connection between differential privacy and robustness against adversarial examples to create a new certification approach and a defense method for neural networks. Since differential privacy guarantees a bounded output change for small changes in a database, it goes well with the definition of adversarial robustness, where small changes of the input should not lead to large output changes of the model. Based on this connection they create a randomized pixel-based scoring function and use the expected value over the differential privacy noise to calculate the maximum attack size tolerated by the model.

Certifying Geometric Robustness of Neural Networks
Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, Martin Vechev in NeurIPS, 2020 [26], *Certification and Verification Methods*
With the proposed algorithm DeepG, the authors address the lack of certification methods for geometrically transformed images. DeepG uses a combination of a Lipschitz optimization algorithm and linear programming to compute linear constraints bounding the set of geometrically transformed images that are correctly classified by the model.

DNNV: A Framework for Deep Neural Network Verification
David Shriver, Sebastian Elbaum, Matthew B. Dwyer in arXiv, 2021 [400], *Certification and Verification Methods*
This paper addresses central problems of verification method researchers and developers, in particular the lack of standardized specification and neural network formats. Due to the lack of standardized formats, It is often challenging to run benchmark experiments for the comparison of different verifiers. This paper presents the Deep Neural Network Verification (DNNV) framework, which helps to create verification benchmarks by standardizing the network and specification format, plus it increases the applicability of existing verification methods to richer properties by performing reductions on the specification and simplifying operations of the neural network. For the network standard, DNNV relies on the Open Neural Network Exchange (ONNX) format. For formulating the desired properties / specifications, the authors develop a

new Pythen-embedded domain-specific language. DNNV translates and simplifies the neural network and properties to the input format of the desired verifier, and finally executes the verification method on the given verification task. The output of the verifier is also handed to the user in a standardized format. The authors provide an open source tool, which contains an implementation of DNNV supporting 13 verification methods.

DeepSplit: Scalable Verification of Deep Neural Networks via Operator Splitting
Shaoru Chen, Eric Wong, J. Zico Kolter, Mahyar Fazlyab in arXiv, 2021 [77], *Certification and Verification Methods*
DeepSplit is a verification method that addresses the issue of convex relaxation, making it sound but incomplete. It uses an operator splitting method that can solve the convex relaxation issue exactly. To that end, it uses artificial decision variables that split the problem into subproblems which can sometimes be solved analytically. The authors utilize a splitting technique called Alternating Direction Method of Multipliers (ADMM) that solves the Lagrangian relaxation. ADMM has several advantages (like scalability, exploitation of sparsity, parallelization) which according to the authors translate to the verification procedure. The method is applicable to standard network architectures.

Differentiable Abstract Interpretation for Provably Robust Neural Networks
Matthew Mirman, Timon Gehr, Martin Vechev in ICML, 2018 [290], *Certification and Verification Methods*
The work shows an implementation and improvement of the AI^2 method for provable robustness called DIFFAI. Code implementing the approach is available.

Efficient Certification of Spatial Robustness
Anian Ruoss, Maximilian Baader, Mislav Balunovic, Martiv Vechev in AAAI, 2020 [369], *Certification and Verification Methods*
This paper introduces robustness certification against spatial adversarial attacks, a.k.a. vector field attacks. To achieve it, they propose convex relaxations that take into account vector field transformations. The certification method is not specific to any model type and can be applied quite generally. It is worth noting that certain kinds of spatial robustness like rotation and translation are special cases of the vector field robustness. To achieve such generality, the authors constrain the vector fields to have a maximum pixel displacement (using the so-called T_p-norm) as well as a certain degree of smoothness (within neighboring pixels). These constraints lead to an optimization problem which can be solved using linear programming.

Efficient Formal Safety Analysis of Neural Networks
Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, Suman Jana in NeurIPS, 2018 [459], *Certification and Verification Methods*
The Neurify framework is based on linear relaxations to find the optimal bounds for robustness certification of ReLU FNNs. It uses a combination of interval analysis and linear relaxations to track relaxed dependencies via the interval propagation. In addition, they introduce directed

constraint refinement to reduce the relaxation error and identify overestimated nodes during the optimization progress.

Efficient Neural Network Robustness Certification with General Activation Functions
Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, Luca Daniel in NIPS, 2018 [527], *Certification and Verification Methods*
The paper introduces CROWN, a generic analysis framework for certifying neural networks using linear or quadratic upper and lower bounds for general activation functions that are not necessarily piecewise linear. Layer for layer the neural networks activation functions, respectively their corresponding weights, are replaced with linear bounds. This starts on the lowest layer and is then proceeded upwards just until the input layer.This results in non-trivial boundaries for the whole NN function, a lower and an upper bound. The lower bound representing minimum distortion can be certified as the largest possible lower bound for the considered input data point.

Efficient Neural Network Verification with Exactness Characterization
Krishnamurthy (Dj) Dvijotham, Robert Stanforth, Sven Gowal, Chongli Qin, Soham De, Pushmeet Kohli in UAI, 2020 [116], *Certification and Verification Methods*
In this paper, the authors present a novel approach combining Lagrangian relaxations and semidefinite programming. They identify that the quadratic constraints describing the certification problem definition can be efficiently relaxed by developing a convex relaxation on them using Lagrangian relaxations. Using the resulting convex relaxation, they introduce PGD-SDP, an efficient algorithm using Projected Gradient Descent to solve the relaxed semidefinite program.

Efficient Verification of ReLU-based Neural Networks via Dependency Analysis
Elena Botoeva, Panagiotis Kouvaros, Jan Kronqvist, Alessio Lomuscio, Ruth Misener in AAAI, 2020 [42], *Certification and Verification Methods*
Venus is a verification method that is based on BaB and only works for ReLU FNNs. It aims at reducing the configuration space generated in BaB by removing redundancies from the problem. This prunes the search tree that is considered in a MILP formulation of the verification problem. As a means of achieving this, the authors introduce and use the notion of a dependency relation between two nodes (neurons). A node *A* is dependent on a node *B*, if whenever *B* is strictly active or inactive, so is *A*. This allows the number of configurations to be reduced by a factor of 1/4 whenever a dependency is true. This leads to a faster, complete certification method.

Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming
Sumanth Dathathri, Krishnamurthy Dvijotham, Alexey Kurakin, Aditi Raghunathan, Jonathan Uesato, Rudy Bunel, Shreya Shankar in arXiv, 2020 [101], *Certification and Verification Methods*
The authors propose a dual semidefinite programming formulation for the robustness verification problem. They show that the dual semidefinite programming is the maximum eigenvalue problem with interval bound constraints, which can be applied to any quadratically-constrained

program like the adversarial robustness specification. Using a subgradient algorithm, they were able to calculate the robustness using only a constant number of forward and backward passes through the network for each iteration. By utilizing the dual semidefinite programming, they were able to minimize the memory requirements from $O(n^4)$ for regular semidefinite programming relaxation to $O(n)$.

Ensuring Dataset Quality for Machine Learning Certification
Sylvaine Picard, Camille Chapdelaine, Cyril Cappi, Laurent Gardes, Eric Jenn, Baptiste Lefevre, Thomas Soumarmon in International Workshop on Software Certification (WoSoCer), 2020 [341], *Certification and Verification Methods*
The authors summarize existing data quality standards and identify their limitations towards the machine learning domain. Based on those limitations they form a new workflow to create certifiable datasets consisting of three documents. The Dataset Definition Standard (DDS) contains general recommendations for datasets while addressing different properties of data quality relevant to the machine-learning domain e.g. reliability or the data annotation process. In the Dataset Requirements Standard (DRS) the recommendations from the DDS are applied to the specific use case and state the requirements regarding the validity, completeness, representativeness and innocuity of the data. Lastly, the Dataset Verification Plan (DVP) defines a plan of action to verify the compliance of the dataset with its specification. In this paper, the authors are able to convey the importance of data quality for machine learning systems and create a workflow aiding the certification of such systems.

Evaluating Robustness of Neural Networks with Mixed Integer Programming
Vincent Tjeng, Kai Y. Xiao, Russ Tedrake in ICLR, 2018 [436], *Certification and Verification Methods*
This paper introduces the use of mixed linear integer programming in combination with a presolve algorithm to minimize the number of variables to be optimized for neural network robustness certification. The method itself is designed for feedforward neural networks with piecewise activation functions but can be applied to convolutional and residual layers as well.

FROWN: Thightened Neural Network Robustness Certificates
Zhaoyang Lyu, Ching-Yun Ko, Zhifeng Kong, Ngai Wong, Dahu Lin, Luca Daniel in AAAI, 2019 [270], *Certification and Verification Methods*
FROWN / Fastened-CROWN is a direct extension of CROWN. It has the same underlying affine bounding framework that is used in CROWN. This means that it is an incomplete and sound verification method that finds affine upper and lower bounds on the output of the neural network which hold on entire input regions. The main difference lies in the optimization of the slopes and intercepts used in the activation bounds.

Fast Geometric Projections for Local Robustness Certification
Aymeric Fromherz, Klas Leino, Matt Fredrikson, Bryan Parno, Corina Pasareanu in ICLR , 2021 [134], *Certification and Verification Methods*

The authors propose a new efficient approach for local robustness certification Fast Geometric Projections (FGP) based on simple geometric projections and activation patterns. This framework is limited to piecewise linear activation functions, as they use this property to partition the input space into regions in which the networks behavior is linear. They define those regions as activation regions in which the activation patterns for the neurons are the same. They define linear inequalities called activation constraints, whose intersections define an activation region, based on the models gradients. Their approach computes, if there exists a projection of the original input that is within an epsilon-ball around the original input for all activation regions that are within an epsilon distance from the original input. Notably, they identified that this approach performs well for l_2 robustness certification, which is neglected by most other certification methods, while also supporting l_∞ robustness certification.

Fast and Effective Robustness Certification
Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Puschel, Martin Vechev in NeurIPS, 2018 [404], *Certification and Verification Methods*
This paper proposes a new certifier, DeepZ, supporting a broader spectrum of machine learning architectures, activation functions and floating point arithmetic. The approach is based on pointwise Zonotope abstract transformers. Zonotope abstract transformers have been shown effective in other certifiers to evaluate robustness properties for all possible perturbations of an input for ReLU activation functions. The proposed pointwise approach extends these transformers to Sigmoid and Tanh activations. In addition, the certifier is applicable to FNNs, CNNs and RNNs. In comparison to the state-of-the-art methods AI^2 and Fast-Lin, the certifier has been shown to be more precise and scalable.

Fast and Precise Certification of Transformers
Gregory Bonaert, Dimitar I. Dimitrov, Maximilian Baader, Martin Vechev in ACM SIGPLAN International Conference on Programming Language Design and Implementation, 2021 [38], *Certification and Verification Methods*
Two threat models are considered for text classification. One threat model is where an adversary can add a l_p-noise to the embedding of an input sequence. In the second threat model, the attacker is able to exchange every word in the input sequence with a synonym. The Multinorm Zonotope, an extension of the classical Zonotope domain, contains new noise symbols bounded by an l_p-norm, which improves certification against l_p-norm bound attacks. Abstract transformers are defined for all operations in the Transformer network,including affine operations, ReLU, tanh, exponential, reciprocal, dot product and softmax. This results in a Multinorm Zonotope representing an over approximation of the possible outputs of the Transformer network. Thus, robustness can be certified if the lower bound is positive.

Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation
Matthias Hein, Maksym Andriushchenko in NeurIPS, 2017 [167], *Certification and Verification Methods*

This is the first paper proposing the use of Cross Lipschitz regularization to calculate a lower bound as a formal guarantee for the classifier and an upper bound as the change needed for adversarial manipulation. They generate the adversarial examples box constrained. They provide a local and global version of their technique and show that the local constant leads to lower bounds that are up to 8 times better as for the global version. Additionally, they address that for Lipschitz methods there is still a lot of room for improvement on the tightness of the bounds.

Formal Security Analysis of Neural Networks using Symbolic Intervals
Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, Suman Jana in USENIX, 2018 [460], *Certification and Verification Methods*
The paper presents the formal verification method ReluVal for feedforward neural networks with ReLU activation functions. ReluVal provides formal guarantees by using interval arithmetic instead of SMT solvers, which can be slow and inflexible. ReluVal propagates the desired security properties through each layer, computes the output ranges and the algorithm terminates by either providing a formal guarantee or by identifying an adversarial example. Due to the strong dependency between layers of a neural network, interval arithmetics come with the risk of extreme overestimation of the output range. To overcome this issue, the authors propose two strategies: First of all, they use symbolic interval representations to consider at least simple linear dependencies between layers. Furthermore, they leverage the Lipschitz continuity of deep neural networks by applying iterative refinements, i.e. they apply the interval propagation algorithm to sub-intervals of the relevant input interval and in this way decrease the size of the resulting output interval.

Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks
Ruediger Ehlers in arXiv, 2017 [118], *Certification and Verification Methods*
The approach introduced in this paper is based on a linear approximation of the network itself, which is then combined with SMT solvers. In addition, an algorithm based on regular Satisfiability (SAT) solvers is presented that employs this approximation to infer the non-linear neurons states.

Formal verification of neural network controlled autonomous systems
Xiaowu Sun, Haitham Khedr, Yasser Shoukr in ACM International Conference on Hybrid Systems: Computation and Control, 2019 [422], *Certification and Verification Methods*
In this paper, the authors claim that existing verification frameworks often consider unrealistic robustness specifications, and that most methods are barely applicable to relevant safety and reliablility statements for cyber-physical systems. Thus, they present a formal verification technique for an autonomous robot moving in a simple two-dimensional workspace, where it tries to avoid objects as well as the boundary of the environment. The authors attempt to compute the set of safe initial states for the autonomous robot equipped with a NN controller that processes LiDAR images, such that its trajectory starting from these initial states is guaranteed to avoid the given obstacles. This is done by constructing a finite state abstraction of the system and by using standard reachability analysis over the finite state abstraction to compute the set

of safe initial states. The reachability analysis is executed with the help of a Satisfiability Modulo Convex (SMC) solver.

Lagrangian decomposition for neural network verification
Rudy Bunel, Alessandro De Palma, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip Torr, M. Pawan Kumar in Conference on Uncertainty in Artificial Intelligence, 2020 [52], *Certification and Verification Methods*
State-of-the-art verification methods view the verification task as a constrained, non-convex optimization problem. In order to solve this problem, the contraints linked to the nonlinear activation functions of the neural network are replaced by convex hull relaxations. Furthermore, a simple Lagrangian relaxation is often applied to make use of the weak duality theorem. In this paper, the authors suggest a more efficient formulation of the dual problem by using a Lagrangian Decomposition technique. Here, the given constraints are assigned to subsets of constraints, where each subset is then equipped with its own copy of variables. The authors are able to prove that their new class of optimization problems generally yields bounds at least as strong as those obtained through Lagrangian relaxation. The presented optimization problems admit efficient optimization methods, in particular supergradient ascent and the use of proximal methods are discussed in the paper. They show that their approach is easily parallelizable independent of the different suggested optimization methods.

Maximum Resilience of Artificial Neural Networks
Chih-Hong Cheng, Georg Nuhrenberg, Harald Ruess in ATVA, 2017 [81], *Certification and Verification Methods*
The approach introduced in this paper uses mixed integer programming based optimization to identify the upper bound on the tolerated perturbations. The authors were able to significantly speed up solving the mixed integer programming formulation by formulating heuristic problem encoding, by introducing a dataflow analysis to generate small big-M formulation and running the BaB in parallel. The dataflow analysis utilizes interval arithmetic to obtain relatively small values for big-M. The heuristic problem encodings include looking at multiple layers to obtain smaller big-Ms, prioritizing during the branching and generating constraints from the samples and the solver initialization.

Optimization and abstraction: a synergistic approach for analyzing neural network robustness
Greg Anderson, Shankara Pailoor, Isil Dillig, Swarat Chaudhuri in ACM SIGPLAN, 2019 [11], *Certification and Verification Methods*
In this paper, a method called Charon is proposed, which improves on the AI^2 method. Charon combines abstract interpretations and the gradient based counterexample search for neural network verification, utilizing the information gathered while searching for the counterexamples to guide the abstract interpretations. Additionally, they develop a verification policy, which is learned during a training phase, to choose the abstract domain for the abstract interpretation and how to partition the input into two subregions.

POPQORN: Quantifying Robustness of Recurrent Neural Networks
Ching-Yun Ko, Zhaoyang Lyu, Tsui-Wei Weng, Luca Daniel, Ngai Wong, Dahua Lin in ICML, 2019 [219], *Certification and Verification Methods*
POPQORN is an extension of CROWN to recurrent architectures like vanilla RNNs and LSTM models. It uses the underlying affine bounding framework introduced in CROWN. This means that it is an incomplete and sound verification method that finds affine upper and lower bounds on the output of the RNN. It is restricted to single-cell RNNs however.

PRIMA: Precise and General Neural NetworkCertification via Multi-Neuron Convex Relaxations
Mark Niklas Muller, Gleb Makarchuk, Gagandeep Singh, Markus Puschel, Martin Vechev in arXiv, 2021 [302], *Certification and Verification Methods*
PRIMA (Precise multi-neuron abstraction) is a method that verifies any specification of neural networks that can be described using polyhedra. The neural networks do not have any constraint regarding their activation function. As most incomplete certification methods rely on finding a convex relaxation of the optimization constraints, this paper addresses issues regarding precision an computational complexity. Many methods use single neuron-wise relaxations, which leads to significant imprecision, because it neglects any dependency between neuron. Former methods address this by using multi-neuron convex approximations. However, this leads to solving several instances of the convex hull problem, which is NP-hard. This paper improves the previous methods by developing a new approach to approximation of the convex hull problem, which they call Split-Bound-Lift. PRIMA is therefore an incomplete certification method.

PROVEN: Verifying Robustness of Neural Networks with a Probabilistic Approach
Tsui-Wei Weng, Pin-Yu Chen, Lam M. Nguyen, Mark S. Squillante, Ivan Oseledets, Luca Daniel in ICML, 2019 [468], *Certification and Verification Methods*
In this paper, the probabilistic framework Proven for robustness certification is proposed. The approach extend the common worst-case certification formulation from other papers such as CNN-Cert and Fast-Lin to a probabilistic setting. The worst-case certification problem is defined as a minimization problem of the lower bound for the margin between network-outputs, where the lower bound is a linear function. They assume that the set of adversarial perturbations follows a probability distribution within an l_p-norm ball around the original input and show this enables using the probability of the margin function to identify certified lower bounds for the network. During their experiments, they show that their method is applicable to different activation functions, probability distribution and is scalable to larger CNN models.

Provable Certificates for Adversarial Examples: Fitting a Ball in the Union of Polytopes
Matt Jordan, Justin Lewis, Alexandros G. Dimakis in NeurIPS, 2019 [206], *Certification and Verification Methods*
GeoCert is a framework that utilizes local geometrical information to compute the exact pointwise robustness of a network. The authors show that piecewise linear neural networks partition

the input space into polyhedral complices and additionally that those have boundary decompositions, which can be computed efficiently. To improve the computational effort of GeoCert, they leverage the lower bounds computed by Lipschitz overestimation as a starting point for their algorithm.

Provable Robustness of ReLU networks via Maximization of Linear Regions
Francesco Croce, Maksym Andriushchenko, Matthias Hein in AISTATS, 2020 [93], *Certification and Verification Methods*
The authors present a certification approach of ReLU-based networks similar to the known Reluplex algorithm. The update is based on linearity maximization of the classifier.

Randomized Smoothing of All Shapes and Sizes
Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya Razenshteyn, Jerry Li in ICML, 2020 [502], *Certification and Verification Methods*
The authors present two methods to compute a robustness certificate for different smoothing distributions and l_p-norms. The first method leverages level sets given by the Wulff Crystal norm for the three norms l_1, l_2 and l_∞, which enables an exact computation of the growth function for spherical distributions. Secondly, they introduce a differential method that calculates an upper bound for the derivative of the growth function, for the l_1 and l_∞ norms and different classes of distribution functions. In addition, they identify the shortcomings of randomized smoothing based methods for different l_p-norms.

Reachability analysis of deep neural networks with provable guarantees
Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska in arXiv, 2018 [368], *Certification and Verification Methods*
In this paper, the authors introduce a new approach to verifying reachability of neural networks concatenated with Lipschitz functions. The approach is not restricted to ReLUs and can handle quite general layers like sigmoid and max-pooling. The proposed method is applicable to all feedforward deep neural networks. The only assumption made by the authors is that a Lipschitz constant of the network is known. To that end, they prove the Lipschitz continuity of certain components. Given this assumption, they utilize methods from global optimization based on adaptive nested optimization to (asymptotically) find global minima. In this method, the neural network is not transformed into linear constraints, as most other papers do. Because of their approach, the verification problem is independent of the size of the network and still (asymptotically) sound and complete. The authors also stress that by solving this reachability problem, several other specifications like adversarial example generation and output range analysis can be solved. Alongside the paper, the authors provide the software tool DeepGO, which contains the proposed verfication method.

Recent Advances in Understanding Adversarial Robustness of Deep Neural Networks
Tao Bai, Jinqi Luo, Jun Zhao in arXiv, 2021 [23], *Certification and Verification Methods*

The mentioned paper provides an survey about different aspects of adversarial robustness in terms of certification aspects like benchmarks, underlying correlations within training data and the dilemma in increasing the robustness.

RecurJac: An Efficient Recursive Algorithm for Bounding Jacobian Matrix of Neural Networks and Its Applications
Huan Zhang, Pengchuan Zhang, Cho-Jui Hsieh in AAAI, 2019 [529], *Certification and Verification Methods*
RecurJac is a proposed recursive algorithm to efficiently calculate the Jacobian matrix of a neural network in polynomial time. By retrieving the Jacobian matrix this method is able to compute the local or global Lipschitz constant and characterize the local optimization landscape. Additionally, by improving on the upper bound of the Jacobian matrix, the algorithm can also calculate the robustness verification for the entire network.

ReluDiff: Differential Verification of Deep Neural Networks
Brandon Paulsen, Jingbo Wang, Chao Wang in ICSE, 2020 [336], *Certification and Verification Methods*
The authors propose a new approach called ReLUDiff to verify feedforward neural networks with ReLU activations. It is a differential verification technique, which focuses on the comparison of two closely related neural networks. It consists an approximate forward interval analysis step, which calculates via affine and ReLU transformation the input and output intervals as well as the corresponding interval differences between related neurons in the networks. To improve the accuracy of this approximation, the authors introduce the backward refinement step to split the input region into smaller sub-regions based on the level of influence on the output difference. To determine the influence on the output difference, the gradient difference of the two networks is calculated. Lastly, in comparison to ReLUVal and DeepPoly, ReLUDiff outperformed them in regards to accuracy and computational speed on different network structures and three different datasets.

Reluplex: An efficient SMT solver for verifying deep neural networks
Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, Mykel J. Kochenderfer in International Conference on Computer Aided Verification, 2017 [214], *Certification and Verification Methods*
This paper presents verification that is both sound and complete (as proven in the appendix) using the SMT approach. It is restricted to NNs with ReLU activations. Reluplex extends the existing SMT solvers by extending the simplex algorithm to be able to handle the ReLU activations (Reluplex with simplex). This approach allows the Reluplex algorithm to verify all kinds of specifications and in particular the robustness specification. Additionally, within this framework, the authors prove that NN verification is NP-complete, a contribution for which they are often cited.

Robustness Certification with Generative Models
Matthew Mirman, Alexander Hagele, Pavol Bielik, Timon Gehr, Martin Vechev in ACM SIGPLAN

International Conference on Programming Language Design and Implementation, 2021 [291], *Certification and Verification Methods*
This method is shown to compute tight deterministic guaranteed bounds on probabilities of outputs given distributions over inputs, and demonstrates the verification of visible specifications based on latent space interpolations of a generator. The threat model considered is a classification network which could change its prediction when presented with images of a head from different angles, produced by interpolating encodings in the latent space of an autoencoder.

Safety Verification and Robustness Analysis of Neural Networks via Quadratic Constraints and Semidefinite Programming
Mahyar Fazlyab, Manfred Morari, George J. Pappas in arXiv, 2020 [129], *Certification and Verification Methods*
In this paper the authors combine quadratic constraints and semidefinite programming to identify a certified bound for the network output.They abstract nonlinear activation functions by defining quadratic constraints that encode their different properties such as monotonicity, bounded slope, bounded values and repetition across layers. Based on those quadratic constraints they derived a Linear Matrix Inequality (LMI) feasibility problem for the entire model, asserting that the input set is enclosed by a safe describing a safety or robustness property with a certified upper bound. A semidefinite program can solve this LMI by minimizing the upper bound through over approximation of the reachable set of outputs. They compare their method to MILP, semidefinite relaxation and LP relaxation and found their method to be able to identify tighter bounds more efficiently than the compared methods.

Safety Verification of Deep Neural Networks
Xiaowei Huang, Marta Kwiatkowska, Sen Wang, Min Wu in CAV, 2017 [191], *Certification and Verification Methods*
In this work the authors propose a framework based on SMT that certifies for the l_1 and l_2-norm. The framework creates small manipulations on the input images by exhaustively searching the region around the original image, through so-called ladders. The ladders are created by exhaustively branching and iterating over successive manipulations created for each layer of the model. This method definitely returns an adversarial example if the certification verdict is not robust. They provide an implementation, which utilizes the Z3 SMT solver. They compare their certification framework to two methods used to find adversarial examples: the FGSM and the Jacobian saliency map algorithm. Their framework performed slower or similar to the compared methods, but provided adversarial examples with lesser l_1 and l_2-distance.

Scalable Polyhedral Verification of Recurrent Neural Networks
Wonryong Ryou, Jiayu Chen, Mislav Balunovic, Gagandeep Singh, Andrei Dan, Martin Vechev in CAV, 2020 [370], *Certification and Verification Methods*
Prover is a verifier specific to recurrent architectures like vanilla and LSTM RNNs. It provides several abstractions for the activations used in these model architectures. In particular, they address the common two-dimensional activation sigmoid times tanh of LSTM models, which has

its own problems. They find these abstractions using sampling and optimization while maintaining the soundness of the algorithm (i.e., it is not probabilistic even though sampling is involved). In order for the certification task to be solved as effectively as possible, they further use gradient-based optimization of hyperparameters intrinsic to the certification method to find optimal relaxations of the activation functions.

Scaling Polyhedral Neural Network Verification on GPUs
Christoph Muller, Francois Serre, Gagandeep Singh, Markus Puschel, Martin Vechev in Proceedings of Machine Learning and Systems3, 2021 [301], *Certification and Verification Methods*
This papers presents a design of sound polyhedra algorithms for GPUs which enables verification of large NNs. The robustness of a 1M neuron, 34-layer deep residual network was proved in 34.5 ms. The DeepPoly algorithm is parallelized to enable fast verification. The algorithm presented in Fast and effective robustness certification.(Singh et al., 2018) is made parallelizable and CUDA enabled.

Scaling the Convex Barrier with Active Sets
Alessandro De Palma, Harkirat Behl, Rudy R. Bunel, Philip Torr, M. Pawan Kumar in ICLR, 2020 [320], *Certification and Verification Methods*
This paper addresses central problems of state-of-the-art relaxation-based formal verification methods. Current approaches often utilize rather loose bounds for the ReLU activation functions, e.g. the Planet relaxation. The missing tightness implies that specifications can often not be verified. Therefore, the authors introduce a primal optimization problem with a tight ReLU relaxation, which comes with the cost of exponentially many constraints. In a second step, the dual problem of the tighter relaxation problem is considered. Here, the authors face the problem of exponentially many dual variables. To overcome this issue, a new dual solver is presented (Active Set Solver), which restricts the set of active variables. Overall, this strategy provides a speed accuracy trade-off where a larger computational capability can provide tighter bounds. The presented verification method is only applicable to feedforward neural networks with ReLU activations.

Scaling the Convex Barrier with Sparse Dual Algorithms
Alessandro De Palma, Harkirat Singh Behl, Rudy Bunel, Philip HS Torr, M. Pawan Kumar in arXiv, 2021 [321], *Certification and Verification Methods*
In this paper the authors present two sparse dual solvers for linear relaxations. They present a dual initializer called Big-M, that calculates an active set of dual variables dynamically, which creates the input for both dual solvers. The first dual solver is a subgradient solver called Active Set to overcome the convex barrier. The second dual solver is called Saddle Point and utilizes a Frank Wolfe style optimizer.

Semidefinite relaxations for certifying robustness to adversarial examples
Aditi Raghunathan, Jacob Steinhardt, Percy Liang in NeurIPS, 2018 [351], *Certification and Verification Methods*

The authors propose semidefinite programming to solve approximate the non-linear ReLU activation function. They achieve this by defining linear and quadratic constraints for the non-linear ReLU activation function and relaxing them to a semidefinite program. They compared their approach to regular relaxation certification and the Gradient Certification method and achieved a tighter upper bound on the worst-case loss

SoK: Certified Robustness for Deep Neural Networks
Linyi Li, Xiangyu Qi, Tao Xie, Bo Li in arXiv, 2020 [240], *Certification and Verification Methods*
This paper is a survey paper giving an overview over and comparing different robustness techniques in their performance and efficiency for different epsilon-values and different model sizes. In order to compare the methods they implemented the methods without provided code themselves.

Specification-guided safety verification for feedforward neural networks
Weiming Xiang, Hoang-Dung Tran, Taylor T. Johnson in arXiv, 2018 [479], *Certification and Verification Methods*
This paper provides a verification method for FNNss, which is based on layer-by-layer interval propagation. In order to approximate the resulting output intervals, the authors make use of the Lipschitz continuity of standard activation functions. Furthermore, they recognize that the quality of their interval analysis approach is highly affected by the size of the input interval. Thus, a subdivision of the input interval is proposed, where the subdivision algorithm is inspired by the Moore-Skelboe algorithm.

Statistical Verification of Neural Networks
Stefan Webb, Tom Rainforth, Yee Whye Teh, M. Pawan Kumar in arXiv, 2018 [465], *Certification and Verification Methods*
The authors propose a method that is extends an already existing Monte Carlo approach called amlş to fit robustness verification. The Adaptive Multi-Level Splitting (AMLS) algorithm estimates the probability of rare events, which is used in this paper to estimate the probability that a property is violated under an input distribution model (violation probability). This approach provides not only information that a model is robust or non-robust, but also how robust the model is.

The Convex Relaxation Barrier, Revisited: Tightened Single-Neuron Relaxations for Neural Network Verification
Christian Tjandraatmadja, Ross Anderson, Joey Huchette, Will Ma, Krunal Patel, Juan Pablo Vielma in NeurIPS, 2020 [435], *Certification and Verification Methods*
In this paper, two verification algorithms OPTC2V and FastC2V are introduced. OPTC2V is a LP-based method that improves on the commonly used Δ-LP relaxation method by dynamically generating bounding inequalities from a newly introduced family of inequalities, based on the pre-activation and post-activation variables for the ReLU-neuron. The FastC2V is a propagation method, that dynamically changes the an initial set of bounding functions by computing the

bounding problem in a backward pass and then decides the change of inequality for each neuron after the full solution for the network is computed in a forward pass. They compared their algorithms to an optimized implementation of DeepPoly, the common Δ-LP relaxation, kPoly and RefineZono and achieved a larger amount of verified images.

The Marabou Framework for Verification and Analysis of Deep Neural Networks
Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah in International Conference on Computer Aided Verification, 2019 [215], *Certification and Verification Methods*
The Marabou framework is based on SMT and builds upon the Reluplex algorithm. The framework itself can transform queries about the networks properties by transforming them into constraint satisfaction problems. It extends Reluplex by supporting different piecewise-linear activation functions and improving on the overall performance of Reluplex.

Tight Certificates of Adversarial Robustness for Randomly Smoothed Classifiers
Guang-He Lee, Yang Yuan, Shiyu Chang, Tommi S. Jaakkola in NeurIPS, 2020 [231], *Certification and Verification Methods*
The authors extend the definition of certifying randomly smoothed classifier with additive isotropic Gaussian noise to fit alternative noise distributions. In addition, they define a discrete distribution for l_0 robustness certification and compare it to the standard isotropic Gaussian distribution. They also show that this approach is applicable to different data domains (image and molecule), as well as different architectures (Deep Neural Network (DNN) and decision trees).

Towards fast computation of certified robustness for relu networks
Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, Luca Daniel in ICML, 2018 [469], *Certification and Verification Methods*
In this paper, two certification algorithms for ReLU FNNs are introduced. While FastLin utilizes linear approximations, FastLip computes the upper bound on the local Lipschitz constant to calculate the certified lower bound on the minimum perturbation for a NN. Experiments showed that both methods outperformed both LP, Lipschitz and formal verification methods, such as Reluplex.

Towards safety verification of direct perception neural networks
Chih-Hong Cheng, Chung-Hao Huang, Thomas Brunner, Vahid Hashemi in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020 [80], *Certification and Verification Methods*
The authors develop a verification workflow, which addresses the specification and scalability problem of computer vision verification. The workflow is then used to verify various properties of a perception network used by Audi to determine waypoints and orientation for self-driving cars. The formal specifications are generated with the help of input property characterizer networks. These are binary classefiers trained on close-to-output representations of the training

data. Scalability limitations are avoided by concentrating on the last layers of the direct perception network. For this, it has to be assumed that for every possible input data point in the Operational Design Domain (ODD), the computed neuron activation pattern of a certain close-to-output layer is contained in a polyhedron. The edges of the polyhedron are calculated by analyzing the layer activations on the training data. This also implies that the verification workflow is limited to statistical guarantees.

Verification of RNN-Based Neural Agent-Environment Systems
Michael E. Akintunde, Andreea Kevorchian, Alessio Lomuscio, Edoardo Pirovano in AAAI, 2019 [7], *Certification and Verification Methods*
In this paper, the authors discuss the formal verification of RNN-based systems. They focus on reachability and robustness verification of ReLU-based vanilla RNNs. To the best of their knowledge, they are the first to consider RNN verification. To solve the verification task, they introduce a method called unrolling, where a RNN is transformed into a FNN by specifying certain weights of the FNN. Further they show that the FNN and RNN represent the same function. Based on this, it is easy to verify the RNN - simply by verifying the FNN. The authors discuss complete and sound verification using MILP formulations, however in theory other verification procedures could be applicable.

Verifying Recurrent Neural Networks using Invariant Inference
Yuval Jacoby, Clark Barrett, Guy Katz in International Symposium on Automated Technology for Verification and Analysis, 2020 [197], *Certification and Verification Methods*
In this paper, the authors use overapproximation to reduce RNNs to FNNs, which reduces the verification complexity. Then the authors use already existing verification methods to verify the reduced FNN.

Verifying probabilistic specifications with functional lagrangians
Leonard Berrada, Sumanth Dathathri, Robert Stanforth, Rudy Bunel, Jonathan Uesato, Sven Gowal, M. Pawan Kumar in arXiv, 2021 [32], *Certification and Verification Methods*
The authors propose a framework that uses functional Lagrange multipliers to verify probabilistic NNs. Firstly, they define the stochastic verification problem for probabilistic NNs and identify two problems with the traditional Lagrangian relaxation of the verification problem. They address these problems by defining functional Langrangians and show that the functional Lagrangian Dual can be used to solve the stochastic optimization problem. Additionally, they show that their framework can be applied to prove different properties such as adversarial robustness and out-of-distribution detection.

l1 Adversarial Robustness Certificates: a Randomized Smoothing Approach
Jiaye Teng, Guang-He Lee, Yang Yuan in ICLR, 2020 [430], *Certification and Verification Methods*
The authors introduce a method for robustness certification via randomized smoothing for the asymmetric l_1-norm, which is often neglected by other randomized smoothing methods. To ad-

dress the asymmetry they employ isotropic Laplace distributions for smoothing and combining differential privacy and the Neyman-Pearson method for randomized smoothing.

2.2.3 Defense Methods

A Direct Approach to Robust Deep Learning Using Adversarial Networks
Huaxia Wang, Chun-Nam Yu in ICLR, 2019 [456], *Defense Methods*
Adversarial training using a generative network is proposed. Concretely, the generator learns to produce additive noise that is put on top of the image (which is given to the generator as input). Interestingly, the discriminator network is the final (robust) classifier that is trained jointly with the generator (and not an intermediate step to ensure indistinguishability of the original and perturbed images). Regularization for the discriminator aims at stability in training for both networks.

On Adaptive Attacks to Adversarial Example Defenses
Florian Tramer, Nicholas Carlini, Wieland Brendel, Aleksander Madry in NeurIPS, 2020 [438], *Defense Methods*
The authors present the methodology that should help tune adaptive attacks correctly. In particular, they show that defenses from top conferences can be broken when using the correct evaluation setup with adaptive attacks. The authors say say that other papers use adaptive attacks that broke previous defenses but in fact, automating the attacks is not possible and there need to be adaptations.

Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey
Samuel Henrique Silva, Peyman Najafirad in arXiv, 2020 [402], *Attacks on Deep Learning Systems*
The survey provides a taxonomy for defenses and a detailed list of the prominent adversarial attacks, adversarial training, certified and regularized defenses. Regarding defenses the authors divide the methods into three categories: (1) Gradient Obfuscation/Masking (2) Robust Optimization (3) Adversarial Example Detection. In the survey the authors focus on the second category of defenses which include the following approaches: Adversarial Training, Bayesian Approach, Certified Defenses, and Regularization Approaches.

Systematic evaluation of backdoor data poisoning attacks on image classifiers
Loc Truong, Chace Jones, Brian Hutchinson, Andrew August, Brenda Praggastis, Robert Jasper, Nicole Nichols, Aaron Tuor in CVPR, 2020 [444], *Attacks on Deep Learning Systems*
The paper presents an extensive experimental study on the effects of architecture and regularization decisions of the developer on the success of various trigger-based backdoor attacks. The experiments are conducted on two different datasets, namely Flowers and CIFAR-10. The authors then derive four key results, which should guide developers in their design decisions. In particular, the experimental results suggest that certain architectures (e.g. NasNet-Mobile) and regularization methods (e.g. SNNL) can significantly decrease the vulnerability of a classifier to

backdoor attacks. Furthermore, it is claimed that a short retraining with clean data can be used as an effective method for removing existent backdoors.

A new defense against adversarial images: Turning a weakness into a strength
Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, Kilian Q. Weinberger in NeurIPS, 2019 [182], *Defense Methods*
The authors present an adversarial example detection method based on two threshold operations. The first threshold tests if the target models predictions are robust to Gaussian noise. This is achieved by adding noise to the inputs and measuring the distance between the respective outputs and outputs triggered by the original samples. The second threshold operation enforces that samples are still close to the decision boundary when running an adversarial attack. Hence, during inference, the method internally runs a PGD attack with the current sample and quantifies the distance to the decision boundary with the required attack steps to successfully alter the classification output. In a later study presented by Tramer et al. [438], the defense method was show to contain errors in the internally used PGD implementation. Furthermore, the evaluation of the adaptive attacks could further be improved by Tramer et al. [438] fully circumventing the detection approach.

ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation
Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, Xiangyu Zhang in CCS, 2019 [260], *Defense Methods*
The authors present a backdoor detection method called Artificial Brain Stimulation (ABS). The method is based on the intuition that contaminated neurons connected to backdoor triggers exhibit special and distinctive behavior when being stimulated with different inputs. ABS leverages this intuition and analyzes single neurons in the NN for different potentially artificial inputs. Subsequently the outputs of the neurons are analyzed. The authors observe that for contaminated neurons the output activation is often elevated in hence distinguishable from the activations of benign neurons. In the second step, the potentially contaminated neurons are used to reverse-engineer possible backdoor triggers. In the final step, these triggers are then tested. If the reconstructed backdoor triggers lead to misclassifications, the NN is assumed to be trojaned. For a successful application of ABS, several assumptions need to be fulfilled: (I) there is only one trigger for each target class. (II) observing one neuron at the time is sufficient to detect backdoor triggers. (III) the backdoor triggers are classified as the chosen target class with a high probability.

APE-GAN: Adversarial Perturbation Elimination with GAN
Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, Yongdong Zhang in ICASSP, 2019 [205], *Defense Methods*
The authors present APE-GAN, a GAN-based defense method to protect against adversarial examples. APE-GAN is closely related to the autoencoder-based defense method called MagNet. Here, a GAN is used to preprocess inputs prior to the classification. The goal is to use the pretrained GAN and map currently processes inputs onto the benign data manifold. With this pro-

cess, adversarial perturbations are aimed to be removed, such that the inputs can again be classified currently by the NN to protect.

ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity
Kangkang Lu, Cuong Manh Nguyen, Xun Xu, Kiran Chari, Yu Jing Goh, Chuan-Sheng Foo in ICLR, 2021 [268], *Defense Methods*
A method that combines ensemble learning (diversifying the models via regularization) and semi-supervised learning and can be added to AT is proposed. Concretely, the network's output on non-target classes is diversified and the unlabeled samples where the network predictions match are treated as labeled ones for the current mini-batch.

Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning
NhatHai Phan, Xintao Wu, Han Hu, Dejing Dou in ICDM, 2017 [339], *Defense Methods*
The authors present the adaptive Laplace mechanism (AdLM) which tries to guarantee differential privacy for deep NNs. For this purpose the authors perturb affine transformations of the neurons and loss functions used in the models. Furthermore, to improve their approach, the noise is added adaptively based on the contribution of each output to the results. More noise is added to features which are less important to the model and vice versa to preserve the performance of the NN. To this end, the authors are able to employ their approach in a wide range of NNs, since the required privacy budget is independent of the number of training epochs. Hence, AdLM can be used in complex scenarios where higher numbers of training steps are required. Still, for complex learning tasks, the method may have negative impacts on the accuracy of the resulting models.

Adversarial Defense by Stratified Convolutional Sparse Coding
Bo Sun, Nian-Hsuan Tsai, Fangchen Liu, Ronald Yu, Hao Su in CVPR, 2019 [421], *Defense Methods*
The authors present a defense method which tries to preprocess input images such that adversarial perturbations are filtered out and the samples can again be classified correctly. For this purpose, the authors introduce a novel sparse transformation layer (STL) prior to the input layer of the target NN. This layer projects samples into a stratified low- dimensional quasi-natural sample such that they can be classified correctly. This process is performed during the training and testing phase. In their paper, the authors state that their approach is mainly designed to protect in black-box and grey-box settings. In white-box settings in which the attacker is aware of the defense method the proposed approach is vulnerable to adaptive attacks. To show this, the authors perform BPDA attacks since parts of the preprocessing pipeline contain non-differentiable calculations. In summary, the proposed method is partly based on obfuscated gradients and cannot protect against white-box adaptive attacks.

Adversarial Detection and Correction by Matching Prediction Distributions
Giovanni Vacanti, Arnaud Van Looveren in arXiv, 2020 [451], *Defense Methods*
The paper presents an autoencoder-based adversarial defense method, which can be used for detection and reconstruction of adversarial input data. In the past, several autoencoder defenses

have been proposed (e.g. MagNet). However, these approaches do not consider the output of the classifier while training the autoencoder. In contrast, the proposed autoencoder is trained with a Kullback-Leibler divergence loss function, which compares the output distributions of the classifier on the original image and the output image of the autoencoder. The authors claim that this custom loss function allows the autoencoder to make use of significantly different areas of the input space, in particular areas with different decision boundary shapes. The adversarial perturbation therefore fails to transfer to these new areas of the input.

Adversarial Distributional Training for Robust Deep Learning
Yinpeng Dong, Zhijie Deng, Tianyu Pang, Hang Su, Jun Zhu in NeurIPS, 2020 [107], *Defense Methods*
Adversarial distributional training (ADT) is proposed. Its key idea is to circumvent the need to produce all possible specific attacks (for good coverage of the attack space) to train a robust model. So the approach considers the adversarial distribution around a data point in the inner maximization instead of single adversarial examples (the threat model remaining the same, i.e., l_∞ here). To avoid distribution collapse into a Dirac distribution in one point, a regularization term that includes entropy is used. The adversarial distributions are then parametrized with parameters (i.e., mean and std.). These parameters can be trained. One can use the low-variance reparameterization trick and Monte Carlo sampling to estimate the expectation. Alternatively, a generator can be used to learn the parameters or an implicit distribution.

Adversarial Example Defense: Ensembles of Weak Defenses are not Strong
Warren He, James Wei, Xinyun Chen, Nicholas Carlini, Dawn Song in 11th USENIX Workshop on Offensive Technologies, 2017 [165], *Defense Methods*
In this paper the authors investigate the question whether ensembles of adversarial defenses improve the robustness of the protected NNs. For this purpose, the authors investigate three different ensemble defenses. The first two sets (feature squeezing and specialist) are designed to work as ensembles themselves while the final set consists of the simultaneous use of three independent detection methods (Feinman et al. [130], Gong et al. [151], Metzen et al. [285]). To bypass the ensemble defenses, the authors use the C&W attack method and adaptively apply it to the different set of defenses. The authors find that all three combinations can be bypassed 100% of times either with less perturbations or a slight increase of required perturbations which is still human imperceptible. Hence, the authors conclude that ensembles which can be independently bypassed do not increase the level of robustness of NNs. Furthermore, the authors finds that adversarial examples can be transferred between independently trained detectors.

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods
Nicholas Carlini, David Wagner in AISec, 2017 [61], *Defense Methods*
The paper revisits 10 already published defense methods against adversarial examples. By performing adaptive attacks, the authors are able to bypass all attack methods and succesfully create adversarial examples for the protected neural networks.

Adversarial Examples: Attacks and Defenses for Deep Learning
Xiaoyong Yuan, Pan He, Qile Zhu, Xiaolin Li in IEEE Transactions on Neural Networks and Learning Systems (Journal), 2019 [516], *Defense Methods*
This survey which was published in 2019 provides a good overview of state-of-the-art research in adversarial machine learning. Due to high rate of newly published papers in this research area, the authors only considered methods published until November 2017. Hence, some new contributions are missing in this survey. Together with a good overview and taxonomy of attacks and defenses, the survey discuses some further points worth mentioning. (1) Why do adversarial examples exist (2) Why do adversarial examples transfer (3) How to create an environment in which neural networks and defense methods can be evaluated and their robustness be quantified. Finally the authors show that the majority of research focuses on the image domain and at the time of publication no study discusses the proposed attack or defense method in a more general scheme unrelated to the underlying domain.

Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses against Attacks
David J. Miller, Zhen Xiang, George Kesidis in Proceedings of the IEEE (Journal), 2021 [287], *Defense Methods*
The authors present a brief overview of some findings in the field of adversarial machine learning including test-time-evasion attacks and defenses as well as data poisoning attacks and attempts of reverse engineering of deployed models. Even though the authors provide an interesting overview, some important findings known in the field are missing or only introduced briefly. Still, some important notes on assumptions generally referred to in the field are worth mentioning. The authors discuss the limitation in numerous publications that attackers are assumed to know the ground truth label of attacked samples. This assumption or precondition cannot be guaranteed in real-world setting. Furthermore, the authors discuss the relationship between defenses based on the detection of attacks and required perturbations induced by attackers. Oftentimes, researchers assume that increasing the attackers effort (i.e., the allowed perturbation budget) leads to monotonically increasing attack success. The authors argue that this may not be the case, since detectors should be able to detect adversarial examples with greater distortions more easily compared to slightly perturbed examples. To this end, the authors also show arguments why the evaluation of adaptive white-box attacks may be unfair from the viewpoint of defenses.

Adversarial Logit Pairing
Harini Kannan, Alexey Kurakin, Ian Goodfellow in arXiv, 2018 [211], *Defense Methods*
Logit pairing for natural images and their adversarial example counterpart is performed. It consists of encouraging similar logits for pairs of images (giving information that these images are close and thus should be classified identically).

Adversarial Robustness Against the Union of Multiple Perturbation Models
Pratyush Maini, Eric Wong, J. Zico Kolter in ICML, 2020 [278], *Defense Methods*

A generalization of the PGD adversarial training (called MSD, Multi steepest descent) is introduced. It combines different perturbation models and is supposed to achieve robustness against the union of different perturbation attacks. This paper build on the work of Tramer et al. [437] in the sense that it also considers robustness to multiple perturbations simultaneously. However, the authors claim that their work improves on the previously mentioned one and is more efficient/successful in the sense of better convergence properties. In particular, it aggregated the individual adversaries into one, leveraging the joint knowledge about adversarial regions.

Adversarial Robustness through Local Linearization
Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, Pushmeet Kohli in NeurIPS, 2019 [346], *Defense Methods*
An approach for training robust models which builds on locally linearizing the loss around training data (and thus preventing gradient obfuscation) is presented. This is done with a regularizer term called local linearity regularizer using the Taylor expansion. The idea behind this approach is that with increasing iterations of PGD attacks during AT, the loss becomes increasingly linear around training points - thus linearity seems to correlate with robustness. It is experimentally shown that models trained with this method are robust against strong as well as weak attacks.

Adversarial Training Methods for Semi-Supervised Text Classification
Takeru Miyato, Andrew M. Dai, Ian Goodfellow in ICLR, 2017 [292], *Defense Methods*
The authors propose to adapt AT and virtual AT for recurrent neural networks in the text domain (sentiment and topic classification). The perturbations are produced on the word embeddings (and not on one-hot input vectors) to allow for infinitesimal and continuous changes. This method can be used semi-supervised and supervised.

Adversarial Training against Location-Optimized Adversarial Patches
Sukrut Rao, David Stutz, Bernt Schiele in ECCV Workshops, 2020 [357], *Attacks on Deep Learning Systems*
The authors propose a way to generate untargeted, image-specific adversarial patches and optimize also their locations. Adversarial training based on these strategies is introduced to robustify models. The patch attack is based on LaVAN (Karmon et al. [212]). To optimize the patch location, the patch is moved by a certain amount of pixels. If the loss does not increase, the location is not updated. The updates of the patch content as well as the location are conducted in every iteration. During adversarial training, adversarial patches are produced for half of the batch.

Adversarial Training for Free
Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, Tom Goldstein in NeurIPS, 2019 [388], *Defense Methods*
This paper introduces a setup for adversarial training that is much faster than usual adversarial training since it uses one backward pass for gradient computation for both model parameter update and image perturbations. That means that the gradient of the loss w.r.t. the image and

w.r.t. the model parameters is computed in the same pass. Since no multi-step adversarial update is possible in such a setup, the authors propose to train on the same mini-batch several times in a row (then dividing the number of epochs by m). The perturbation generated on a given mini-batch is used as initialization for the perturbation in the next mini-batch. The adversarial for free trained models exhibit similar accuracies as the naturally trained counterparts and the overhead compared to normal training is neglectable.

Adversarial Vertex Mixup: Toward Better Adversarially Robust Generalization
Saehyung Lee, Hyungyu Lee, Sungroh Yoon in CVPR, 2020 [233], *Defense Methods*
The authors address the problem of adversarial feature overfitting and propose soft-labeling (label smoothing function), in particular in combination with Adversarial Vertex mixup (AVmixup), a data augmentation approach that interpolates between existing datapoints, as solution. The label smoothing function gives weight lambda to the true class and uniformly distributes among the remaining classes, taking a linear combination of two label-smoothings together. The adversarial mixup selects datapoints between real examples and scales adversarial examples. The authors experimentally show that their approach can be successfully combined with feature scattering.

Adversarially Robust Generalization Requires More Data
Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, Aleksander Madry in NeurIPS, 2018 [383], *Defense Methods*
The authors observe that adversarial training can lead to overfitting (in particular on the adv. examples) and that more data is beneficial for robust generalization (larger capacity needed).

Adversarially robust transfer learning
Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, Tom Goldstein in ICLR, 2020 [390], *Defense Methods*
The authors address the problem of producing robust models in a transfer learning setting. In particular, they observe that using the feature extractor or the robust (source domain) model leads to also a robust (target domain) model. So, they recommend fine-tuning the last layers for the target domain and leaving the robust feature extraction part. Another possibility explored by the authors is fine-tuning the whole model on the target domain. On itself, it leads to forgetting of the robust features. But it can successfully be combined with methods from lifelong learning to prevent this.

Are Labels Required for Improving Adversarial Robustness
Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, Pushmeet Kohli in NeurIPS, 2019 [449], *Defense Methods*
An approach for unsupervised adversarial training (UAT - actually semi-supervised) is introduced. It is claimed to be competitive in robustness outcome with supervised training and is motivated by the observation that robust models require much larger labeled datasets for robust generalization due to increased sample complexity (Schmidt et al. [383]). The approach

requires some labeled examples which can then be extended by unlabeled ones. The key idea of the first proposed strategy is to decompose the adversarial risk into classification and smoothness loss (the latter does not require labeled examples and is expressed as KL-divergence). The second strategy uses a base classifier trained on the labeled data. This base classifier then produces pseudo-labels for the unlabeled data. Regular AT can then be performed on the resulting pseudo-labeled data set which enforces smoothness around the labeled examples.

Attacks Which Do Not Kill Training Make Adversarial Learning Stronger
Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, Mohan Kankanhalli in ICML, 2020 [531], *Defense Methods*
The authors propose friendly adversarial training (FAT) that employs not the worst but the least adversarial data (friendly one) that minimizes the loss (compared to the adversarial data given) and is misclassified. This can be achieved with early stopping of PGD as soon as the adv. example is first misclassified. The approach is based on the observation that strong adversarial examples may already cross decision boundaries and thus negatively influence training as it makes it difficult to fit both natural and adversarial examples (cross-over mixture).

Bag of Tricks for Adversarial Training
Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, Jun Zhu in to appear in ICML, 2021 [328], *Defense Methods*
A review of several implementations (20) of AT methods on CIFAR10 concerning their setup, training details, hyperparameters, etc. is conducted. By evaluating a set of training tricks, the authors find that the wrong setup choices can decrease the model robustness quite strongly. Current defenses are found to be implemented differently, making it difficult to benchmark them and attribute the successes/failures to the method itself or rather the setup. The authors propose an effective training setup for the case of CIFAR10.

Barrage of Random Transforms for Adversarially Robust Defense
Edward Raff, Jared Sylvester, Steven Forsyth, Mark McLean in CVPR, 2019 [350], *Defense Methods*
The authors combine multiple preprocessing transformations into a defense method. Among these are popular previous defense methods that have been broken, e.g. based on image compression - however, when combined in a random cascade, they become effective countermeasures. The paper profits from a carefully optimized attack model and an elaborate evaluation. Recent findings like obfuscated gradients or the robustness under EoT attacks were considered. Although some follow-up studies suggest that the transformations must be carefully selected, we believe this paper gives a good overview about transformation-based defenses in the Image combined with a sound evaluation.

Beware the Black-Box: on the Robustness of Recent Defenses to Adversarial Examples
Kaleel Mahmood, Deniz Gurevin, Marten van Dijk, Phuong Ha Nguyen in arXiv, 2020 [277], *Defense Methods*

This survey advocates for adaptive black-box attacks when evaluating the performance of defense methods. In the same style as popular white-box attack papers (e.g. Tramer et al. [438]), the authors analyze the adversarial of nine common adversarial defenses against black-box attacks. As black-box attacks are only based on input-output relations and the training data, these attacks are more generally applicable. Their evaluation shows that most methods only increase black-box robustness by less than 25%. The survey provides an extensive empirical analysis, yet lacks more theoretical insight.

Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks
Jianyu Wang, Haichao Zhang in ICCV, 2019 [457], *Defense Methods*
Bilateral Adversarial Training (BAT), which consists of training with perturbed images (single-step targeted attack using PGD with random starts) as well as labels (heuristic approach, similar to label smoothing for the case of equal gradients wrt. label in non-ground truth classes), is presented. The goal is to have a low loss and a small magnitude of the gradient (locally flat loss surface, making it more difficult to produce adv. examples), as the latter is observed to be linked with robust models.

Boosting Adversarial Training with Hypersphere Embedding
Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, Hang Su in NeurIPS, 2020 [329], *Defense Methods*
This paper describes a way to improve existing AT approaches with methods from representation learning (Hypersphere embedding for the features: feature normalization, weight normalization and angular margins). The hypersphere embedding is applied for adversarial attack generation and model parameter updates, leading to more efficient adv. example computation due to improved update directions.

Breaking Transferability of Adversarial Samples with Randomness
Yan Zhou, Murat Kantarcioglu, Bowei Xi in arXiv, 2018 [544], *Defense Methods*
The authors try to improve the adversarial robustness by introducing a pool of NNs, each trained on the same data set. An attacker has full access to one of these NNs, but cannot attack the others. Predications are made based on a randomly selected NN. In their evaluation, the authors show that an adversarial example generated on the known model may not transfer to the randomly selected NN if it was modified with enough randomness. This randomness may come from random weight initializations or additive Gaussian noise. Unfortunately, the authors do not question their threat model by suitable adaptive attacks, nor discuss the impacts on accuracy. Thus, the advantages of the method are rather vague.

Cascade Adversarial Machine Learning Regularized with a Unified Embedding
Taesik Na, Jong Hwan Ko, Saibal Mukhopadhyay in ICLR, 2018 [307], *Defense Methods*
The authors propose cascade adversarial training, a method that aims at improving robustness against unknown iterative attacks. For the cascade training, also adv. examples for already

trained (defended) networks are employed (iterative FGSM - using the knowledge of already defended nets) as well as one-step attacks during the training of the respective network itself. Moreover, a regularization is applied during AT, which enforces low distance between clean and corresponding adversarial examples. The resulting defended networks exhibits better robustness to iterative attacks but worse on one-step attacks.

Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality
Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, James Bailey in ICLR, 2018 [273], *Defense Methods*
The authors present a method to detect adversarial examples based on theLocal Intrinsic Dimensionality (LID) metric. This metric measures the distance from an input to its neighbors and thus allows the detection of out-of-distribution adversarial examples. Even though the authors state that their method is not intended to be a defense method, their evaluation contains experiments with adaptive attacks in which the authors try to show the robustness of the LID-based adversarial example detection. The approach is evaluated in more detail by Athalye et al. [16]. Here, Athalye et al. report thatLID cannot detecthigh confidence adversarial examples (even in the grey-box setting). Hence, no further evaluation and experimental setup of the adaptive attacks was necessary to bypass the defense method.

Characterizing audio adversarial examples using temporal dependency
Zhuolin Yang, Bo Li, Pin-Yu Chen, Dawn Song in ICLR, 2019 [508], *Defense Methods*
This defense detects adversarial examples created to attack automatic speech recognition systems. In order to detect adversarial audio files, the authors split the waveform into two parts. The first part, as well as the complete file are processed by the targeted NN and a transcription for both is generated. During detection, the transcription of the first part and the first part of the complete transcription are compared to each other and the distance between the produced sentences is calculated. If the distances surpasses a predefined threshold, the sample is marked as adversarial. Even though the paper extensively evaluates adaptive adversaries, Tramer et al. [438] find in a later study that the proposed detection approach can be bypassed without significantly higher perturbation budgets. Tramer et al. achieve this by redefining the attack loss function which is optimized using gradient descent to perform targeted attacks.

Combatting Adversarial Attacks through Denoising and Dimensionality Reduction: A Cascaded Autoencoder Approach
Rajeev Sahay, Rehana Mahfuz, Aly El Gamal in CISS, 2019 [373], *Defense Methods*
The authors try to minimize the success of adversarial attacks by filtering the adversarial noise. They do so by adding an denoising autoencoder in front of the original classifier. Autoencoders are networks reconstructing the input itself - in this case it was trained on reconstructing the input from perturbed samples. According to their evaluation, the method shows increased robustness. However, the threat model does not consider adaptive attackers, which may easily break this defense by incorporating the autoencoder. Moreover, we believe that the autoencoder may not easily generalize to all input samples and may severely affect the general model

performance.

Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks
David Stutz,Matthias Hein,Bernt Schiele in ICML, 2020 [417], *Defense Methods*
The authors propose confidence calibration during adversarial training (CCAT) to enforce a quick confidence decay and uniformly low confidence outside of the epsilon radius of l_∞ attacks that are used for training (aiming at correct one-hot distribution on correct examples and uniform distributions on corrupted examples). With that, the model becomes (more) robust even to unseen adversarial examples while having better accuracy on uncorrupted inputs.

Countering adversarial images using input transformations
Chuan Guo, Mayank Rana, Moustapha Cisse, Laurens van der Maaten in ICLR, 2018 [157], *Defense Methods*
The authors present a defense method which is based on input preprocessing during training and test time to break the adversarial characteristics of the processes inputs. This preprocessing is done in a random manner which is shown to be more effective compared to deterministic approaches and includes the following pre-processing steps: bit-depth reduction, JPEG-compression, total variance minimization, and image quilting. The transformations are performed either during training or at test time. The training is performed on cropped and rescaled images. At test time, foreach image the classifier randomly samples 30 crops of the input, rescales them, and averages the model predictions over all crops. In later publications and evaluations of this defense method, it was shown that the approach provides a reasonable level of robustness in the case of grey-box and black-box settings. Still, during white-box attacks based on the EoT approach, it was shown that the defense can be bypassed.

CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy
Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, John Wernsing in ICML, 2016 [147], *Defense Methods*
The authors present their homomorphic encryption method for NNs called CryptoNets. The approach combines the use of homomorphic encryption (HE) and NNs which allows the classification of encrypted data. This preserves the privacy of the user in machine-learning-as-a-service environments. To enable the use of this encryption scheme, some changes to the NNs are necessary. This includes the replacement of the activation functions by polynomial activation functions and the use of scaled mean pooling instead of max pooling. To further increase the efficiency of the encryption, the authors propose the use of leveled HE instead of fully HE. Even though the authors achieve good results for the MNIST data set, the approach cannot be applied to real-world NNs as the training process becomes computational expensive and sometimes unstable for deeper models.

Curriculum Adversarial Training
Qi-Zhi Cai, Min Du, Chang Liu, Dawn Song in IJCAI, 2018 [54], *Defense Methods*

An adversarial training approach using curriculum learning is presented. It is supposed to be more effective than standard PGD adversarial training as proposed by Madry. The basic idea is to use an increasing attack strength (defined by the curriculum, reflected in the number of iteration steps for the attack) to prevent overfitting on the strong examples at early stages of training (and thus enable generalization).

DLA: Dense-Layer-Analysis for Adversarial Example Detection
Philip Sperl, Ching-Yu Kao, Peng Chen, Xiao Lei, Konstantin Bottinger in EuroS&P, 2020 [415], *Defense Methods*
The authors present an adversarial example detection method based on the analysis of the dense layer neuron activations called DLA. In their paper, the authors observed that the activation patterns of NNs differ depending on the nature of the current input. Adversarial examples processed by the NNs trigger distinctive patterns in the activation space making the detection possible. Based on this observation, the authors trained a secondary NN, called alarm model, which observes the dense layer activations of the original target NN. The alarm model performs the detection of the adversarial examples by automatically analyzing the activations of the target model triggered by either benign or malicious inputs. During the evaluation of DLA, the authors showed the applicability of the method in the image, audio, and text domain. In a later study, the method was shown to be broken when adaptively attacking the complete system using orthogonal PGD.

Deep Defense: Training DNNs with Improved Adversarial Robustness
Ziang Yan, Yiwen Guo, Changshui Zhang in NeurIPS, 2018 [499], *Defense Methods*
A regularization approach is introduced that is supposed to enhance robustness. It incorporates a heuristic form of adversarial perturbation and distinguishes between correctly and incorrectly classified examples.

Deep Learning with Differential Privacy
Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, Li Zhang in CCS, 2016 [1], *Defense Methods*
The authors present the differentially private stochastic gradient descent (DP-SGD) algorithm which can be used during the training of DL models. With this optimizer, the gradients of the NNs are randomly perturbed allowing a private training process decreasing the chance of information leakage. Additionally, gradient clipping is applied to bound the gradient norm which is usually necessary during training. The main drawback of the approach is the fact, that it yet cannot be applied to complex deep NNs.

Deep k-NN Defense Against Clean-Label Data Poisoning Attacks
Neehar Peri, Neal Gupta, W. Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, John P. Dickerson in ECCV, 2020 [337], *Defense Methods*
The authors present a new and intuitive defense method against clean-label data poisoning attacks. This class of attacks tries to provoke the NN to misclassify a particular target test sample

during runtime. The defense method is based on the observation that the inner representation of poisoned inputs often lie closer to the distribution of the target class compared to the distribution of the benign data with the same label. Therefore, poison samples are typically surrounded by samples of the target class.This is due to the nature of the attack in which the poison samples are visually similar to the benign data of the same class but are intended to be classified differently by the attacked NN. The authors leverage this observation and the intuitive k-nearest neighbors method using the inner representation of the samples to detect poison samples before training.

DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks
Huili Chen, Cheng Fu, Jishen Zhao, Farinaz Koushanfar in IJCAI, 2019 [71], *Defense Methods*
The authors present a defense method protecting NNs against trojan/poisoning/backdoor attacks called DeepInspect. As the authors improve the previously shown defense method called Neural Cleanse, both methods share the initial hypothesis: By performing inference with slightly perturbed inputs and observing the changes in the classification outputs, backdoors can be detected. This is due to the observation that backdoored NNs change their classification output upon smaller changes to the input compared to trustworthy NNs. Subsequently, the detected backdoors are identified to finally remove them and again allow a secure operation of the NN. In their paper, the authors improve Neural Cleanse in the following three aspects: As the authors train a conditional GAN which estimates theprobability density function of potential triggers for any target class of the data set, all classes are scanned at once. In Neural Cleanse, the detection was only possible for one class at a time. Furthermore, DeepInspect does not require white-box access to the NN and operates in a black-box setting as well. Finally, since the authors first perform a model inversion to extract training data and create a substitute data set, the method does not require original data samples, increasing the applicability of DeepInspect.

Defending Against Adversarial Attacks by Randomized Diversification
Olga Taran, Shideh Rezaeifar, Taras Holotyak, Slava Voloshynovskiy in CVPR, 2019 [429], *Defense Methods*
The authors present a general defense framework, where multiple classifiers are trained on permutated version of the training data. These permutations are determined by a key, which is used during training and testing. An aggregation method combines the output of all classifiers. In their evaluation, the authors show that more classifiers in parallel result in less successful attacks. As example, the authors used the discrete Fourier transform (DCT) as permutation with random sign flips. Generally, the evaluation is not extensive enough to judge if the framework is applicable to other use cases than image classification. However, if a suitable permutation can be found for the respective data type, the very same ideas should be applicable.

Defending Against Neural Network Model Stealing Attacks Using Deceptive Perturbations
Taesung Lee, Benjamin Edwards, Ian Molloy, Dong Su in IEEE Symposium on Security and Privacy Workshops, 2019 [234], *Defense Methods*

The authors present a defense method to protect against model extraction attacks. As the majority of model extraction techniques relies on theprediction probabilities of NNs analyzed during mulitple queries, the authors base their defense on perturbing these outputted prediction probabilities. This is achieved by perturbing the final output activation layer. In a later publication by Juuti et al. [208], the introduced defense approach was shown to be bypassed using a new model extraction approach not relying on the prediction probabilities.

Defending Against Physically Realizable Attacks on Image Classification
Tong Wu, Liang Tong, Yevgeniy Vorobeychik in ICLR, 2020 [477], *Defense Methods*
It is demonstrated experimentally that known defenses such as adversarial training (Madry, and curriculum AT by Cai et al. and randomized smoothing are not very effective against physically realizable attacks (Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition - eyeglasses, and Physical Adversarial Examples for Object Detectors stickers on stop signs). Employing their new attack of placing adversarially chosen rectangular stickers (content of patch and position optimized) into images, the authors show that adversarial training with these untargeted occlusion-based attacks (ROA- rectangular occlusion attack) is more effective against physically realizable attacks. The attack involves finding an optimal position for a grey rectangle and then performing PGD on this position to find the optimal adversarial content for the rectangle. Usual AT is then performed on these attack images.

Defending Neural Backdoors via Generative Distribution Modeling
Ximing Qiao, Yukun Yang, Hai Li in NIPS, 2019 [345], *Defense Methods*
The authors present a method to detect backdoor triggers in NNs calledmax-entropy staircase approximator (MESA). Opposed to previous work, the authors do not reconstruct single specific triggers but rather use a GAN which models the distributions of all possible triggers.

Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training
Haichao Zhang, Jianyu Wang in NeurIPS, 2019 [525], *Defense Methods*
The authors propose to produce adversarial examples in feature space (feature scattering) in an unsupervised manner for AT. This is done by maximizing the optimal transport distance between the empirical distributions of features. To this end, also data points in the current batch are considered when computing the adversarial example (collaborative learning).

Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser
Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu in CVPR, 2018 [249], *Defense Methods*
The authors present their high-level representation guided denoiser (HGD), which uses a U-net model to perform input denoising.The loss of this denoising network is based on the difference between top-level outputs of the original target model when processing original or adversarial examples. Hence, this modellearns to reproduce the adversarial perturbations rather than the complete input images. Finally, with this approach, new inputs are cleansed and then classified

by the targets. In a later study, the defense was shown to be ineffective against adaptive attacks [15].

Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models
Pouya Samangouei, Maya Kabkab, Rama Chellappa in ICLR, 2018 [378], *Defense Methods*
The authors present the defense method called DefenseGAN. In this approach, during training, the distribution of the benign input samples is trained using a GAN. During test time, the inputs and the trained GAN are then used to generate a new sample for each input which is then used as a proxy during the classification. The authors show that this approach effectively protects neural networks against non-adaptive attacks. The experimental setup is limited to the MNIST and Fashion-MNIST data sets. Hence, it is not clear whether the approach can be used in real-world scenarios and complex data sets. Furthermore, the approach was further evaluated and it was shown that DefesneGAN is partially based on gradient obfuscation and therefore provides a limited level of security [16]. It is worth mentioning that adaptive attacks using BPDA only reached a success rate of 48%.

Detecting AI Trojans Using Meta Neural Analysis
Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, Bo Li in S&P, 2021 [496], *Defense Methods*
The authors present a backdoor detection method called Meta Neural Trojan Detection (MNTD). MNTD consists of a meta classifier which can distinguish between benign and trojaned models. The authors introduce a set of benign and trojaned models, which they use during the training of their method. A meta classifier then distinguishes between the benign and trojaned models in the training set. As MNTD only operates on the outputs of the models triggered by a specific set of inputs, the detection approach also works in back-box settings. During training, the authors optimize this set of queries to further improve the detection process. In their thorough evaluation, the authors test MNTD against adaptive attacks and report that such adversaries can fully bypass the detection. Hence, the authors introduce a robust version of MNTD in which the detection is based on internal randomness. With this measure, the authors are able to circumvent nearly 90% of adaptive attacks. Furthermore, in their experiments, the authors show that their approach outperforms four other detection methods.

Detecting Adversarial Samples for Deep Learning Models: A Comparative Study
Shigeng Zhang, Shuxin Chen, Xuan Liu, Chengyao Hua, Weiping Wang, Kai Chen, Jian Zhang, Jianxin Wang in IEEE Transactions on Network Science and Engineering, 2021 [534], *Defense Methods*
The paper presents a comparison of the latest and best performing methods to detect adversarial examples. The compared methods are: SPBAS, ML-LOO, KDBU, LID, and MAHA. The paper provides a good overview of the chosen detection schemes and provides valuable insights on each method. Furthermore, the quality of the technical comparison of the chosen detection schemes presented in this paper is on a high level. This results in a concise and easy to follow comparison of the methods. The paper concludes that the ML-LOO technique provides the most

reliable approach of detection adversarial examples. The downside of ML-LOO is its computational cost and hence, time needed to train and use the detection approach. Furthermore, the authors provide valuable insights in currently used adversarial generation techniques and show how to assess the computational time of various detection schemes.

Detecting adversarial examples from sensitivity inconsistency of spatial-transform domain

Jinyu Tian, Jiantao Zhou, Yuanman Li, Jia Duan in AAAI, 2021 [433], *Defense Methods*
The authors design a detection based on the assumption that adversarial examples lie within decision regions of high curvature. Previous work by Fawzi et al. [128] showed that these regions minimize the perturbation budget as the sample under attack can easily be shifted to another class. Benign inputs, however, usually lie next to flat decision boundaries. In their method, the authors introduce a second model, which is conditioned on flattening highly curved parts. They do so by using a wavelet transform on the training data. By measuring the inconsistency between the original and the flattened model, adversarial inputs are detected. Future work showed that the detection method is prone to orthogonal PGD attacks.

Detecting adversarial examples through image transformation
Shixin Tian, Guolei Yang, Ying Cai in AAAI, 2018 [434], *Defense Methods*
The authors present an adversarial example detection method. First, the authors create multiple perturbed instances of the original inputs which are fed to the target NN to perform classifications. The classification outputs of all versions of the inputs are then fed to a detector which is trained to distinguish between adversarial and benign ensembles of inputs. In their evaluation, the authors perform a proper evaluation of adaptive attacks and report that their method can be bypassed with the C&W attack. Motivated by this, the authors further adapt their defense and perform the required perturbations in a randomized manner. This randomized version of their defense is then again shown to be robust to the previously used adaptive attacks. It its worth mentioning, that the authors did not use EoT-based attacks which is typically done to bypass randomized defenses.

Detecting adversarial samples from artifacts
Reuben Feinman, Ryan R. Curtin, Saurabh Shintre. Andrew B. Gardner, 2017 [130], *Defense Methods*
The authors present a method to detect adversarial examples. The method uses two features extracted from dropout neural networks as input to train a simple logistic regression model performing the detection. First, the density estimate: based on the analysis of the last hidden layer of the target network this feature quantifies the distance between a given sample and the submanifold of the class. Second, the Bayesian uncertainty estimate: with this feature the authors try to detect samples which lie in low-confidence regions of the original input space.

Detection Based Defense Against Adversarial Examples From the Steganalysis Point of View

Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, Nenghai Yu in CVPR, 2019 [252], *Defense Methods*
The authors build an adversarial example detection method inspired by steganalysis, i.e., the detection of hidden information in e.g. images. They do so by porting two feature extractors for steganalysis: both model the input image as Markov process, where each pixel is dependent of the neighboring ones. Each feature dimension describes the pixel-wise difference depending on the direction of the neighboring pixel. The weight of the adversarially modified pixels is increased. A separate classifier, based on a Fisher linear discriminant analysis, then learns to distinguish between benign and adversarial inputs. Future work showed that the detection method is prone to orthogonal PGD attacks [48].

Detection by Attack: Detecting Adversarial Samples by Undercover Attack
Qifei Zhou, Rong Zhang, Bo Wu, Weiping Li, Tong Mo in ESORICS, 2020 [543], *Defense Methods*
Detection by Attack (DBA) describes an adversarial detection method based on attacking a certain classifier. A simple binary classifier is used as detection method: it receives the hidden activations in the classifier based on the normal sample and the attacked sample. During training, each sample is transformed to an adversarial example by an FGSM-like attack. Weaknesses of this paper are the prerequisites on the original classifier and the rather convoluted evaluation.

Disentangling Adversarial Robustness and Generalization
David Stutz, Matthias Hein, Bernt Schiele in CVPR, 2019 [416], *Defense Methods*
The authors analyze generalization and robustness properties linked to the data manifold and propose on-manifold AT for the cases where the manifold is known (or can be approximated), arguing that this improves robustness.

Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks
Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami in S&P, 2016 [334], *Defense Methods*
The distillation method was originally designed to reduce the size of NNs by transferring the knowledge from trained models to smaller ones. This is achieved by letting the original NN classify inputs and extracting its output probabilities. These are then used as training inputs for the smaller NN. A temperature value T controls the softmax outputs of the original NN. Papernot et al. found that adversarial attacks usually aim at the sensitivity of the NNs. Hence, the authors argued that using high-temperature softmax reduces the smaller models sensitivity to small perturbations. Later studies showed that defensive distillation can be bypassed by newer attacks and therefor provides not robustness enhancement [62].

EMPIR: Ensembles of Mixed Precision Deep Networks for Increased Robustness Against Adversarial Attacks
Sanchari Sen, Balaraman Ravindran, Anand Raghunathan in ICLR, 2020 [386], *Defense Methods*
The authors present a defense method which is based on the observation thatquantized neural networks often show higher levels of robustness to adversarial examples compared to full

precision models. Hence, to build their defense method the authors train multiple NNs for the same task with different levels of precision. For this purpose, the authors quantize either the activations, the weights, or both using different numbers of bits. The created models are then simultaneously used during inference and the final decision is based on a majority vote. In a later analysis presented by Tramer et al. [438], the defense method was shown to be ineffective against adaptive attacks.

Efficient Adversarial Training With Transferable Adversarial Examples
Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, Atul Prakash in CVPR, 2020 [541], *Defense Methods*
In this paper, a method for more efficient adversarial training is proposed. Based on the observation that images remain adversarial for models in neighboring epochs (i.e. are transferable), the examples are re-used, an iterative ad. attack training based on an accumulative PGD-k attack is introduced.

Efficient Defenses Against Adversarial Attacks
Valentina Zantedeschi, Maria-Irina Nicolae, Ambrish Rawat in AISec, 2017 [517], *Defense Methods*
The authors present a defense method that performs a Gaussian data augmentationduring training and uses the BReLU activationfunction in the NNs to protect. With this preprocessing step, the authors tryto break the induced adversarial features such that the samples can again be classified correctly. Theauthors do not claim perfect security, yet their defense method increases the visual perception of attacks, thus allowing easier detection of attacks for human observers.

Enhancing Adversarial Defense by k-Winners-Take-All
Chang Xiao, Peilin Zhong, Changxi Zheng in ICLR, 2020 [481], *Defense Methods*
The authors introduce a new activation function which is claimed to improve the models robustness once applied during training and inference. Thediscontinuous k-Winners-Take-All (k-WTA) function is designed to intentionally mask the gradients of the NNs calculated during backpropagation. Even though defense methods based on gradient masking/hiding are shown to be vulnerable to black-box or gradient-free attacks, the authors report promising results and argue that their method might even be improved in combination with adversarial training. In a later study by Tramer et al. [438], the defense method was shown to be ineffective against adaptive adversaries using decision based attacks. Furthermore, Tramer et al. [438] shown, that the defense method even reduces the effectiveness of adversarial training when used in combination.

Ensemble adversarial training: attacks and defenses
Florian Tramer, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Patrick McDaniel in ICLR, 2018 [439], *Defense Methods*

The paper proposes a slightly different way to generate adversarial examples by FGSM and LLC via adding a random step in the beginning. They also propose an advanced adversarial training, where adversarial examples are produced not for one target model, but for an ensemble of models.

Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks
Gunjan Verma, Ananthram Swami in NeurIPS, 2019 [453], *Defense Methods*
This paper presents an ensemble method which trains multiple models to allow robust classifications. Each model performsa binary classification of a subproblem of the dataset. Additionally, the method enforces diversity during the classifications such that the produced redundancy can act as error correcting codes. In a following evaluation by Tramer et al. [438], it was shown that the defense relies on obfuscated gradients due tonumerical instabilities of the outputs including the use of a softmax layer. By adapting the attack accordingly, Tramer et al. were able to bypass the defense method.

Evaluating Differentially Private Machine Learning in Practice
Bargav Jayaraman, David Evans in USENIX, 2019 [201], *Defense Methods*
The authors present a general analysis of differential privacy in the field of machine learning. Specifically, the authors analyze the impact of DP-related parameters when applied for logistic regression models and NNs. For this purpose, the authors evaluate different relaxations in differential privacy and quantify their impacts on the resulting privacy leakage level. The key finding of the authors is the observation that the privacy for ML models does not come for free and depends on the trade-off between performance and privacy. Hence, the authors argue thatby reducing the required added noise which achieves better classification results, the privacy leakage is increased.

Evaluating and Understanding the Robustness of Adversarial Logit Pairing
Logan Engstrom, Andrew Ilyas, Anish Athalye in NeurIPS SECML, 2018 [120], *Defense Methods*
The adversarial logit pairing technique is shown to be not effective.

Explaining and harnessing adversarial examples
Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy in ICLR, 2015 [152], *Defense Methods*
The fast and simple way (FGSM) to generate adversarial examples, that also allows for adversarial training, is presented. In this paper, the authors assume that linearity of a network is makes it attackable by adversarial examples and they observe that adv. Examples are transferrable (across models). The FGSM method is based on computing the sign of the loss functoin gradient and adding it (scaled with epsilon) to the original input (one-step approach).

Exploring Connections Between Active Learning and Model Extraction
Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, Songbai Yan in USENIX, 2020 [66], *Defense Methods*

The authorsreport similarities between research fields of active learning and model stealing methods. Based on this observation, the authors present the first formalization of model stealing attacks. Furthermore, leveraging the knowledge from active learning and transferring related concepts, the authors present new approaches to improve model stealing attacks and defenses. Even though the paper presents important findings in this field, the approach cannot be applied to complex and non-linear models like NNs yet.

Fast homomorphic evaluation of deep discretized neural networks
Florian Bourse, Michele Minelli, Matthias Minihold, Pascal Paillier in CRYPTO, 2018 [43], *Defense Methods*
The authors present a fully homomorphic encryption and operation scheme for NNs. For this purpose, the authors use quantized NNs with simple sign activation functions to allow the FHE operation. The authors call their models discretized neural networks (DiNNs), which the authors argue, pose a special form of binarized neural networks (BNNs). Furthermore, the authors make use of the previously introduced bootstrapping technique [84] to reduce the complexity of the performed operations.

Fast is better than free: Revisiting adversarial training
Eric Wong, Leslie Rice, J. Zico Kolter in ICLR, 2020 [472], *Defense Methods*
An effective and fast method for adversarial training is proposed that uses FGSM with some modifications (in particular using random initialization points and speed-up methods for general DNN training). This adversarial training is shown to be as effective as the PGD-based adv. training (which is considered the much stronger attack).

Feature Denoising for Improving Adversarial Robustness
Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, Kaiming He in CVPR, 2019 [488], *Defense Methods*
The authors present a defense method which is based on adversarial training and model modifications to make NNs more robust. The authors first argue, that adversarial perturbations areamplified layer-by-layer when propagated through the NNs. Hence, this effect results in a large amount of noise in the NNs feature maps. Motivated by this hypothesis, the authors propose to extend NN architectures with feature denoising blocks which aim torectify the features learned by the intermediate layers of the models. Finally, the authors adversarially train these new architectures using known approaches and successfully improve the robustness of simply adversarially trained NNs. Moreover, the authors show that their approach increases the robustness for ImageNet processing NNs.

Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks
Weilin Xu, David Evans, Yanjun Qi in NDSS, 2018 [495], *Defense Methods*
This paper introduces the defense method called Feature Squeezing which is a combination of input pre-processing and adversarial example detection. First, a given input is smoothened using two techniques: reducing the color bit depth and performing spatial smoothing. Then, the

original input as well as the two smoothened versions of it are classified by the neural network to protect. If the classification outputs of the neural network for the three given inputs differ significantly, the inputs are rejected and considered adversarial. The authors use a threshold to decide when the outputs differ too much and the inputs should be considered adversarial. Even though the paper is well cited and accepted by the research community and achieves good results when detecting adversarial examples in a grey-box setting, feature squeezing can by circumvented by adaptive attacks.

Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems
Bao Gia Doan, Ehsan Abbasnejad, Damith C. Ranasinghe in ACSAC, 2020 [105], *Defense Methods*
The authors present a defense method, called Februus, against backdoor trojan attacks for NNs. Februus follows an input purification approach consisting of two steps. In the first step, the backdoor triggers present in the input images are detected and removed. For this purpose, the authors use the saliency method called GradCAM. This approach allows to quantify and visualize which regions of the inputs were the most important for the decision of the NNs. Hence, by removing these regions, the authors argue that in most attack cases, the backdoor triggers are removed. In the second step, to preserve the performance of the model, the authors try to reconstruct the original images using a GAN specifically trained for this purpose. The advantage of Februus is the fact that the NNs are not changed and hence the classification performance is preserved.

Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks
Kang Liu, Brendan Dolan-Gavitt, Siddharth Garg in RAID: International Symposium on Research in Attacks, Intrusions, and Defenses, 2018 [253], *Defense Methods*
The authors present a defense method against backdoor poisoning attacks. In the first step, the authors perform pruning to remove the neurons which are responsible for the backdoor triggers. The authors argue that such neurons are usually only active if the current input contains the trigger and otherwise are dormant. To leverage this observation, the pruning is performed using benign inputs which are fed to the NN while iteratively pruning an increasing amount of neurons until the accuracy falls below a predefined threshold. The authors note that this defense approach disables triggers but can be bypassed if the attacker performs pruning-aware attacks. Therefore, the authors suggest to subsequently perform a fine-tuning process. Here, the pruned NNs are fine-tuned which is a short training process using a smaller learning rate. In their evaluation, the authors perform experiments using image and audio processing NNs.

Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking
Erwin Quiring, Daniel Arp, Konrad Rieck in EuroS&P, 2018 [349], *Defense Methods*
The authors present a defense method which detects model stealing attacks. Similar to other model extraction defense methods, the authors record the queries made to the NN and measure the distances of the queries to the respective class boundaries. This distance is then used to distinguish benign queries and queries related probably related to extraction attacks. Even though

the approach marks an important step towards defense methods against model stealing attacks, the approach cannot be applied to deep NNs usually found in real-world applications. Furthermore, the approach relies onlinearly separated prediction classes which is usually not found in NNs.

Fully homomorphic encryption using ideal lattices
Craig Gentry in STOC, 2009 [144], *Defense Methods*
This paper introduces fully homomorphic encryption. With this approach, operations on encrypted data are possible which can be applicable in the context of machine and deep learning. Following this paper, a series of publications tried to use the concept to allow learning and inference using encrypted data to enhance the privacy of the users.

GAT: Generative Adversarial Training for Adversarial Example Detection and Robust Classification
Xuwang Yin, Soheil Kolouri, Gustavo K Rohde in ICLR, 2020 [513], *Defense Methods*
This paper presents a defense method which uses adversarially trained models to detect adversarial examples. For each class of the dataset, the authors adversarially train one detector model.The training of the detectors is designed such that the detector of the correct class recognizes the sample as benign while the other detectors reject perturbed versions of the input. In their paper, the authors present thorough evaluations of adaptive white-box attacks and argue that their method shows high robustness in this case. This observation was partially confirmed by Tramer et al. [438]. In this study, Tramer et al. further present an improved adaptive attack and which is able to reduce the robustness of the system well below the level of robustness obtained with standard PGD adversarial training.

GangSweep: Sweep out Neural Backdoors by GAN
Liuwan Zhu, Rui Ning, Cong Wang, Chunsheng Xin, Hongyi Wu in ACM MM, 2020 [547], *Defense Methods*
The authors present a defense method called GangSweep which first detects backdoor triggers and then makes them ineffective. The authors leverage the previously presented defense method called Neural Cleanse and further improve upon the findings. Opposed to Neural Cleanse, the authors in this paper use GANs to generate perturbations which are added to inputs fed to the NNs. Again, the outputs of the NNs are observed for clean and perturbed data samples. The method assumes the analyzed NNs to be backdoored if a universal perturbation mask can be found which leads tomisclassifications for all samples. In Neural Cleanse, the authors used standard evasion attack algorithms rather than GANs. Once a model is considered to be backdoored, standard mitigation methods are applied, like fine-tuning in which the models are retrained with perturbed samples and the original ground-truth labels.

Geometry-aware Instance-reweighted Adversarial Training
Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, Mohan Kankanhalli in ICLR, 2021 [532], *Defense Methods*

A reweighting approach for AT is proposed: geometry-aware instance-reweighted adversarial training (GAIRAT). The idea is to reweight clean examples (and thus the loss of the corresponding adversarial ones) according to the difficulty to attack them (i.e. also to their closeness to decision boundary: natural data that is misclassified gets higher weights since the decision boundary needs to be tuned there). Reweighting is performed by giving weight according to the minimal number of PGD iterations needed (different weight functions are constructed). This should help the model to focus on the important points and helps fit the adversarially robust model given a limited model capacity (in light of the complex problem), preventing overfitting. The method can be added to existing AT approaches.

Gotta CatchEm All: Using Honeypots to Catch Adversarial Attacks on Neural Networks
Shawn Shan, Emily Wenger, Bolun Wang, Bo Li, Haitao Zheng, Ben Y. Zhao in CCS, 2020 [391], *Defense Methods*
The authors propose a detection method based on the insertion of deliberate weaknesses in neural networks, i.e., by introducing so-called honeypots. When an adversary mounts an attack, the adapted loss function causes to find the honeypot with high likelihood. A honeypot could e.g. be a specific shape like a square in the image. In other words, the defender shifts the attack to her desire, thus can compare the input to the known attack. The similarity between the input and the honeypot image is used as detection score. Future work showed that the detection method is prone to orthogonal PGD attacks. Nonetheless, the paper introduces a novel and interesting direction for defenses, which may be improved in future work.

GraN: An Efficient Gradient-Norm Based Detector for Adversarial and Misclassified Examples
Julia Lust, Alexandru P. Condurache in European Symposium on Artificial Neural Networks, 2020 [269], *Defense Methods*
The authors present an approach to detect adversarial examples based on an analysis of the gradients observed during back-propagation using benign and adversarial examples. In the training phase of the detector, the authors perform back-propagation using the training samples as well as smoothened versions of them. For each pair, the norm of the gradient is computed. This process is performed using adversarial, as well as benign pairs of inputs. To finally detect adversarial examples, the observed norms of the gradients are trained via logistic regression. With this detector, the authors are able to detect adversarial examples created for the unsecured target model with high accuracy and relatively low computational cost compared to state-of-the-art detectors. The authors perform no adaptive attack evaluation. Hence, the actual robustness increase based on the approach is not evaluated.

Image Super-Resolution as a Defense Against Adversarial Attacks
Aamir Mustafa, Salman H. Khan, Munawar Hayat, Jianbing Shen Ling Shao in IEEE Transactions on Image Processing (Journal), 2020 [306], *Defense Methods*
The authors present a defense method which is based on input preprocessing breaking the adversarial perturbations to again allow a correct classification of the samples. For this purpose,

the authors combine two image processing methods into one model which applied to the inputs prior of being fed to the target NN.First, the adversarial perturbations are suppressed by applyingsoft wavelet denoising. Second, the visual quality of the images is enhanced by performing asuper resolution step. The authors state that parts of their approach are non-differentiable and thus robust to adaptive attackers. In recent studies it was shown, that non-differentiable steps in defenses may lead to obfuscated gradients and thus to a false sense of robustness. To evaluate if this holds true for their approach, the authors perform adaptive attacks using BPDA and EoT. Yet it remains unclear whether the attacks are performed correctly.

Improved Network Robustness with Adversary Critic
Alexander Matyasko, Lap-Pui Chau in NeurIPS, 2019 [280], *Attacks on Deep Learning Systems*
An approach of adversarial training formulated as a GAN-framework is introduced. The adversarial attack (needs to be differentiable) serves as generator, and a critic network is the discriminator (the attack image should be indistinguishable from the target class to make sure that the attack image does not). The classifier loss considers that the critic should get confused by the attack. The labels should get modified by the adversarial examples (true images of the target class are wanted, that would also confuse humans) in contrast to the usual assumption in AT that labels should be preserved by the perturbations added. The basic assumption is that a classifier is considered robust if the adversarial examples for it correspond to the target class visually. The loss for this method is composed of classifier loss as well as a distance between adversarial and clean data distribution. The problem of estimating these distributions is tranferred to a GAN-setup, where a discriminator tries to distinguish between clean and adversarial examples. A cycle-consistency term ensures that the generated examples are close to the original datapoints. A new attack algorithm is proposed that also constraints on the confidence (aiming at highest possible confidence reduction), similar to basic iterative method. The authors argue that their approach can be extended to other domains such as audio or text.

Improving Adversarial Robustness Requires Revisiting Misclassified Examples
Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, Quanquan Gu in ICLR, 2020 [463], *Defense Methods*
Misclassification Aware adveRsarialTraining (MART) is proposed. The core idea is to weight misclassified examples - in particular also misclassified natural images - differently than those correctly classified and thus performing regularization for misclassified examples (aiming at the network to be stable against adv. examples from misclassified images). An extension to also include unlabeled data is provided. The proposed MART performs well on black and white-box test attacks outperforming many other AT approaches (such as TRADES and standard AT) across different attack types like FGSM, CW_∞ and PGD.

Improving Adversarial Robustness via Channel-wise Activation Suppressing
Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, Yisen Wang in ICLR, 2021 [24], *Defense Methods*

The authors present a defense method which combines adversarial training and an analysis of the activation values of NNs. The authors study the behavior of normally and adversarially trained NNs when processing benign and adversarial examples. For this purpose, the authors introduce two metrics: the channel-wise activation magnitude and the channel-wise activation frequency. With these metrics at hand, the authors find that adversarially trained models behave differently compared to normally trained NNs. Furthermore, the authors observe, that adversarial examples trigger redundant channels of both naturally and adversarially trained NNs with a slightly higher level in normally trained ones. Therefore, the authors present their approach calledChannel-wise Activation Suppressing (CAS) trying to counteract this effect. In combination with adversarial trained, CAS increased the robustness of NNs.

Improving Adversarial Robustness via Promoting Ensemble Diversity
Tianyu Pang,Kun Xu,Chao Du,Ning Chen,Jun Zhu in ICML, 2019 [326], *Defense Methods*
The authors suggest to promote ensemble diversity (using an adaptive diversity promoting (ADP) regularizer) in the non-maximal predictions of the individual networks (allowing for high overall accuracy since output classes are not changed). The diversity aims at reducing transferability of adv. examples among the networks in the ensemble and thus is supposed to increase robustness. The networks are then trained simultaneously on the same dataset, employing a regularization term as penalty. The regularization consists of a term for Shannon entropy and a term for the logarithm of ensemble diversity (defined via determinant of matrix multiplications of all outputs except the true class).

Improving adversarial robustness via promoting ensemble diversity
Tianyu Pang, Kun Xu, Chao Du, Ning Chen, Jun Zhu in ICML, 2019 [326], *Defense Methods*
The authors present a defense method which ensembles multiple NNs. Additionally, to make attacks more difficult and increase the robustness of the classification, the authors present a new training objective to train the individual NNs forming the ensemble whilesimultaneously encouraging diversity of the NNs outputs. This diversification is argued to further increase attack difficulty. Even though the authors perform adaptive attacks, Tramer et al. [438] find in a later evaluation of the defense method that the authors used only a small number of attack steps during their evaluation. Hence, by simply increasing the number of attack steps, Tramer et al. were able to bypass the defense method and successfully generate adversarial examples.

Improving the Generalization of Adversarial Training with Domain Adaptation
Chuanbiao Song, Kun He, Liwei Wang, John E. Hopcroft in ICLR, 2019 [411], *Defense Methods*
The authors propose AT withdomain adaptation (ATDA), where the target domain is the adversarial domain. They assume that only limited examples from this domain are available (i.e., only some attacks can be performed, not being representative of the whole space). The idea is to minimize the distributional shift between representations of clean and adversarial data and thus to promote robust generalization by formulating a loss over covariance matrices and mean vectors (MMD) in the unsupervised case. A supervised DA loss is also proposed to account for the known

attacks. This approach is applied to FGSM, experimentally showing better robust generalization (l_∞ attacks) but a bit worse clean accuracy.

Improving the Robustness of Deep Neural Networks via Adversarial Training with Triplet Loss
Pengcheng Li, Jinfeng Yi, Bowen Zhou, Lijun Zhang in IJCAI, 2019 [241], *Defense Methods*
The authors propose adversarial training with triplet loss (AT2L) from metric learning, aiming at smoothing the decision boundary of the classifiers. An ensemble AT is also discussed. The triplet loss consists of a positive pair (clean and adversarial example - same label) and a negative pair (adv. example and sample from minibatch having other class). Then, the loss encourages to have smaller distance between the positive than the negative pair and thus to learn a broad margin between the classes. The triplet loss is added to the AT loss. For the ensemble version, several types of attacks and attacks for different models are considered.

Indicators of Attack Failure: Debugging and Improving Optimization of Adversarial Examples
Maura Pintor, Luca Demetrio, Angelo Sotgiu, Giovanni Manca, Ambra Demontis, Nicholas Carlini, Battista Biggio, Fabio Roli in arXiv, 2021 [342], *Defense Methods*
The authors present a framework which allows the evaluation of performed adversarial attacks. The motivation to create this framework stems from the fact that adversarial attacks are usually used to estimate the robustness of NNs and accompanying defense strategies. The authors argue, that oftentimes the quality of the performed attacks are poor and thus falsely suggest a high level of robustness of the NNs. Yet when further improved and optimized, the attacks are more successful and therefor show the level of robustness of the evaluated NN more properly. The presented framework can be added to existing robustness evaluation frameworks to further improve them.

InstaHide: Instance-hiding Schemes for Private Distributed Learning
Yangsibo Huang, Zhao Song, Kai Li, Sanjeev Arora in ICML, 2020 [192], *Defense Methods*
The authors present a privacy preserving defense method for NNs which claims not to reduce the performance of the protected models. InstaHide is designed for the image classification domain. To increase the privacy of the used training samples, InstaHide follows a two-step approach. First, each sample is combined with a set of randomly chosen images. In the second step, for all images, which were previously normalized to contain pixels in the range-1, 1, the signs of the pixels are randomly flipped. The resulting encoded images, which at first glance appear to be random noise, are then used to train the NN. In a later study by Carlini et al. [57], InstaHide was shown to be broken and thus not providing an increase privacy level for the used training samples.

Interpolated Adversarial Training: Achieving Robust Neural Networks Without Sacrificing Too Much Accuracy
Alex Lamb, Vikas Verma, Juho Kannala, Yoshua Bengio in AISec, 2019 [229], *Defense Methods*

The authors propose interpolated AT (IAT), a method that uses interpolations between adversarial examples and clean examples (based on MixUp or Manifold Mixup where random linear interpolations between two datapoints are considered). Also the training is on clean examples (in a first step, before computing the adv. examples). The idea behind it is that the interpolation can serve as increasing the dataset size and also lead to learning more compressed features.

Intriguing properties of neural networks
Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus in arXiv, 2014 [424], *Defense Methods*
First work with demonstration of the epsilon-perturbation attacks vulnerability of deep neural networks. Demonstrated an algorithm for generating such attacks, possible way to protect the model against them and the transferability of such examples.

Is Private Learning Possible with Instance Encoding
Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, Florian Tramer in S&P, 2021 [57], *Defense Methods*
The authors perform a study to evaluate whether instance encoding can improve privacy preserving training for NNs. The authors conclude that the approach of encoding training samples with cryptographic schemes cannot lead to provable privacy. In their evaluation, the authors further present an attack method for the previously presented method called InstaHide. With this attack, the authors are able to fully break the defense and extract samples used to train the attacked NNs.

Learning with a Strong Adversary
Ruitong Huang, Bing Xu, Dale Schuurmans, Csaba Szepesvari in arXiv, 2015 [188], *Defense Methods*
This paper presents an AT approach for robust models, using the min-max formulation to be robust to worst-case examples.

ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation
Yuzhe Yang, Guo Zhang, Dina Katabi, Zhi Xu in ICML, 2019 [504], *Defense Methods*
The authors present a defense method which preprocesses the train and test data to purify the samples to again allow a correct classification. For this purpose, in the first step the authors drop each line of the input matrix showing the current sample with a certain probability. In the next step, based on the altered samples and differentmatrix-estimation techniques, the original images are tried to be reconstructed. The resulting samples are then used either to train the model or to perform the classification. Even though the authors perform adaptive attacks in their evaluation, Tramer et al. [438] find that the method is vulnerable to properly designed attacks.

ML-LOO: Detecting Adversarial Examples with Feature Attribution
Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, Michael I. Jordan in AAAI, 2020 [503], *Defense Methods*
ML-LOO detects adversarial inputs based on the feature attribution method Leave-One-Out (LOO). Here, the importance of each feature is measured by the change in the output when the feature is omitted. The same principle is applied to measure the impact on the inner layers of NNs to increase the information gain. In their evaluation, the authors show that the interquartile range of the feature attribution distribution can be used to detect adversarial inputs with high confidence.

MMA Training: Direct Input Space Margin Maximization through Adversarial Training
Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, Ruitong Huang in ICLR, 2020 [104], *Defense Methods*
The authors propose Max-Margin Adversarial training (MMA, find maximal margins between inputs to decision boundaries) to achieve adversarial robustness. For each datapoint, an individual epsilon-value is allowed (adaptive epsilon) - thus solving the problem of choosing the correct epsilon for AT. This is realized by minimizing the classification loss (w.r.t. model parameters) at the shortest successful perturbation point for correctly classified natural examples and minimizing classification loss on incorrectly classified data points. Finding the shortest successful perturbation is achieved via Adaptive Norm-PGD attacks that tries to find an attack right at the decision boundary. The approach is tested for l_2 and l_∞ perturbations.

MagNet: A Two-Pronged Defense against Adversarial Examples
Dongyu Meng, Hao Chen in CCS, 2017 [284], *Defense Methods*
The authors present the defense method called MagNet which consists of a detector and a reformer used in an autoencoder structure. This autoencoder is used to learn the manifold of the benign data. During inference, MagNet operates in two setups. If a new samples lies close to the previously trained manifold, the reformer processes the data such that new samples lie on the data manifold. This causes the adversarial perturbations to be removed and hence the samples are classified correctly. If the current input lies far away from the learned data manifold, the current sample is considered to be adversarial and is thus rejected.

Making Convolutional Networks Shift-Invariant Again
Richard Zhang in ICML, 2019 [533], *Defense Methods*
The author identifies a major concern of commonly used NN architectures: the used pooling layers oppose any shift-variance. As consequence, the internal feature representation, and moreover, the output may change significantly when the very same input picture is slightly translated or transformed. The author identifies the max-pooling layers as cause, which increase the performance, but also cause severe value shifts under transformations. As solution, an anti-aliasing filter, i.e., a blurring operation, is introduced after the max-pooling layer. The evaluation shows that the resulting model has a less volatile internal feature representation, which results in increased robustness against natural perturbations like noise and blur.

Membership Inference Attacks Against Machine Learning Models
Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov in S&P, 2017 [399], *Defense Methods*
The authors present a new strategy to perform membership inference attacks. Such attacks determine if a specific input sample was part of the training process of the NN. In their evaluation, the authors test various intuitive defense methods and show that these are ineffective against their attack. This shows that more complex defense strategies like differential privacy and homomorphic encryption are need to protect against membership inference attacks. The defenses evaluated include prediction vector tampering in which the outputs of the model are restricted to the k-top-classes or lowering the precision of the prediction vector of the NNs.

Metric Learning for Adversarial Robustness
Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, Baishakhi Ray in NeurIPS, 2019 [279], *Defense Methods*
Triplet Loss Adversarial (TLA) training is introduced. This is based on the observation that adversarial attack representations lie closer to the false class than to the true class and should therefore be pushed closer to the true class, building a margin to the wrong class (same classes are learned to be pushed closer together and further away from other classes). The approach uses a special metric learning regularization term that considers penultimate representations of adversarial examples (anchor, thus considered in both positive and negative pair), of the true class and of the negative class (from a given clean example, an adversarial example is generated with PGD to be used in the loss). As negative example, the nearest wrong-class image of the current minibatch to the adversarial example is chosen.

Mitigating Adversarial Effects Through Randomization
Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan Yuille in ICLR, 2018 [486], *Defense Methods*
The authors present a defense method which is based on input pre-processing during test time to break the adversarial characteristics of the current inputs. This pre-processing is done in a random manner and includes random resizing and random padding of the input images. In later publications and evaluations of this defense method it was shown that the approach provides a reasonable level of robustness in the case of grey-box and black-box settings. Still, during white-box attacks based on the Expectation over Transformation (EoT) approach [17] it was shown that the defense can be bypassed.

Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification
Xiaoyu Cao, Neil Zhenqiang Gong in ACSAC, 2017 [55], *Defense Methods*
The authors introduce the region-based classification defense method. This defense computes each prediction over an ensemble generated from the input sample in a randomized fashion. More precisely, from each input image 10.000 new versions of it are sampled from the cube around the original input itself. The resulting perturbed versions of the original input are classified. Finally, performing a majority vote, the final decision of the model to protect is generated.

Mixup Inference: Better Exploiting Mixup to Defend Adversarial Attacks
Tianyu Pang, Kun Xu, Jun Zhu in ICLR, 2020 [327], *Defense Methods*
The authors present a defense method which tries to purify inputs in order to break the adversarial features allowing a correct classification. The changes to the inputs are performed during training and testing to maintain the accuracy of the protected NNs. During training and testing, the authors add multiple new images to the currently processed input with respect to a weighting factor to control the influence of the added images. The batch of resulting samples is classified by the NN while the mean of the logit outputs is calculated to form the final decision of the classification. Even though the authors perform adaptive attacks, Tramer et al. [438] find that the defense method can be bypassed using better suited attacks. Instead of averaging over multiple adversarial examples, Tramer et al. average the gradients produced during the attacks and successfully circumvent the defense method.

Model Agnostic Defence against Backdoor Attacks in Machine Learning
Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, Sudipta Chattopadhyay in arXiv, 2019 [448], *Defense Methods*
The authors present a backdoor detection and removal method called NEO. The approach consists of three steps. A so-called trigger blocker is created, which is a patch with the dominant color of the currently processed image. In the second step, this trigger blocker is moved randomly across the image. Based on the classification output of the NN, the existence and position of a backdoor trigger is determined. The trigger blocker can then be used to purify the classifiers output.

Model extraction warning in MLaaS paradigm
Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, Sameep Mehta in ACSAC, 2018 [216], *Defense Methods*
The authors present a defense method which protects machine learning models from stealing attacks. The authors record the queries required during model extraction attacks and compute the feature space explored by the set of queries. If the explored space exceeds a predefined threshold, the queries are assumed to be part of an attack. Even though the approach marks an important step towards defense methods against mode stealing attacks, the approach cannot be applied to deep NNs. Furthermore, the approach relies onlinearly separated prediction classes which usually does not hold for NNs.

Model-Agnostic Adversarial Detection by Random Perturbations
Bo Huang, Yi Wang, Wei Wang in IJCAI, 2019 [187], *Defense Methods*
Huang et al. propose an adversarial example detection method, which works solely on the input-output mapping. Thus, their method is widely applicable without needing full access to the protected NN. The authors measure the prediction difference for the current input to the noise-polluted version of itself. Intuitively, adversarial examples are less robust to random perturbations than benign samples. As decision variable, they look at the quantiles of the prediction difference for multiple random transformations.

NIC: Detecting Adversarial Samples with Neural Network Invariant Checking
Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, Xiangyu Zhang in NDSS, 2019 [272], *Defense Methods*
The authors present a method to reliably and accurately detect adversarial examples fed to NNs. The approach is based on the analysis of the hidden activation values of the NNs with a subsequent evaluation and detection using O-SVMs. In the evaluation the authors follow a sound approach and perform adaptive attacks. Here the question arises, why the authors changed their approach prior to performing the adaptive attacks. Contrary to the normal case, in the adaptive attacks the authors use three detectors instead of one. Still, the results presented in this paper are very promising. In adaptive attacks, the required perturbation to fool the attacked system are significantly higher compared to the unsecured model without the detection system.

NNoculation: Broad Spectrum and Targeted Treatment of Backdoored DNNs
Akshaj Kumar Veldanda, Kang Liu, Benjamin Tan, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, Brendan Dolan-Gavitt, Siddharth Garg in arXiv, 2020 [452], *Defense Methods*
The authors present a defense method which retrains NNs circumventing backdoor attacks, calledNNoculation. The defense is deployed in two phases. In the pre-deployment stage, the NN is retrained using clean validation data and randomly perturbed counterparts of the samples. This already reduces the success rate of backdoor attacks slightly. In the post-deployment stage, the original NN and its retrained counterpart both classify inputs. If for a given sample the classification outputs differ, the sample is assumed to contain a backdoor trigger and is therefore rejected and saved to a quarantine data set. Subsequently, thequarantine data set and additional clean data samples are used to train a CycleGAN model which tries to learn the poisoning process. Hence, this CycleGAN model is able to create samples containing working backdoor triggers. With this model an additional data set is generated which is again used to further train the previously retrained target model using the original labels of the artificially generated backdoored samples. This further increases the robustness of the NN towards backdoor attacks.

Neural Trojans
Yuntao Liu, Yang Xie, Ankur Srivastava in ICCD, 2017 [264], *Defense Methods*
The authors present three defense strategies to protect NNs against backdoor attacks. In the first approach, input anomaly detection is performed. Here, the authors try to evaluate if the current input samples come from the same data distribution as the known benign data. The downside of this approach is the fact that the defender requires knowledge of the benign data set which might not be feasible in practice. The second approach presented by the authors is the retraining of the target models using benign training samples. The authors argue that this process overwrites weights in the NNs related to the backdoor triggers and hence making them ineffective. Finally, the third method performs input data pre-processing using autoencoders, which render the attack triggers ineffective.

Neural cleanse: Identifying and mitigating backdoor attacks in neural networks
Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, Ben Y. Zhao in S&P, 2019 [455], *Defense Methods*
The authors present a defense strategy to protect deep NNs against backdoor attacks called Neural Cleanse. The defense is based on an initial detection of triggers in the models under evaluation. If the detection reports the presence of a backdoor trigger, a reconstruction process is started. By reconstructing the triggers, the authors are subsequently able to repair the models and remove the earlier found triggers. The initial detection of backdoor triggers is based on the observation that models containing triggers require less changes to the inputs to provoke a change in the classification output. Hence, by systematically perturbing inputs and analyzing the classification outputs of the NNs, the presence of triggers can be detected. By simultaneously observing which classes are changed with the least amount of input perturbations, the triggers are identified. For the purpose of perturbing the inputs, the authors suggest to use standard algorithms used in evasion attacks like the C&W or the BIM method. In the final step, pruning (removing responsible neurons) and fine-tuning (unlearning the trigger using the correct ground-truth labels) are used to remove the backdoor triggers and again allow a secure operation of the NNs.

NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations
Xijie Huang, Moustafa Alzantot, Mani Srivastava in arXiv, 2019 [190], *Defense Methods*
The authors present a method to detect if a NN contains a backdoor, called NeuronInspect. The method is based on the observation, that saliency maps of backdoored NNs differ from those of benign models. Hence, in NeuronInspect, the authors use benign data samples to generate multiple saliency maps. From these explanation heat-maps, the authors extract multiple features like the sparseness or the smoothness. Using these features, the authors leverage outlier detection methods to finally decide whether the model contains a backdoor.

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples
Anish Athalye, Nicholas Carlini, David Wagner in ICML, 2018 [16], *Defense Methods*
The paper introduces the notion of gradient obfuscation. Gradient obfuscation provokes similar effects compared to gradient hiding yet is not intentionally introduced to actively hide the gradients of attacked neural networks (Gradient Hiding is a defense method that specifically tries to hide the gradients. This method is known to be broken). Gradient Obfuscation can be divided in three groups: 1) shattered gradients, 2) stochastic gradients, 3) vanishing & exploding gradients. The authors present a list of properties, which indicate gradient obfuscation: 1) one step attacks perform better than iterative attacks, 2) black-box attacks perform better than white-box attacks, 3) unbounded attacks do not reach 100% attack success, 4) random sampling finds adversarial examples, 5) increasing the distortion budget (i.e., epsilon) does not increase the attack success. Finally, the authors show methods to attack defenses, which are based on gradient obfuscation. The following describes which attack approach can be used to circumvent defense methods based on obfuscated gradients:1) defenses relying on non-differentiable

add-ons (e.g. quantization): approximation of the defense method using differentiable functions to approximate them, 2)defenses relying on non-deterministic operations (e.g. random transformations): approximation of the general gradient direction under possible transformations by Expectation Over Transformation (EOT), 3) defenses relying on exploding or vanishing gradients: introduction of a change of variable using differentiable functions not resulting in exploding or vanishing gradients.

On Adaptive Attacks to Adversarial Example Defenses
Florian Tramer, Nicholas Carlini, Wieland Brendel, Aleksander Madry in NeurIPS, 2020 [438], *Defense Methods*
The paper revisits a series of recently published defense methods against adversarial examples. By performing adaptive attacks, the authors are able to bypass all attack methods and successfully create adversarial examples for the protected neural networks. All evaluated defense methods are published on well-established and highly ranked ML-conference. This shows the need for proper evaluation methods when presenting new defense methods. Additionally, this shows the lack of properly working defense methods providing an increased security level of neural networks. Finally, a major point shown by the authors is the fact, that adaptive attacks need to be carefully implemented and cannot generally be used for multiple defense methods. Attacking each new defense method requires a new set or form of adaptive attacks, specifically designed to intentionally bypass the defense.

On Detecting Adversarial Perturbations
Jan Hendrik Metzen, Tim Genewein, Volker Fischer, Bastian Bischoff in ICLR, 2017 [285], *Defense Methods*
The authors present a method to detect adversarial examples which is based on the analysis of the inner representations of the processed input images. Specifically, the authors use the outputs of each convolutional layer in the NNs as input to a binary classifier finally detecting adversarial examples. With this approach the authors are able to detect adversarial examples crafted with different attack methods for the CIFAR-10 data set and a ten-class version of the ImageNet data set. In a later published study [61], it was shown that this approach can be easily bypassed by adaptive adversaries even though this threat model was considered in the original paper.

On Evaluating Adversarial Robustness
Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, Alexey Kurakin in arXiv, 2019 [56], *Defense Methods*
The authors present a guideline for the evaluation of adversarial robustness of neural networks. The authors focus on the robustness introduced by defense methods and how to properly evaluate and determine the security and robustness of defense methods. As recently numerous defense methods have been shown to be easily bypassed by adaptive attacks, the evaluation of future defense methods need to pay attention to be properly executed. This paper provides the best practices.

On the (Statistical) Detection of Adversarial Examples
Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, Patrick McDaniel in arXiv, 2017 [154], *Defense Methods*
The authors present an adversarial example detection method which extends the original data sets processed by the NN with one class summarizing adversarial examples. First, the authors train the base NN on the normal N classes of the data set. For this NN, adversarial examples are generated. In the second step, this NN is retrained with N1 classes while the previously generated adversarial examples are summarized in the new class. With this approach, the NN itself flags and detects the attacks. The proposed method fails to correctly identify adversarial examples created with the C&W attack.

On the Connection Between Adversarial Robustness and Saliency Map Interpretability
Christian Etmann, Sebastian Lunz, Peter Maass, Carola Schoenlieb in ICML, 2019 [122], *Defense Methods*
The paper presents the observation that robust neural networks produce saliency maps which are more human-interpretable compared to saliency maps produced by less robust neural networks.

On the Convergence and Robustness of Adversarial Training
Yisen Wang,Xingjun Ma,James Bailey,Jinfeng Yi,Bowen Zhou,Quanquan Gu in ICML, 2019 [462], *Defense Methods*
In this paper the authors introduce a score (FOSC: First-Order Stationary condition) that aims at evaluating how good the adversarial example is in terms of convergence quality (i.e. how well the inner maximization is solved at the current iteration round). FOSC is observed to correlate with the adversarial attack strength. Using this score, it is observed that in the early stages of training, adv. examples with high convergence quality are not needed (even on the contrary - they can harm robustness), while at the later stages it is crucial to use such high quality examples. Based on this and using the FOSC (best: 0) as a way of monitoring adversarial training, they propose a dynamic training scheme (dynamic AT), starting with weak examples and then using stronger examples (i.e., with decreasing FOSC score).

On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping
Sanghyun Hong, Varun Chandrasekaran, Yigitcan Kaya, Tudor Dumitras, Nicolas Papernot in arXiv, 2020 [175], *Defense Methods*
The authors present a defense method against backdoor attacks on NNs. Their approach is based on the observation that samples containing the backdoor triggers provoke a special behavior of the NNs gradients. More specifically, such samples lead to gradients with higher magnitudes and different orientations compared to gradients based on benign samples. To leverage this observation, the authors suggest to bound the gradient magnitudes and minimize the angular differences directly counteracting the previously mentioned effects. Concretely, the authors suggest to use DP-SGD during the training of the NNs which clips and perturbs the gradients.

On the Security of Randomized Defenses Against Adversarial Samples
Kumar Sharad, Giorgia Azzurra Marson, Hien Thi Thu Truong, Ghassan Karame in Asia CCS, 2020 [392], *Defense Methods*
The authors present a survey-like paper analyzing three randomness-based defense strategies. Two of the methods are presented in the papers Countering adversarial images using input transformations and Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification, respectively. The third method is called Randomized Squeezing and is introduced by the authors themselves. Their approach is partially based on the adversarial example detection method Feature Squeezing. Here, the modification approaches are performed at random to break the adversarial features forcing the potentially adversarial inputs to be classified correctly. After a thorough evaluation of all three defense methods the authors conclude that randomized defenses provide a certain level of robustness against black-box and grey-box attacks. Unfortunately, the authors find that randomized defenses cannot protect against white-box attacks and can be circumvented by adaptive attacks.

Overfitting in adversarially robust deep learning
Leslie Rice,Eric Wong,Zico Kolter in ICML, 2020 [359], *Defense Methods*
The authors analyze the overfitting in AT setups and find that classical PGD AT with early stopping performs as good as newer developments in AT (i.e., improvements made currently can also be attained by just early stopping).

Perceptual Adversarial Robustness: Defense Against Unseen Threat Models
Cassidy Laidlaw, Sahil Singla, Soheil Feizi in ICLR, 2021 [228], *Attacks on Deep Learning Systems*
The authors propose an AT approach that aims at robustness against all possible imperceptible attacks (threat models) (approximated with a DNN), even those unseen during training. Under the neural perceptual threat model, adversarial examples which are close w.r.t. neural network perception (using Learned Perceptual Image Patch Similarity as distance, approximating the space of imperceptible changes to humans the distance can be measured with the same or another, fixed network) but fool the network, are considered. Perceptual Adversarial Training (PAT) is then introduced, based on two attacks under this threat model (Perceptual Projected Gradient Descent - PPGD- and in particular under (Fast) Lagrangian Perceptual Attack - LPA). During AT, they do not project on the feasible set for computational time. Both attacks rely on the margin loss (C&W) and constrain on the perceptual distance. It is experimentally shown that the resulting model is robust against even unseen threat models, outperforming TRADES and other AT methods and that humans confirm the perceptual indistinguishability of the generated attacks.

PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples
Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, Nate Kushman in ICLR, 2018 [413], *Defense Methods*

The authors present their defense method called PixelDefend which consists of two parts: an adversarial example detection mechanism and a subsequent adversarial example purification. In order to detect adversarial examples the authors use statistical hypothesis testing and report modern neural density models to be usable in detecting imperceptible image perturbations. The concept is based on the ideathat adversarial examples mainly lie in the low-probability region of the data distribution. Once an adversarial example is detected PixelDefend tries to purify the inputs such that adversarial examples are transformed back to the benign training distribution. For this purpose, the authors use a probabilistic generative model calledPixelCNN to perform the preprocessing and purification step. The defense method was evaluated by Athalye et al. [16]. Here, the authors are able to bypass the defense usingBPDA. Even in combination with PGD-based adversarial training, PixelDefend does not provide robustness for NNs.

Playing the Game of Universal Adversarial Perturbations
Julien Perolat, Mateusz Malinowski, Bilal Piot, Olivier Pietquin in arXiv, 2018 [338], *Defense Methods*
Formulating adversarial training as a game-theoretic problem with two players, where the goal is to find the best response to the past strategies of the opponent (classifier vs. dataset manipulator). Robustness to universal adversarial examples as well as adversarial patches is considered.

Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases
Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, Meng Wang in ECCV, 2020 [458], *Defense Methods*
The authors propose a method against poisoning attacks with focus on Trojan attacks. Great care was taken to reduce the amount of data required for the detection to work. Two methods were proposed: 1) a data-limited detector, where only one example of each available class is needed, and 2) a data-free detector, which works based on the NN and random inputs only. The main intuitions are the following: 1) in presence of a Trojan shortcut, universal attacks and per-image attacks share strong similarities - doing both attacks and comparing the output activations leads to Trojan NNs 2) past research has motivated that Trojan NNs have unexpectedly high activation patterns - generating a perturbation that maximizes the activations and applying it to random inputs, we can see if the samples are pushed to similar target classes. Both detection methods are successful during the evaluation, but the paper lacks a discussion how attackers can circumvent the detection.

Privacy-Preserving Classification on Deep Neural Network
Herve Chabanne, Amaury de Wargny, Jonathan Milgram, Constance Morel, Emmanuel Prouff in IACR, 2017 [65], *Defense Methods*
The authors present an improved version of CryptoNets. The methods introduced in CryptoNets involved changes to the NNs even during training. In some cases instabilities were triggered. Furthermore, CryptoNets is only applicable for NNs with a maximum amount of two non-linear layers which restricts the application to simple data sets. To circumvent these downsides, the authors in this paper present the following improvements. Major changes to the NNs are added

during the inference. Hence, the NNs are still trained using the standard ReLU activation functions. Only for inference, the ReLU activations are replaced by low degree polynomials. Furthermore, during training, max pooling layers are replaced by average pooling layers and batch normalization layers are applied before the activation functions. With these measures, the authors preserve the stability of the training runs and hence achieve good accuracy during training. Then, during inference, the high level of accuracy is preserved while applying homomorphic encryption to enhance the privacy.

Privacy-Preserving Deep Learning
Reza Shokri, Vitaly Shmatikov in CCS, 2015 [398], *Defense Methods*
The authors present a collaborative training approach for NNs, which they call selective stochastic gradient descent (SSGD). The approach is designed for settings in which multiple, independent parties try to collaboratively learn on individual samples. Hence, the parties try to not share the training samples between each other to preserve privacy. To achieve this goal, the authors individually train the parties on their samples and then enforce the parties toasynchronously share a fraction of the computed gradients with each other. With this approach, the individual training samples remain private but the NNs can leverage the gradients of the other parties to further improve the local training process thus allowing a higher accuracy. To further protect the training, the authors additionally apply differential privacy to the parameter updates. The limitations of the approach are the fact that the authors assume IID data and the sharing of the gradients may leak information and thus allow extraction of local data samples. This attack vector was shown by a subsequently published paper [340].

Privacy-Preserving Deep Learning via Additively Homomorphic Encryption
Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai in IEEE Transactions on Information Forensics and Security, 2018 [340], *Defense Methods*
The authors leverage the findings presented in [398] to present an improved concept to collaboratively train NNs. First, the authors argue that in the previously shown method the process of sharing parts of the gradients between multiple NNs can be attacked. The shared gradients can be exploited to extract private data samples. Therefore, the authors propose to improve the system by using additive homomorphic encryption in the gradient sharing process. The authors encrypt the shared gradients with this approach, which allow subsequent computation and training using the encrypted information. The downside of this approach is the fact that the additional encryption and computation on encrypted data produces computational overhead.

Rademacher Complexity for Adversarially Robust Generalization
Dong Yin,Ramchandran Kannan,Peter Bartlett in ICML, 2019 [512], *Defense Methods*
Rademacher complexity in the adversarial setting is introduced and its connection to robust generalization is studied.

Recent Advances in Adversarial Training for Adversarial Robustness
Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, Qian Wang in arXiv, 2021 [23], *Defense Methods*

Good taxonomy and overview of adversarial training against l_p-norm restricted epsilon-perturbation based attacks.

Rethinking softmax cross-entropy loss for adversarial robustness
Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, Jun Zhu in ICLR, 2020 [325], *Defense Methods*
The authors present a new loss function used during training to make NNs more robust.Instead of using the softmax cross entropy for example, the authors use the Max-Mahalanobis center (MMC) loss. In their evaluation, the authors did not assume adaptive adversaries. Thus, in a later study presented by Tramer et al. [438], the proposed defense method was shown to be ineffective against such attackers. Tramer et al. simply used the newly presented loss function to optimize during the generation of adversarial examples.

Robust Local Features for Improving the Generalization of Adversarial Training
Chuanbiao Song, Kun He, Jiadong Lin, Liwei Wang, John E. Hopcroft in ICLR, 2020 [410], *Defense Methods*
The authors propose an approach called Robust Local Features for Adversarial Training (RLFAT), which can be added to existing AT methods. The key idea is to apply Random Block Shuffle (dividing the image into blocks and shuffling, first vertically, then splitting horizontally and shuffling again) on adversarial attack images during AT to break global features and thus to enforce the network to learn robust local features that generalize well to new data (instead of global ones which might not generalize). These features are then transferred to AT with normal adversarial examples by minimizing the distance between the learned robust local features and the normally extracted ones in the logit layer (this is added to the usual AT loss). These two steps are combined to form one AT framework.

Robustness to adversarial examples through an ensemble of specialists
Mahdieh Abbasi, Christian Gagne in ICLR (Workshop Track), 2017 [2], *Defense Methods*
This paper presents an ensemble method to protect NNs against adversarial examples. The authors observed that for typically used data sets, some subsets of the available classes are frequently confused by NNs either during attacks or due to the naturally occurring inaccuracy. This motivated the author to introduce so-called specialists to the overall decision process of the protected system. The specialist NNs are trained on the often confused classes and are thus capable of separatingthem with higherreliably. To estimate which classes need to be distributed among the experts, the authors perform untargeted attacks and combine the most confused classes accordingly. During test time, two scenarios are possible: (1) all specialistsand the original NN (generalist) suggest the same class for the processed input. Then this class is put out by the system. (2) the specialists and the generalist do not agree. Here a majority vote is used to determine the output. For both cases, if the average confidence among the voting classifiers is low, the input is considered adversarial and is hence rejected.

STRIP: a defence against trojan attacks on deep neural networks
Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C.Ranasinghe, Surya Nepal in ACSAC, 2019 [139], *Defense Methods*
The authors present a defense method against poisoning/trojan/backdoor attacks called STRIP. Similarly to previous work, the method is based on the observation that backdoored NNs behave differently when processing perturbed inputs. More specifically, inputs aiming at backdoor attacks are more robust to different perturbations compared to benign inputs. To leverage this effect, the authors introduce a new entropy measure, which quantifies the changes in the prediction by these perturbations.

Safetynet: Detecting and rejecting adversarial examples robustly
Zahra Ghodsi, Tianyu Gu, Siddharth Garg in NeurIPS, 2017 [146], *Defense Methods*
The authors present a defense method which is based on the detection of adversarial examples. The underlying intuition stems from the observation that adversarial examples produce different patterns of ReLU activations compared to benign samples. In order to leverage this observation, the authors append aRadial Basis Function SVM classifier to the NNs which uses thediscrete codes computed by the late stage ReLUs as input. In the training phase using adversarial and benign samples the SVM is trained to detect attacks. One drawback of this defense is the fact that it only works with ReLU activated NNs. Furthermore, no strong adaptive adversaries were considered during the evaluations. This may pose a challengeddue to the non-differentiable classifier at the decision stage.

Scalable Private Learning with PATE
Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, Ulfar Erlingsson in ICLR, 2018 [335], *Defense Methods*
The authors present an improved version of PATE [330] which is anprivacy preserving defense strategy. Compared to the original approach, the authors in this paper optimize the aggregation system and thus allow more complex settings and data sets. Furthermore, with the improved version of PATE, tighter DP guarantees can be made while profiting from a higher level of utility.

Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data
Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow, Kunal Talwar in ICLR, 2017 [330], *Defense Methods*
The authors introduce a privacy preserving defense strategy called Private Aggregation of Teacher Ensembles (PATE). The method is based on the concepts of differential privacy performing label perturbations. In their method, the authors introduce an ensemble of teacher models. Each teacher trains ondisjoint subsets of the sensitive data. In the aggregation phase, the student model trains on public data samples which are labelled using the previously trained teachers. With this approach, the student model is not directly trained on the sensitive training data. In the aggregation process,differential private noise is injected to the labels to further ensure privacy. In their evaluation the authors perform proof-of-concept experiments using simple data sets and settings.

SentiNet: Detecting Localized Universal Attacks Against Deep Learning Systems
Edward Chou, Florian Tramer, Giancarlo Pellegrino in IEEE Symposium on Security and Privacy Workshops, 2020 [85], *Defense Methods*
Chou et al. propose an adversarial example detection method with focus on localized universal attacks, i.e., adversarial patches. Although not tested on physical attacks, this paper is a step towards adversarial detection in real-world settings. The method combines a segmentation algorithm and a boundary analysis: parts of the image that have high influence on the output are identified, cut out and applied to a known test set. Intuitively, if these segments have high influence on the output in the test images as well, the identified patch may be of adversarial nature. Unfortunately, the method seems computationally inefficient and is highly dependent on a good segmentation.

Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression
Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, Duen Horng Chau in KDD, 2018 [100], *Defense Methods*
A defense framework SHIELD (Secure Heterogeneous Image Ensemble with Local Denoising) based on JPEG-compression is presented. As such, it is based on image preprocessing. A special training with compressed adversarial and clean images aims at increasing the robustness of the network to various (random compressions).

ShieldNets: Defending Against Adversarial Attacks Using Probabilistic Adversarial Robustness
Rajkumar Theagarajan, Ming Chen, Bir Bhanu, Jing Zhang in CVPR, 2019 [431], *Defense Methods*
The authors introduce a novel defense method against evasion attacks by removing parts of the adversarial perturbation. They introduce a probabilistic model called PixelCNN, which learns the joint probability between pixel values. Based on its objective, it tries to push adversarial images to regions of less attack success within its direct neighborhood. The very same process is repeated, resulting in several pictures passed to the original classifier. Based on the average of the output logits, the output prediction is determined. Unfortunately, adaptive attacks are not discussed, which rises questions on the real-world robustness gain.

Single-Step Adversarial Training With Dropout Scheduling
Vivek B.S., R. Venkatesh Babu in CVPR, 2020 [49], *Defense Methods*
An approach for adversarial training using the single-step attack method FGSM, (SADS: Single-step Adversarial training with Dropout Scheduling), that is supposed to be robust also to multi-step attacks is presented. Based on the observation that single-step AT methods usually lead to overfitting to these attacks and to gradient masking effects, the authors propose to include dropout after every non-linear layer and to decay the dropout rate with the training progress.

Spectral Signatures in Backdoor Attacks
Brandon Tran, Jerry Li, Aleksander Madry in NIPS, 2018 [442], *Defense Methods*

The authors present a defense method against poisoning/trojan/backdoor attacks using their so-called spectral signatures. The method is based on the intuition that the internal representation of trained classifiers amplifies signals useful during the classification process. Hence, by analyzing these internal representations, backdoors should become detectable. For this purpose, the authors use the internal representation for all samples of each class of the data separately and leverage recent findings from robust statistics tools to detect backdoors in the analyzed NNs.

Stateful Detection of Black-Box Adversarial Attacks
Steve Chen, Nicholas Carlini, David Wagner in SPAI (Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence at Asia-CCS), 2020 [76], *Defense Methods*
The authors present a defense method which detects query-based black-box attacks on NNs. For this purpose, the authors introduce the notion of stateful defenses. Prior defenses and detection methods handled each sample fed to the NN independently and thus stateless. To incorporate the knowledge of previous queries, the authors introduce a stateful detection scheme aggregating all queries performed by each user. If new queries performed by a user are too similar to previously executed ones, the defense method is triggered and assumes a query-based black box attack. For non-query-based black-box attacks, the authors suggest to combine their defense with adversarial training. During their adaptive attack evaluation, the authors introduce a new attack method called query blinding which is capable of successful hiding the queries and thus again allows adversarial example generation.

Stochastic Activation Pruning for Robust Adversarial Defense
Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, Anima Anandkumar in ICLR, 2018 [103], *Defense Methods*
Inspired by mixed strategies from game theory the authors present the defense method called Stochastic Activation Pruning (SAP). During test-time the authors randomly prune some of the activations observed in the NNs to protect. This is preferably done for the activations with smaller magnitudes. To balance the NNs, the remaining activations are rescaled to allow a more stable classification output. In the paper, the authors validate the approach using image data sets and perform a proof of concept in the field of reinforcement learning. SAP was later shown to be ineffective against adaptive attacks.

Strong Data Augmentation Sanitizes Poisoning and Backdoor Attacks Without an Accuracy Tradeoff
Eitan Borgnia, Valeriia Cherepanova,Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, Arjun Gupta in ICASSP, 2021 [40], *Defense Methods*
The authors propose a training data augmentation to regularize class boundaries and lower the impact of poisoning samples. Their method CutMix pastes pixels from one training sample into another while also merging the training labels. As result, poisoned training samples may lose some of their poisoned regions. Unfortunately, the weak evaluation does not motivate the method well enough.

TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems
Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, Dawn Song in arXiv, 2019 [158], *Defense Methods*
The authors present an improvement of the backdoor detection method Neural Cleanse, which they call TABOR. In their paper, the authors enhance the fidelity of the backdoor trigger reconstruction process by using heuristic regularization methods. With this approach, the authors argue that TABOR outperforms Neural Cleanse.

TAPAS: Tricks to Accelerate (encrypted) Prediction As a Service
Amartya Sanyal, Matt Kusner, Adria Gascon, Varun Kanade in ICML, 2018 [380], *Defense Methods*
The authors present their approach, called TAPAS, to accelerate fully homomorphic encryption in combination with NNs. For this purpose, the authors build upon previous findings and leverage the method called FHE-DiNN introduced in [43]. ForFHE-DiNN, the authors identify the following downsides and restrictions: Only modest accuracy values are achieved for the MNIST data set. The method parameters depend on the NN architecture. Finally, the processed data needs to be re-encrypted if the model is updated. To improve upon recent findings, the authors propose specialized circuits for layers typically used in NNs and additionally suggest to use binary quantized NNs.

TextShield: Robust Text Classification Based on Multimodal Embedding and Neural Machine Translation
Jinfeng Li, Tianyu Du, Shouling Ji, Rong Zhang, Quan Lu, Min Yang, Ting Wang in USENIX, 2020 [238], *Defense Methods*
The authors present TextShield which is a defense method fordeep learning-based text classification systems (DLTC) specialized for Chinese text only. Opposed to English texts, defense methods face some challenges when applied to the Chinese language for example due to the large amount of characters. TextShield consists of two parts: amultimodal embedding and neural machine translation (NMT) model and the DCTC model for text classifications. In the first step, the NMT model is trained with pairs of adversarial and benign inputs such that Chinese input texts are translated to English and back to Chinese. With this approach, the authors argue that a first form of text correction is performed and the adversarial perturbations are removed. In the second step, the cleaned inputs are fed to the DLTC model which extracts features (semantic, glyph, and phonetic-level features) used for the classification of the text. With this approach, the authors are able to correctly classify adversarial texts in a black-box setup. Note, that a white-box evaluation of the method was not performed and first evaluations of adaptive attacks suggest that the method may be vulnerable under this threat model.

The Odds are Odd: A Statistical Test for Detecting Adversarial Examples
Kevin Roth, Yannic Kilcher, Thomas Hofmann in ICML, 2019 [365], *Defense Methods*
The authors present an adversarial example detection method which is based on the assumption that adversarial examples are less robust to noise than their benign counterparts. Based on this

idea, the defense method first adds a random noise vector to a currently analyzed input sample before classification. Then the logits produced by the original input and the noisy input are compared to each other using a specifically designed distance metric. If the distance between the logits surpasses a predefined threshold the sample is considered adversarial and is hence rejected. Even though the authors perform a complete set of evaluations including an analysis of adaptive adversaries, in two later studies presented by Tramer et al. [438] and Hosseini et al. [178], the detection method was bypassed by further improved adaptive attacks. Using a combination of EoT and feature level attacks, adversarial examples can be generated which bypass the detector.

Theoretically Principled Trade-off between Robustness and Accuracy
Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, Michael I. Jordan in ICML, 2019 [528], *Defense Methods*
A defense that can trade off robustness and accuracy, TRADES, is introduced. It relies on decomposing the adversarial loss into a classification and boundary loss and providing an upper bound to it, leading to a new formulation for the adversarial training (TRadeoff-inspired Adversarial Defense via Surrogate-loss minimization).

Thermometer Encoding: One Hot Way To Resist Adversarial Examples
Jacob Buckman, Aurko Roy, Colin Raffel, Ian Goodfellow in ICLR, 2018 [50], *Defense Methods*
The authors present a defense method which is based on adversarial training closely related to the approach presented by Madry et al. [276]. Additionally to the PGD-based training approach, the authors introduce the notion of thermometer encoded models. This modified training process tries to break the linearity of the NNs which is assumed to be the reason adversarial examples exist. In an in-depth analysis presented by Athalye et al. [16], thermometer encoding is shown to provoke gradient obfuscation which can be bypassed by specifically crafted adaptive attacks using BPDA.

Thwarting adversarial examples: An l0-robust sparse Fourier transform
Mitali Bafna, Jack Murtagh, Nikhil Vyas in NeurIPS, 2018 [20], *Defense Methods*
The authors present adefense method which claims robustness against L0 based adversarial examples. To make NNs more robust the authors follow a purification approach such that the adversarial features are broken and the inputs can again be classified correctly. The preprocessing procedure which is based on theIterative Hard Thresholding approach consists of two steps performed over multiple iterations: First,inputs are compressed by projecting them to the top-kcoefficients of the discrete cosine transform. In more detail, the authors perform a Fourier transform to accomplish this. Second,the samples are then inverted to recover approximate images which are classified by the NNs. In a later evaluation by Tramer et al. [438], the defense method was shown to be ineffective against adaptive attacks directly using the L0-version of the C&W method.

Towards Certifiable Adversarial Sample Detection
Ilia Shumailov, Yiren Zhao, Robert Mullins, Ross Anderson in AISec, 2020 [401], *Defense Methods*
The authors present a defense method which tries to provide a certified adversarial example detection scheme, called CCT. CCT is based on the previously introduced Taboo Trap detection scheme which itself is based on the analysis of the activation values in NNs triggered by various inputs. An attack is detected if the activation values a driven beyond a predefined range. The authors extend Taboo Trap and present three detection modes. In the most strictly acting mode, a certified detection of adversarial examples is achieved. In their evaluation, the authors do not perform adaptive attacks. Hence, it is not clear, if the defense is robust towards attackers which are fully aware of the detection scheme.

Towards Deep Learning Models Resistant to Adversarial Attacks
Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu in ICLR, 2018 [276], *Defense Methods*
The paper provides the projected gradient descent (PGD - iterative attack) attack description, and a discussion about obtaining max of the loss function through random runs of PGD. Moreover, the authors discuss how to employ this maximum in the adversarial training framework.

Towards Interpretable Deep Neural Networks by Leveraging Adversarial Examples
Yinpeng Dong, Fan Bao, Hang Su, Jun Zhu in AAAI Workshop on Interpretability for Deep Learning, 2019 [106], *Defense Methods*
An adversarial training method that aims at consistency of neurons (firing neurons unambiguously) is presented. This is done by considering a consistency loss (feature-matching) which aims at ensuring that the features of the worst-case adversarial example are aligned with the clean data point. It is assumed that the adv. examples used can be an approximation for the worst-case one. The authors employ FGSM for adv. example generating during adv. training.

Towards Robust Detection of Adversarial Examples
Tianyu Pang, Chao Du, Yinpeng Dong, Jun Zhu in NeurIPS, 2018 [324], *Defense Methods*
The authors propose a modified training procedure for classifier NNs, which allows to detect attacks more easily. A modified loss function, the reverse cross-entropy together with a label smoothing regularizer, favors uniformly distributed non-maximum classes. As result, inputs of normal classes have a more compact latent representation, requiring more severe adversarial perturbations for successful attacks. The detection is then based on a Gaussian kernel estimation for each output class. Drawbacks of this method are the extra set of hyperparameters, the unclear non-attack performance and the limited usefulness to existing models.

Towards Robust Neural Networks via Random Self-ensemble
Xuanqing Liu, Minhao Cheng, Huan Zhang, Cho-Jui Hsieh in ECCV, 2018 [255], *Defense Methods*
The authors present the approach ofrandom self-ensembles (RSE) which combines the strategies of introducing randomized processes to NNs and assembling strategies. The approach can be counted to the category of defenses which introduce changes to the NNs-to-protect. RSE

adds a noise layer before each convolution layer in both, the training and testing phase. During test-time, the model performs multiple forward-passes such that RSE ensembles the prediction results over the randomized runs to stabilize the models outputs. This approach is applicable in the Image for neural networks using convolutional layers. The authors state that they used an adapted C&W attack. Here, the attack is adapted such that the randomization procedure is included in the C&W-function. Furthermore, the authors followed the guideline on breaking ensemble-based defenses provided in [15]. Still, it remains unclear how robust this defense really is. For example, gradient-free attacks were not tested in this paper.

UnMask: Adversarial Detection and Defense Through Robust Feature Alignment
Scott Freitas, Shang-Tse Chen, Zijie J. Wang, Duen Horng Chau in IEEE Big Data, 2020 [133], *Defense Methods*
UnMask introduces a combined detection and defense framework against adversarial attacks by considering robust features in images. These robust features, e.g. parts of an object like a tire of a car, are harder to attack since the entire semantic context must be faked. As defense method, the authors propose mapping the input to its robust features and then comparing these with a precomputed table of expected features for each class. Despite the intuitive principle, the performance may severely depend on the quality of the robust features - thus, UnMask may not be universally applicable, nor was tested under adaptive attacks.

Understanding and Improving Fast Adversarial Training
Fung, Clement, Chris JM Yoon, Ivan Beschastnikh in NeurIPS, 2020 [135], *Defense Methods*
The authors show that previously proposed solutions for fast adversarial training, i.e., [472] and [388] do not solve the problem of catastrophic overfitting (as they claim to), especially for large perturbation values epsilon. This works introduces GradAlign as a regularization method to solve the problem and successfully apply it to FGSM adversarial training, leading to good results and reducing overfitting.

Understanding catastrophic overfitting in single-step adversarial training
Hoki Kim, Woojin Lee, Jaewook Lee in to appear in AAAI, 2021 [218], *Defense Methods*
In this paper the authors analyze the catastrophic overfitting that is observed in single-step adversarial training (fast AT approaches that rely on variants of FGSM attacks). This overfitting problem which leads to decreased robustness to PGD attack, is linked to a distorted decision boundary in the neighborhoods of the adversarial examples as a consequence of fixed distances of the adversarial images to the original one. Based on this observation, the authors propose a stable single-step adversarial training, where a scaling parameter is added to the perturbation. This scaling parameter is determined as the minimal value (between 0 and 1) needed to switch the classifier decision (found by forward propagation of several distorted images -checkpoints).

Universal Adversarial Training
Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S. Davis, Tom Goldstein in AAAI, 2020 [389], *Defense Methods*

In the paper authors propose a new method for generating universal adversarial perturbations that is based on clipping the cross entropy loss and maximizing it over all inputs. Further they demonstrate a minimax objection and the optimization technique for it in order to perform adversarial training.

Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs
Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, Heiko Hoffmann in CVPR, 2020 [221], *Defense Methods*
The authors propose a detection method for backdoor attacks. They do so by concurrently optimizing a detector and the input to the detector - in other words, they design input samples that carry the optimal information content to detect backdoored NNs. They show that their detection generalizes to other trigger samples and NN architectures within the same classifier family. Drawbacks of this method are the resource demand by generating a training set of benign and Trojan NNs.

Unlabeled Data Improves Adversarial Robustness
Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, John C. Duchi in NeurIPS, 2019 [64], *Defense Methods*
An independently developed approach to leverage unlabeled data for improvement of AT is presented by Uesato et al. [449]. A semi-supervised robust training approach (RST) (self-training, i.e., pseudo-labeling the unlabeled data and performing supervised training on that) is introduced that is shown to be competitive in adversarial robustness with approaches that use many more labeled examples)

Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty
Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, Dawn Song in NeurIPS, 2019 [168], *Defense Methods*
The authors analyze the benefits of self-supervised training on model robustness, in particular to adversarial robustness. They find that adding a self-supervised loss term (that predicts rotation of the image - with an auxiliary head on the network) to the PGD-AT loss improves robustness. The effect on standard corruptions (blur, label corruptions) and out-of-distribution detection is analyzed as well.

What Doesnt Kill You Makes You Robust(er): Adversarial Training against Poisons and Backdoors
Jonas Geiping, Liam Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, Tom Goldstein in arXiv, 2021 [143], *Defense Methods*
The authors adapt AT to protect against (training time) poison and backdoor attacks. A surrogate attack model is used that has access to the training setup, architecture and defense but cannot change training. Robustness is considered against epsilon perturbations in l_p-norm. The basic idea is to separate each mini-batch randomly into poison and target data. Then malicious labels get drawn for the target data and a poisoning attack is applied to the poison data such that an

attack on the target data will result in classifying it as the malicious label. Finally, the batch is concatenated and used as training data.

When Explainability Meets Adversarial Learning: Detecting Adversarial Examples using SHAP Signatures
Gil Fidel, Ron Bitton, Asaf Shabtai in IJCNN, 2020 [131], *Defense Methods*
The paper combines adversarial ML with explainable AI, more precisely SHAP (Shapley Additive Explanations). SHAP is a method to determine the importance of input features on the given output. The authors analyze the SHAP values on the penultimate layer of benign and adversarial inputs using a binary classifier, which learns to detect attacks. Intuitively. the SHAP values of weak features tend to change more between benign and adversarial samples. Downsides of this paper are the weak evaluation and the missing adaptive attacks.

You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle
Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, Bin Dong in NeurIPS, 2019 [521], *Defense Methods*
The authors propose a speed-up for adversarial training by only taking the first layer as parameters and freezing the rest during the adversary update (reducing the amount of fwd. and backward passes needed and saving runtime (YOPO algorithm)). This is based on a differential game formulation (optimal control) of AT together with Potryagins Maximum Principle (PMP), which leads to the observation that effectively only the first layer is connected to the adversarial perturbation.

2.2.4 Information Extraction

A framework for the extraction of deep neural networks by leveraging public data
Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, Vinod Ganapathy in arXiv, 2019 [318], *Information Extraction*
A common problem in model extraction is selecting the samples with which the adversary queries the victims model. This paper demonstrates that using public datasets, also referred to as universal thief datasets, work better than using uniform noise. ImageNet is used for image based models and Wikipedia article dataset for NLP models as the universal thief datasets. Active learning is also proposed to select the samples from these datasets so that the query budget can be minimized. An ensemble of K-center strategy to maximize diversity and adversarial strategy to select informative samples is used to improve the attack performance.

A survey of privacy attacks in machine learning
Rigaki, Maria, Sebastian Garcia in arXiv, 2020 [360], *Information Extraction*
This survey presents an accurate classification of privacy attacks on the ML models. Furthermore, the survey presents a taxonomy and descriptions of the main approaches presented by the individual papers. An overview of the categrories and specific attacks can be found in Table 1. The authors suggest to split the attacks into the following categories: Membership Inference

Attacks, Reconstruction Attacks, Property Inference Attacks, and Model Extraction Attacks. Finally, the authors show a list of potential defenses against each attack category. For defenses against Membership Inference Attacks the authors further divide the defenses in the following classes: Differential Privacy, Regularization, and Prediction vector tampering.

ActiveThief: Model Extraction Using Active Learning and Unannotated Public Data
Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, Vinod Ganapathy in AAAI, 2020 [319], *Information Extraction*
Replicates experimental setup with a change in active learning algorithm from [318]. Random selection strategy used to select a uniform subset of samples to query the victim model. Greedy K-center algorithm along with DeepFool based active learning used for subset selection on the approximately labeled examples from the substitute model which are used to query the victim model for better training samples.

Auditing data provenance in text-generation models
Congzheng Song, Vitaly Shmatikov in ACM SIGKDD, 2019 [412], *Information Extraction*
The paper proposes to use shadow models trained on the users data to audit whether the target public model (RNN-based) used the (proprietary user) data for training. This is performed as a black-box evaluation, meaning that details about the model under investigation are not known and it can only be queried. Good performing models that are trained on large datasets (from many users) are analyzed.

Beyond inferring class representatives: User-level privacy leakage from federated learning
Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, Hairong Qi in IEEE INFOCOM, 2019 [464], *Information Extraction*
The paper proposes to infer the local training data of the nodes in the federated learning setup via training a multi-task GAN on the side of malicious central server. The GAN not only learns to restore data, but also identify the victim-node - if data belongs to it. Passive mode (when only data is obtained) and active mode (when the victim model updates are isolated) of the attack are proposed. The data is reconstructed to one particular node, using its model as discriminator and training a generator that will be generating training data of the victim node after training.

Black-Box Ripper: Copying black-box models using generative evolutionary algorithms.
Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, Marius Popescu in NeurIPS, 2020 [27], *Information Extraction*
In order to steal the functionality of a black box model, this paper proposes a framework based on two training phases. The framework is based on zero-shot knowledge distillation methods where the teacher model is the black box model that is attacked. In the first phase, a generative model, e.g. a Variational Auto-Encoder or a Generative Adversarial Network is trained on a proxy data set and the training is independent of the student model. The generated samples are labelled by the teacher/black-box model via API queries. Since the generator is trained to model the probability density , the data samples are likely not representative for any class in the true

data set. This, the teacher is likely not going to produce a high probability for a certain class. To this end, an evolutionary strategy is used in the second step of the framework which modifies the generated data samples such that they exhibit a high response for a certain class when given as input to the teacher.

CSI NN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel
Lejla Batina, Shivam Bhasin, Dirmanto Jap, Stjepan Picek in USENIX, 2019 [30], *Information Extraction*
The paper presents a side-channel attack to extract neural network properties such as activation functions, weights and number of hidden layers and neurons per layer. The attack bases on timing information and on electromagnetic emanation observed by an adversary. This side-channel information is then correlated to neural network computations. For instance, activation functions are recovered by exploiting that different activation functions take different time to compute. Another example is the extraction of weights by using statistical tests to identify the most likely computed matrix multiplications based on the power consumption. Experiments demonstrate the effectiveness of the side-channel attacks on different network architectures and micro-controllers. The paper also mentions defense mechanisms such as random permutation of neuron computations, but these defenses are generic to any side-channel attack and typically have significant computational cost.

Cache Telepathy: Leveraging Shared Resource Attacks to Learn DNN Architectures
Mengjia Yan, Christopher W. Fletcher, Josep Torrellas in USENIX, 2020 [498], *Information Extraction*
This paper proposes an approach called Cache Telepathy that extracts various hyperparameter values, such as the number of layers and activation functions, for fully-connected and convolutional networks, based on a cache side-channel attack. Cache Telepathy exploits that inference in neural networks relies on matrix multiplication, and the shape of the matrices and their multiplications rely on optimized CPU caching. Since caching behavior provides evidence for the characteristics of the matrix multiplication, one can establish a mapping between cache behavior and the choice of hyperparameter values of a neural network. The paper derives such a mapping by observing execution times and code invocation counts with established methods like PrimeProbe and FlushReload. This allows to derive even complex properties of a network, such as whether a layer is connected sequentially or with a shortcut connection. Empirical results show that Cache Telepathy can reduce the number of candidate architectures when running model extraction on a victim model from an intractable amount to only a few hundred candidates.

CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples
Honggang Yu, Kaichen Yang, Teng Zhang, Yun-Yun Tsai, Tsung-Yi Ho, Yier Jin in NDSS, 2020 [514], *Information Extraction*

The authors rely on adversarial attacks to fine-tune the substitute model and to minimize the query budget to carry out a more efficient fidelity extraction. The substitute model is first trained on a dataset built with query outputs to randomly chosen query inputs. A new active learning technique called margin based uncertainty which boosts examples where the victim model is least confident is also proposed to craft informative samples. Then this substitute model is used to craft adversarial examples using a new technique called FeatureFool. The adversarial examples are then used to query the victim model again and a new synthetic dataset is created with the resulting query outputs. This synthetic dataset is used to retrain the last few layers of the substitute model. FeatureFool uses a new loss function which uses a distance measure of two images to find if they have similar inner feature mappings and a parameter to optimize this objective along with the box constraint of L-BFGS algorithm. The attack is also tested on five MLaas providers including Microsoft, Face, IBM, Google and Clarifai. The authors upload different trained models to these services and simulate the attack and show that highly effective substitute models can be constructed with very small budgets in $.

Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacksagainst Centralized and Federated Learning
Milad Nasr, Reza Shokri, Amir Houmansadr in EuroS&P, 2019 [311], *Information Extraction*
Analysis of the possibility to infer the membership of data points in white box scenario is presented. The attack can be done during training in a one-node or also in a federated learning setup and can be performed through active manipulation as well as passive output observations. The main idea is to make use of problems with stochastic gradient descent (each datapoint influences it) and to use the gradient of the model when inputting the target data. Then, a probability for membership is computed. The method can be used for unsupervised attacks (i.e., when no training data is known) as well as for supervised ones. The attack is even more dangerous in the federated learning setup, as more information can be obtained from the gradients.

Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data

Jacson Rodrigues Correia-Silva, Rodrigo F. Berriel, Claudine Badue, Alberto F. de Souza, Thiago Oliveira-Santos in IJCNN, 2018 [91], *Information Extraction*
The article is an experimental study on extracting a victim image classification model with varying knowledge of the training data domain. The study considers an attacker that obtains labels from a public model API to train a surrogate model with i) access to random images without knowledge of the genuine problem domain, ii) access to a small subset of the genuine problem domain, or iii) access to the genuine problem domain only for fine-tuning a model that has been pre-trained on random images. Results on different benchmark data sets show that the extracted models yield accuracies close to the victim model even without knowledge of the genuine problem domain. Access to data from the problem domain can result in higher classification accuracy, albeit improvements on the benchmark data are not significant.

Cryptanalytic Extraction of Neural Network Models
Nicholas Carlini, Matthew Jagielski, Ilya Mironov in CRYPTO, 2020 [58], *Information Extraction*
The paper frames the extraction of neural network parameters as a chosen-plaintext attack, a cryptographic attack to recover a function only based on selected input-output pairs. Albeit the analogy, there are challenges that are specific to neural network extraction such as fixed- and floating-point arithmetic in neural networks compared cryptography which generally relies on finite fields. Based on this theoretical framework, the paper proposes an attack of extracting weights based on well-selected queries similar to ReLU hyperplane recovery methods [288], but with an extension to deep networks. Experiments show that the proposed method indeed recovers neural networks more accurately than existing work while requiring less queries to the victim model.

Deep leakage from gradients
Ligeng Zhu, Zhijian Liu, Song Han in NeurIPS, 2019 [546], *Information Extraction*
The authors propose a technique for reconstructing both input and labels based on the leaked gradients during distributed training. They are optimizing the initial random input and label with respect to the difference between gradients and can reconstruct everything. Furthermore, the authors present a defense strategy against their attack method based on a few approaches. First, they try adding noise to the gradients, which turns out to be sensitive to the noise level and thus not always successful (or leads to degraded network performance). Next, they use gradient compression or sparsification in a sense that gradients that have a small magnitude (smaller than some predefined threshold) get pruned to zero. In their experiments, the authors report robustness against privacy evading attacks with 20% of the gradients set to zero. Furthermore, the performance of the model was not reduced significantly, so that pruning can be an effective defense. In case the training setup can be altered, changing the batch size, image resolution and encrypting gradients can be helpful.

Deep models under the GAN: information leakage from collaborative deep learning
Hitaj, Briland, Giuseppe Ateniese, Fernando Perez-Cruz in ACM SIGSAC, 2017 [172], *Information Extraction*
The paper proposes to construct a GAN network for an adversary who is integrated in a federated setup with the discriminator being the communicated model (white box access). Then generator will learn to recreate the data that belongs to the victim (with some particular label that the adversary does not have). In particular, the attacker is part of the collaborative setup (i.e. a normal node) and can, by manipulating the training process, make other nodes (victims) share sensitive information. The authors point out that their attack is also effective on CNNs and can bypass setups with differential privacy.

Deepsniffer: A dnn model extraction framework based on learning architectural hints
Xing Hu, Ling Liang, Shuangchen Li, Lei Deng, Pengfei Zuo, Yu Ji, Xinfeng Xie, Yufei Ding, Chang Liu, Timothy Sherwood, Yuan Xie in International Conference on Architectural Support for Programming Languages and Operating Systems, 2020 [185], *Information Extraction*

A deep neural network can be viewed as a computational graph of different layers. These are executed in runtime on hardware primitives as kernel sequences which leak information about the model architecture. In this threat model, the adversary has physical access to the hardware platform and two attack scenarios are considered: in the EM side channel attack, the adversary has access to the read& write memory while in the bus snooping attack, the adversary can gather the memory access traces. In these scenarios, kernel execution time to infer the input/output data volume and the distance between dependent kernels is gathered. This is used by two correlation models to infer architectural hints about the victims model. The first model correlates the kernel information received as tuples of architectural hints while the second model uses the premise that models are often follow a certain logic with regards to the sequence of layers. Convolutional layers are often followed by normalization, then non-linear layers and so on. The dimensions of the layers can also be extracted using the read/write volumes as these indicate the size of the feature maps passing from one layer to the next. Experiments used pytorch models using CUDA optimization of a Nvidia K40 GPU.

Demystifying membership inference attacks in Machine Learning as a Service
Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, Wenqi Wei in IEEE Transactions on Services Computing , 2019 [443], *Information Extraction*
The authors investigate membership inference attacks and propose a technique where a shadow dataset is used to train an attack inference model. This attack model is later used to understand whether the data point was inside of the training data or not. They consider a black-box setup where an attacker can only query the target model. The shadow training dataset can be generated by different strategies, such as statistics or active learning-based methods, etc. Then the attacker creates the dataset used for the attack model training, i.e., a binary model that will distinguish between examples which were part of the training and samples which were not. Furthermore, with their experiments the authors investigate the transferability between datasets, success of the attack for different models, and the applicability of the attack in the federated learning setup with insider knowledge. Finally, the authors evaluate a set of known defense methods to assess the success of their proposed attack for hardened target models. In particular, the authors test the following four defense strategies: dimension reduction, regularization, adversarial regularization, and differential privacy. For all applied mitigation techniques, the attack accuracy could not be reduced below 50.8% (for differential privacy with ϵ=1). Furthermore, for well performing defense strategies, the accuracy scores of the model were significantly reduced, limiting the utility of systems.

ES Attack: Model Stealing against Deep Neural Networks without Data Hurdles
Xiaoyong Yuan, Lei Ding, Lan Zhang, Xiaolin Li, Dapeng Wu in arXiv, 2020 [515], *Information Extraction*
The ES Attack starts with a synthetic dataset which is used to query the model and trains a substitute model on these input-output pairs. To refine the model extraction and data extraction, this is done iteratively by generating new synthetic samples from the substitute model and retraining.

Exploiting unintended feature leakage in collaborative learning
Luca Melis, Congzheng Song, Emiliano De Cristofaro, Vitaly Shmatikov in S&P, 2019 [282], *Information Extraction*
The paper considers the collaborative learning setup, where members are performing together gradient updates or federated setup with aggregation. They show a possibility for property of local data inference. The problem the authors tackle is the inference of the particular properties of the inputs, that might be not correlated with the label at all. The models can leak information about properties from the embedding layer (text models) and from the gradients itself. The attacker can be passive (only getting information) or active (changing the gradients in a way, that target models after update would leak more information). There are particular assumptions that are done: that the attacker has a dataset with labeled property of interest, that the amount of collaborating models is not high, that the property in general is inferable from the data.

Exploring Connections Between Active Learning and Model Extraction
Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, Songbai Yan in USENIX, 2020 [66], *Defense Methods*
This paper presents a formalization of model extraction and draws parallels between model extraction and active learning. The mathematical formalization is useful for designing effective defenses.

Extraction of complex dnn models: Real threat or boogeyman
Buse Gul Atli, Sebastian Szyller, Mika Juuti, Samuel Marchal, N. Asokan in International Workshop on Engineering Dependable and Secure Machine Learning Systems, 2020 [18], *Information Extraction*
Knockoff Nets [317] is empirically evaluated for 5 complex DNN architectures. A defense strategy is also presented to detect Knockoff nets by differentiating in- and out-of-distribution queries (attackers queries). This defense correctly detects up to 99% of adversarial queries. Despite this, the authors say a strong adversary can carry out a model extraction attack even in a realistic scenario by evading the existing defenses. They propose watermarking and fingerprinting to be studied more to reduce the incentives of carrying out model extraction attacks.

GAN-leaks: A taxonomy of membership inference attacks against generative models
Dingfan Chen, Ning Yu, Yang Zhang, Mario Fritz in 2020 ACM SIGSAC , 2020 [70], *Information Extraction*
The authors propose a detailed taxonomy of MIA for generative models (as opposed to discriminative models for classification). Generative models are GANs and VAEs. The logic of attack is to approximate the distribution of the generated examples and see if the checked example is fitting into the distribution.The distinguished modes of black-box/white-box are 1) availability of discriminator 2) availability of generator 3) availability of the latent code 4) full black-box access. The technique used to estimate the probability of an example to be in the training set is Parzen window, which effectively measures the distance between generated output and checked

instance. Distance is defined as combination of L_2, image similarity and regularization. The authors also propose a calibration error with respect to harder and easier instances - they conclude that the quality of reconstruction is dependent on the easiness of the instance representation. As a defense they propose differential private SGD.

Good Artists Copy, Great Artists Steal: Model Extraction Attacks Against Image Translation Generative Adversarial Networks
Sebastian Szyller, Vasisht Duddu, Tommi Grondahl, N. Asokan in arXiv, 2021 [425], *Information Extraction*
The paper proposes an attack to extract GANs for image style transfer. In the attack, the adversary queries the victim model and creates pairs of original and style-transferred images. This makes up the training data for creating the surrogate model. In this scenario, the adversary has the advantage that the training data consists of pairs of original inputs and style-transferred outputs. This is different to the victim model that only has access to the original inputs and a set of target-style images. Thus, the adversary does not require knowledge of the victim model architecture and instead can use paired image-to-image translation models, i.e., where the original and style-transferred image both are part of the training data. Experiments show that the approach is successful on three different tasks: Monet painting style to photo transfer, anime style to selfie transfer, and creating high-resolution images from low-resolution ones. In some cases, the surrogate is successful in producing correctly styled images, although the specific outcomes differ. A user study confirms the finding that the extracted models indeed are successful in creating images that are equivalent to the ones obtained from the victim model.

Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers.
Giuseppe Ateniese, Giovanni Felici, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali in International Journal of Security and Networks 10.3, 2015 [14], *Information Extraction*
The paper proposes to train a metaclassifier that allows to understand if the dataset, used for the training of the target model had some particular property or not. For the approach one should represent the target model with a vector of parameters and also find a way to generate the datasets with and without the property of interest.

Hermes Attack: Steal DNN Models with Lossless Inference Accuracy
Yuankun Zhu, Yueqiang Cheng, Husheng Zhou, Yantao Lu in USENIX, 2021 [548], *Information Extraction*
The paper introduces a side-channel attack Hermes to extract neural networks from unencrypted PCIe traffic between CPU and GPU. The idea is to infer a graph of GPU kernel launches and data movements from noisy PCIe traffic. This is challenging, since some information, e.g., the layer type, is not explicit in the GPU kernel launches. Closed-source and undocumented hardware and drivers complicate this extraction attack further. Hermes overcomes these challenges in two phases. In the first offline phase, Hermes profiles PCIe traffic for known white-box models to build a knowledge base to identify non-relevant packages, a mapping between instruction

sets and neural network layers as well as package offsets to locate hyperparameter values within kernel launch instructions. During the second online phase, Hermes observes the traffic of an inference and uses the knowledge base built in the offline phase to reconstruct the model architecture. Experiments show that Hermes can fully recover benchmark neural networks up to the elements that are not relevant during inference, e.g., dropout layers.

High Accuracy and High Fidelity Extraction of Neural Networks
Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, Nicolas Paperno in USENIX, 2020 [198], *Information Extraction*
Evaluating model extraction falls into two categories: accuracy of the extracted model and fidelity of the extracted model with the victim model. The paper focus is on fidelity and makes both formal contributions on the hardness of extracting functionally equivalent models and algorithmic contributions to extract functionally equivalent models for two-layer networks. On the theoretical side, the paper shows that the number of queries required to extract a functionally equivalent model is exponential in the input size. Further, the equivalence test for two networks is shown to be NP-hard. The non-determinism of models training further limits how close an extracted model can come to the victim model.On the algorithmic side, the paper first investigates how semi-supervised training and active learning methods can improve model extraction. Results suggest that these methods can indeed reduce the number of labels required to achieve good accuracy of the extracted model. A second algorithmic contribution is the proposal of a novel extraction attack. The basic idea is to exploit piece-wise linearity of ReLU activated networks to search critical points where the gradient of a neuron changes from 0 to 1 by varying the queries to the victim model. Based on the critical points, one can derive the full weight matrix of the network. Experiments show that a combination of the critical point method with learning-based model extraction results in extracted models with high fidelity, even for large models with more than 400,000 parameters.

How to 0wn NAS in your spare time
Sanghyun Hong, Michael Davinroy, Yigitcan Kaya, Dana Dachman-Soled, Tudor Dumitras in arXiv, 2020 [176], *Information Extraction*
The threat model used in this paper involves an adversary who has a VM co-located on the same host machine as the container in which the victims model is deployed. This allows the adversary to access the last layer cache (which is shared between all users of a particular machine) and carry out a cache side-channel attack such as FlushReload. Only a single trace of the victims system calls are required to build a mapping between various architectural properties and the time required to execute, which generates a set of candidate computational graphs of the victims model. Invariant rules of deep learning and the corresponding computation times is used to eliminate the unlikely candidates and hence narrow the search space.

Knockoff nets: Stealing functionality of black-box models
Tribhuvanesh Orekondy, Bernt Schiele, Mario Fritz in CVPR, 2019 [317], *Information Extraction*

The article proposes a method to create a model that is functionally equivalent to a victim black-box based on random queries. The random queries are selected from an arbitrary sample distribution, a discrete set of images, by two different sampling strategies. The first strategy is uniform random sampling. The second strategy is adaptive and updates the sampling distribution based on query results. The idea is to reward queries that either explore the sample space or focus areas where both the extracted model is performing poorly and the victim model is confident. Experiments show that a good weighting of exploration and exploitation depends on how much the sample distribution overlaps with the victims training distribution. The quality of the extracted model further depends on the model architecture. To this end, experiments indicate that extracted models with complex architectures are better than the ones with compact architectures. In benchmark experiments, the proposed methods achieve 0.84-0.97 of the victim model accuracy if the sample distribution is equivalent to the victim training data, and 0.81-0.96 if there is no overlap.

LOGAN: Membership Inference Attacks Against Generative Models.
Jamie Hayes, Luca Melis, George Danezis, Emiliano De Cristofaro in PoPETs (Proceedings on Privacy Enhancing Technologies), 2019 [161], *Information Extraction*
The authors present new membership inference attacks against generative models. The goal is to assess if specific samples were part of the training process of the GANs under attack. The idea of the authors is to provide the data that is suspected to be in the training set of the model to the discriminator of the GAN and check the probabilities assigned. In the case of black box attacks, they train a local discriminator using examples generated by the generator of the targeted GAN. Black-box setup means that the attacker does not have access to the internal parameters and only can make queries to the model, while in white-box setup there is an access to the parameters. In the white-box setup the discriminator of the target model is used, in the black-box setup it is trained locally. Also, the authors consider a case when some auxiliary knowledge of the dataset is available to the attacker. In this case this data can be used to improve an attack. To further evaluate their new attack method, the authors test two common defense methods protecting the privacy of models. Namely, the authors use regularization methods and concepts from differential privacy. As privacy attacks are more successful for overfitting models, the authors use weight normalization and dropout layers which try to prevent the overfitting effect. The authors find that this method is not effective against their attack and furthermore decreases the natural performance of the protected GAN. Similarly, for the differential privacy defense in which Gaussian noise is added in the training of the discriminator, the privacy of the models cannot be preserved.

ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models.
Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, Yang Zhang in arXiv, 2021 [263], *Information Extraction*
The paper is an experimental study on comparing membership inference, model inversion, attribute inference and model stealing methods on different data sets and model architectures.

Results show that the complexity of a data set is relevant to the attack success: membership inference, for instance, benefits from complex training data model stealing becomes more difficult. The paper hypothesizes that this is because complex data may lead to overfitting which in turn eases membership inference but makes synthesizing queries to train an accurate surrogate model difficult. Experiments show that defenses such as Knowledge Distillation and Differential Privacy are only effective against some of the attacks and also may reduce the accuracy of the victim model.

ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models
Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, Michael Backes in NDSS, 2019 [377], *Information Extraction*
The paper proposes to make membership inference attacks that do not require knowledge about data, target model. They propose to train the attack model on different data and show that the attack is still possible. Also they propose to use prediction statistics (like entropy) when no shadow model is trained. Furthermore, to protect models against membership inference attacks, the authors present a first line of defense. They argue that overfitting is a major reason why membership inference attacks work. Therefore, the authors propose to use drop-out layers, which is the standard DL-approach to prevent overfitting. During training, a fixed number of neurons are randomly dropped. Additionally, the authors present an ensemble technique called model stacking, which also prevents overfitting but is applicable for standard ML applications.

Machine learning models that remember too much
Song, Congzheng, Thomas Ristenpart, Vitaly Shmatikov in ACM SIGSAC, 2017 [409], *Information Extraction*
The authors consider the attack when a victim is using training algorithm provided by malicious user. The training algorithm can be modified in such a way that later it is easy to extract training data from the model. Techniques are based on encoding the dataset information in the attributes or augmenting dataset with the artificial examples that leak needed information.

Membership Inference Attack against Differentially Private Deep Learning Model
Md Atiqur Rahman, Tanzila Rahman, Robert Laganiere, Noman Mohammed, Yang Wang in Trans. Data Priv. 11.1, 2018 [352], *Information Extraction*
The authors investigate how differentially private models (with differential privacy applied during training) are protected against membership inference attacks (performed with shadow models). Their study analyzes the white box case, i.e., when the parameters of a model can be accessed. They find that although differentially private models are more resistant against such attacks, there is a tradeoff w.r.t. performance. Moreover, the effect of the privacy parameter on the degree of protection is studied.

Model Extraction Attacks on Graph Neural Networks: Taxonomy and Realization
Bang Wu, Xiangwen Yang, Shirui Pan, Xingliang Yuan in arXiv, 2020 [474], *Information Extraction*

The article proposes a framework to extract graph neural networks. The challenge of extracting a model in this domain is that input data might only be partially known, e.g., if an adversary can observe some nodes and connections of a social network but does not have access to the entire graph. The paper introduces three categories of adversary knowledge: i) a set of nodes with attribute values ii) knowledge on the connections between nodes and iii) knowledge of a subgraph of the same domain as the one used to train the target model. An adversary can use the knowledge to generate inputs to get input output pairs to train a surrogate model on. If an adversary does not have knowledge in one category, they can synthesize inputs, e.g., the attribute values can be approximated by those of neighboring nodes. Experiments show that the quality of the surrogate generally improves with increasing adversary knowledge.

Model Extraction and Adversarial Transferability, Your BERT is Vulnerable
Xuanli He, Lingjuan Lyu, Qiongkai Xu, Lichao Sun in arXiv, 2021 [166], *Information Extraction*
This paper demonstrates a model extraction attack on a BERT based API in a black box scenario. After extraction of the model, different distributions are used to build transfer data sets. The authors show that the produced surrogate model can be used to generate well-working adversarial examples to attack the original victim model. To defend against the initial extraction attacks, the authors also propose two defenses which manipulate the output of the model. The first defense adds a coefficient to the softmax layer resulting in a slightly altered posterior probability, while the second defense adds noise with a variance to the predicted probability distribution. The authors argue that a certain drop in accuracy of the model is required as a trade-off in order to protect the system against these attacks.

Model Reconstruction from Model Explanations
Smitha Milli, Ludwig Schmidt, Anca D. Dragan, Moritz Hardt in ACM Conference on Fairness, Accountability, and Transparency, 2019 [288], *Information Extraction*
The paper proposes to extract a neural network by using gradients in cases where gradient information is available, e.g., through saliency maps that are exposed by a victim model. The idea is, similar to existing extraction methods that rely on predictions only, to find hyperplanes that separate linear regions of the ReLU networks by a binary search through the model input space. Such hyperplanes are identified by a change in gradients with respect to the model input. One can use them in a system of linear equations to recover the weights of the network. The paper proofs for two-layer networks that the number of queries required to reconstruct the victim model is less than the number required by methods that use only membership queries. For networks with more than two layers, the paper suggests a heuristic that adds the difference of gradients between the victim and the extracted model to the training loss of the extracted model. Experiments show that this strategy successfully reduces the number of queries required to extract the model compared to membership only queries, in particular for models with low complexity.

Model extraction from counterfactual explanations
Ulrich Aivodji, Alexandre Bolot, Sebastien Gambs in arXiv, 2020 [6], *Information Extraction*

This attack builds the transfer set with query-output pairs not just from the model API but also the explanation API which returns counterfactual examples. The authors claim that the model extraction attack is more effective when counterfactuals are also used and can lead to a better decision boundary extraction within a minimal query budget. Both single counterfactual as well a set returned by the DiCE framework by Mothilal et al. [299] is considered. The authors explain that counterfactuals provide much more information about the model and lead to more successful attacks because of the nature of counterfactuals that they construct explanations by finding examples close the input example but belonging to the desired class with only slightly different attributes. This enables a better exploration leading to better reconstruction of the decision boundary.

Model inversion attacks against collaborative inference
He, Zecheng, Tianwei Zhang, Ruby B. Lee. in ACSAC, 2019 [162], *Information Extraction*
The attacks are modeled in order to reconstruct the test (inference) data samples. The setup is the collaborative inference, when each party has only part of the network and sends intermediate predictions to the next part (in particular only two parts are considered). The reconstruction proposed either simply optimizes the input till it has the same intermediate representation (white box case), or constructs surrogate or shadow model and then trains reconstruction (blackbox). The conclusions of the authors include an insight that the split point matters for the attack - so the deeper the layers, the harder it is to perform reconstruction also the fully connected layers are harder to attack than convolutional layers.

Model inversion attacks that exploit confidence information and basic countermeasures.
Matt Fredrikson, Somesh Jha, Thomas Ristenpart in ACM SIGSAC, 2015 [132], *Information Extraction*
Model inversion attacks strive to infer information about training data given information on ground truth values, such as a class label, and known feature values of the input. This paper presents inversion attacks for two different scenarios: decision trees with a low-dimensional feature space and face recognition models based on neural networks with a large number of input dimensions. For decision trees, the basic idea is to use a maximum a posteriori (MAP) algorithm to search the input space for the feature value of interest, e.g., a sensitive attribute such as marital infidelity, given knowledge of the non-sensitive attributes and labels. Empirical results on this model class show that MAP works well for low-dimensional inputs, i.e., the sensitive attribute is predicted with high precision. However, the MAP algorithm does not scale with dimensionality since it relies on comparing combinations of feature values in the input space, i.e., it is intractable for high-dimensional inputs such as images. For this case, the paper proposes a gradient-based method that iteratively alters the input such that the model prediction is close to an expected label, with additional steps such as denoising. Experimental results show that the gradient-based extraction method works well for both reconstructing and for de-blurring images with different model architectures. They also indicate that rounding of gradient information and placement of sensitive features within a decision tree can protect against respective inversion attacks.

Model weight theft with just noise inputs: The curious case of the petulant attacker
Nicholas Roberts, Vinay Uday Prabhu, Matthew McAteer in arXiv, 2019 [361], *Information Extraction*
The paper studies empirically how well one can extract the weights of a CNN network with a known architecture based on random samples from a probability distribution. The experimental setup varies the selected distributions, e.g., uniform and Bernoulli distributions, to extract victim models trained on variants of the MNIST dataset. Overall, most models extracted based on the random samples have high accuracy. This indicates that random samples indeed can be sufficient to extract model weights. Further, the paper observes a correlation between the intuitive complexity of a data set, e.g., FashionMNIST being more difficult than MNIST, and the ratio of extracted model validation accuracy and victim validation accuracy. The paper suggests that one can use this ratio as a measure of dataset complexity.

Monte carlo and reconstruction membership inference attacks against generative models
Hilprecht, Benjamin, Martin Harterich, Daniel Bernau in PoPETs (Proceedings on Privacy Enhancing Technologies), 2019 [171], *Information Extraction*
Two attacks on generative models are presented. One is applicable to all generative models, the other is specific to VAEs. The authors propose to exploit the tasks of GANs and VAEs in order to extract training data information from them. The idea is that the models are trained to generate examples similar to training ones, so their response should be positive around the examples. This is approximated with Monte-Carlo integration for the first attack - just average of several requests. The second attack is called reconstruction attack (samples close to the real training data have high reconstruction scores).

Neural network inversion in adversarial setting via background knowledge alignment
Ziqi Yang, Ee-Chien Chang, Zhenkai Liang in ACM SIGSAC, 2019 [505], *Information Extraction*
The authors propose a model inversion technique in the so-called adversarial setting (as black-box), when the adversary whose intent is to reconstruct data does not have access to the model or training data. The approach consists of finding a replacing dataset (general dataset corresponding to the task) and training an inversion network on this dataset.

PRADA: Protecting Against DNN Model Stealing Attacks
Mika Juuti, Sebastian Szyller, Samuel Marchal, N. Asokan in EuroS&P, 2019 [208], *Information Extraction*
The authors present two important contributions to the field of model stealing research. First, the authors introduce a new attack method to extract DNN models which circumvents previously introduced defense methods. The attack does not rely on using the output prediction probabilities of the models to extract. In addition to training a model on the training pairs formed from querying the victim model, the authors add duplication rounds where synthetic samples are generated to increase coverage of the input space. Cross validation using Bayesian optimization with dropout is used for hyperparameter search. Synthetic samples in model extraction attacks are constructed either using the partially trained substitute model (Jacobian-

based) or independently of it(Random). Adversarial examples are crafted by modifying samples with the Jacobian matrix for a given DNN, which in turn tells what the impact of each feature is on the overall classification loss. Second, the authors introduce a new defense method against model extraction attacks called PRADA. The method observes the queries made to the model under attack. With this information, the authors calculate the pairwise l_2 distances between the observed queries to a predefined normal distribution. Based on thresholding this distance, the authors are able to detect if the queries were executed in the context of a model extraction attack. In a later study by Chen et al. [76], the defense was shown to be bypassed using the newly introduced query blinding attack. Furthermore, PRADA does not protect against sybil attacks.

Practical Black-Box Attacks against Machine Learning
Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, Ananthram Swami in Asia CCS, 2017 [332], *Information Extraction*
(I) The authors consider the task of attacking a black-box model that can be used only as an oracle. They propose a technique for recreating a training dataset from oracle predictions and then use FGSM and LLC attacks generated from the surrogate model.(II) The authors suggest to train a substitute network using generated inputs labeled by the oracle. The generation technique is a Jacobian-based augmentation technique. In essence it means that the small initial set of examples is extended according to the directions where prediction varies the most - so queries are done only in such directions identified by Jacobian of the substitute network. The architecture of the substitute model is selected only to match input-output dimensions. The adversarial samples itself are crafted with the Goodfellow algorithm and the Papernot algorithm.(III) Furthermore, the authors suggest to hide the NN outputs to protect against model stealing attacks.

Reverse-engineering deep relu networks
David Rolnick, Konrad P. Kording in ICML, 2020 [362], *Information Extraction*
The paper gives a formal approach to extract weights from ReLU-activated fully-connected convolutional neural networks. The approach builds on the observation that ReLU functions induce hyperplanes in the input space which separate linear regions of the input space. Theorems show that identifying the boundaries between these linear regions is sufficient to derive the weights and biases of the victim network. This in turn is achieved by querying the network for outputs. The sample complexity is approximately the number of parameters in case of constant-width networks. Since changing the order of neurons in a network and scaling of weights both result in a network that is isomorphic to the original one, the approach presented in the paper also yields a network isomorph to the victim model. Experiments validate the theoretical results on a network trained on MNIST data.

Security analysis of deep neural networks operating in the presence of cache side-channel attacks
Sanghyun Hong, Michael Davinroy, Yiitcan Kaya, Stuart Nevans Locke, Ian Rackow, Kevin Kulda, Dana Dachman-Soled, Tudor Dumitras in arXiv, 2018 [177], *Information Extraction*

This paper proposes DeepRecon, a cache side-channel attack to extract neural network hyperparameters. The threat scenario for this attack is an adversary co-located on the host system with a shared CPU instruction cache and with knowledge on the deep-learning framework used to train the neural network. DeepRecon derives function invocations by measuring cache access times using FlushReload, an established method for cache side-channel attacks. The function invocations in turn relate to neural network architecture hyperparameters, such as the number of layers, or to the control flow, e.g., they indicate when the CPU runs an inference. The paper demonstrates that DeepRecon can also help to identify if a model builds upon a common backbone architecture, e.g., ResNet, applied in a transfer learning setting an information of particular interest to an adversary that strives to combine model extraction with adversarial training. The paper concludes with two counter measures to obfuscate the model architecture by either executing dummy models in parallel to camouflage function invocations or by adding identity layers in random locations of the network. Both of these measures work well, although increasing the computational burden for the victim model inference.

Simulating Unknown Target Models for Query-Efficient Black-box Attacks
Chen Ma, Li Chen, Jun-Hai Yong in CVPR, 2021 [271], *Information Extraction*
To construct a generalized substitute model which can mimic any model that an adversary desires to attack, this attack uses several classification models over which a meta-model is trained on using a knowledge distillation loss objective. This results in a meta model which can mimic any attacked model with just a few queries, significantly reducing the required query budget for a successful attack. Bandits using random images are used to generate the input queries.

Special-purpose Model Extraction Attacks: Stealing Coarse Model with Fewer Queries
Rina Okada, Zen Ishikura, Toshiki Shibahara, Satoshi Hasegawa in TrustCom, 2020 [316], *Information Extraction*
The authors present a study in which methods from model stealing attacks are further evaluated. In particular, the authors try to use methods to generally steal models for the use-case of special-purpose model extraction. Here, the authors differentiate target models based on the number of classes in the classification task. For higher number of classes, the authors consider the models as general-purpose targets. In contrast, for a small number of classes (e.g. two), the authors consider the models to be part of a special-purpose system. Special purpose models do not try to reconstruct all the classes like general purpose, but only try to distinguish a few classes, which is already adequate for the purpose of model theft. Based on this observation, the authors test the applicability of standard model extraction techniques to specifically reconstruct surrogate models which can be seen as special-purpose models. The authors show in experiments on the CIFAR-10 dataset, that only a fraction of the queries is required to construct a special-purpose model using a CNN architecture, opposed to extracting the general-purpose counterparts. In their discussion section, the authors briefly show potential defense strategies. First, the authors argue that limiting the number of possible quires would not protect against their attack. Special-purpose model extraction attacks require less queries compared to the standard attacks. Second, the authors propose to randomly include false predictions in the system. This would not be

favorable by developers, maintainers, or users of the systems.

Stealing Machine Learning Models via Prediction APIs
Florian Tramer, Fan Zhang, Ari Juels, Michael K. Reiter, Thomas Ristenpart in USENIX, 2016 [440], *Information Extraction*
These attacks rely on the fact that ML APIs (e.g. by AWS or Azure) return confidence scores along with the prediction class label. Using the confidence scores, the authors construct equations solving attacks for classifiers based on logistic regression and neural networks and present two novel attacks for decision trees. In experiments, attack is tested against models hosted on BigML and AWS. Even though the attack may not be applicable to real-world sized NNs, the paper still makes important first steps into this new research direction. This attack can be preventing by rounding the output of the model via the API or simply not returning the confidence scores. In a later publication by Juuti et al. [208], the first line of defenses was shown to be bypassed using a new model extraction approach not relying on the prediction probabilities.

Stealing hyperparameters in machine learning
Binghui Wang, Neil Zhenqiang Gong in S&P, 2018 [454], *Information Extraction*
Hyperparameter value selection is computationally expensive since it relies on methods like cross-validation. However, when training a model on a ML-as-a-Service (MLaaS), the hyperparameters selected for the final model usually are not included in the API result, often to protect intellectual property of hyperparameter selection methods. The paper shows, however, that an adversary can still probe the MLaaS with a data subset which is computationally cheap to estimate hyperparameters for, extract the hyperparameters from the trained model, and use them outside the MLaaS platform for more expensive training hence avoiding service cost.For this purpose, the paper introduces a method to extract hyperparameters based on knowledge of the objective function and of model parameters. The idea is that an optimal choice of hyperparameter values sets the gradient of the objective function of an ML model to zero. The method exploits this by setting the gradient of the objective function to zero to obtain a set of linear equations which yields the unknown, protected hyperparameter value as a solution. Experiments show that the method proposed works well for a wide range of model classes with actual MLaaS offers. Defenses such as numerical rounding of model parameters only offer limited protection.

Stealing neural networks via timing side channels
Vasisht Duddu, Debasis Samanta, D Vijay Rao, Valentina E. Balas in arXiv, 2018 [113], *Information Extraction*
This paper proposes a two-step method to estimate a neural network architecture from a blackbox victim model. The first step is to estimate the depth of the neural network based on the execution times of the neural network. Here, the paper proposes to learn a regression model based on meta-dataset of architectures and their execution times to predict the victim model depth. The predicted depth then is used to constrain the search space of model architectures and only consider remaining architecture parameters such as convolutional kernel size. The second step

is the actual architecture search. There, a reinforcement-learning model first samples and trains a neural network architecture and then calculates a reward by how well the sampled network imitates the victim model, i.e., how accurate it predicts the victim output. Experiments indicate that the two-step method can achieve high accuracy both on the depth prediction and on the reconstructed model accuracy.

The secret revealer: Generative model-inversion attacks against deep neural networks
Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, Dawn Song in CVPR, 2020 [535], *Information Extraction*
The paper assumes white-box setup and constructs a GAN that is trained on publicly available data. Then it is used to reconstruct sensitive features from damaged inputs using some additional knowledge. The technique is termed GMI - generative model inversion. The observation at the basis of the technique is that general features can be learned even without knowing the exact training dataset and that high-accuracy models are prone to remembering more, which leads to easy privacy attacks. The considered setup is white-box and the attack should reveal features that are connected to the label (prediction).

The secret sharer: Evaluating and testing unintended memorization in neural networks.
Nicholas Carlini, Chang Liu, Ulfar Erlingsson, Jernej Kos, Dawn Song in USENIX , 2019 [60], *Information Extraction*
The authors describe the effect of memorization in neural networks when the secret private training data can be further extracted from the model. They consider language models, where some text parts can be restored after training, even if they are not needed to be memorized for generalization. They propose to use canaries - some specific phrases inserted in the text to identify if the model memorized them. The main requirement to a canary is to contain some random part that is not needed to be learned for the task learning. They show that overfitting is not connected to this - and regularization does not prevent this. Only differential privacy measures can help. They distinguish overfitting and overtraining (when the validation error stops decreasing) and use for experiments not overfitted models. They demonstrate that random information is remembered even before the model is trained till the end. The proposed technique measures the perplexity of canaries, thus identifying if it was learned or not. The authors propose two approaches for calculating the exposure to the canary.

Thieves on SesameStreet Model Extraction of BERT-based APIs
Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, Mohit Iyyer. in ICLR, 2020 [222], *Information Extraction*
The paper proposes a model extraction attack which is tailored towards a transfer learning setting with a BERT-based classifier as the victim model. To steal the victim model, two types of queries are constructed: nonsensical, random sequences of tokens (random strategy) and sentences / paragraphs from WikiText103 (Wiki strategy). In experiments on classification and question-answer models, both strategies are effective at reconstructing the victim model. Random queries turn out to only be slightly less effective than proper sentences in this setting.The

paper also reports results on two non-specific defenses against model extraction: membership inference and watermarking. While effective to some extend, both of these defenses make strong assumptions on the capabilities of an attacker, e.g., that the attacker cannot fool membership inference. Watermarking only allows identification of extracted models after a successful attack and does not prevent the leakage of private information. Like membership inference, an attacker can circumvent watermarking with reasonable effort.

Towards Reverse-Engineering Black-Box Neural Networks
Seong Joon Oh, Max Augustin, Mario Fritz, Bernt Schiele in ICLR, 2018 [315], *Information Extraction*
The subject of the paper is to predict victim model attributes such as architecture, e.g., the number of layers, details on the optimization method used to train the victim model, e.g., the batch size, and descriptive information on the training data. The basic idea to extract this information is to use meta-learning. The first step is to create a data set of white-box models with known attributes and collect data on their query behavior, i.e., the outputs they produce for carefully selected inputs. The paper uses three different methods to craft such queries: kennen-o, which uses a fixed set of samples kennen-i, which uses specially crafted samples and kennen-io, which combines both approaches for multi-attribute prediction. The second step is to train a meta-model that predicts attributes based on input-output pairs. At inference time, the victim model is probed with the crafted queries. The obtained input-output pairs are submitted to the meta-model to predict the attributes of the victim model. Experiments show that the meta-model indeed is capable of inferring model attributes, significantly better than a random baseline.

Understanding membership inferences on well-generalized learning models
Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, Kai Chen in arXiv, 2018 [266], *Information Extraction*
The authors follow the work of Shokri, but put attention on the fact that Shokri attack requires the model to be overfitted to work. They propose to identify the examples that have high influence on the model while training and perform membership inference through them - then the model can generalize well, but still be vulnerable.

You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors
Gong, Neil Zhenqiang, Bin Liu in USENIX Security Symposium, 2016 [150], *Information Extraction*
Attack that is capable to infer some private attributes (for example city where user lives) from the other connected information from social networks.

Chapter 3

Detailed Analysis

F. Assion, *neurocat*
B. K. Sreedhar, *neurocat*
B. Srinivasan, *neurocat*
Dr. H. Trittenbach, *neurocat*

Literature on adversarial deep learning is manifold and shows an increasing degree of specialization in the different fields. A typical consequence of such increasing specialization is an narrowing of research questions. For adversarial deep learning, we observe that most often, research questions are addressed in independent silos: Evasion attacks rarely are used in conjunction with poisoned data; hyperparameters such as the norm to measure perturbation sizes, are often fixed to a few options; effectiveness of methods relative to the choices of data sets not explicitly considered.
A reason for independent efforts is the prohibitively large space of attack and defense configurations. Comparing evasion attacks and defenses with a wide range of different norms is a large experimental burden and often beyond scope of an individual publication. Adversaries, however, are not bound to an individual silo, and must expect different attacks and defense mechanisms to co-occur in practice. It is not difficult to imagine that such co-occurrence may create interaction effects, e.g., a stronger evasion attack if the model was trained on poisoned data. The question is how to assess complex setups that involve multiple attack and defense configurations systematically.

In this chapter, we investigate the interaction of different classes of attacks and defenses empirically. The focus is on evasion and poisoning threats, i.e., adversaries that try to harm a victim model during training as well as after deployment. Specifically, we analyze different methods along an end-to-end workflow by modifying hyperparameters of model training, add-on defense methods, and inference attacks for different data sets. Here, an important objective of our experiments is to single out which methods perform well with each other, and which ones have a mitigating effect. An example would be to investigate if there is an interaction between adversarial training and a defense add-on with respect to the success of an attack.

Our investigations require a factorial design of different types of data, model and methods, as well as an evaluation by state-of-the-art metrics. We use two data sets in our experiments. The first one is a COVID-19 detection on the basis of chest X-ray images. The second one is CIFAR-10, an established benchmark data set that facilitates comparison of our results with observations reported in the literature. Based on the two data sets, we train several models. They differ by the classification type, e.g., multi-class vs. binary, by the optimization method used for training, and by whether they use data augmentation or not. For the attack and defense methods, we select the candidates based on the literature review shown in Chapter 2.

We implement our experiments in a modular way, which is extensible and reproducible. This is, our implementation allows to run an arbitrary number of different combinations of adversarial attacks, poisoning attacks, adversarial defenses, and poisoning defenses. Most attacks and defense methods further have configurable hyperparameters. This results in an experimental space which is too large to be evaluated exhaustively. As a consequence, we formulate guiding research questions that focus on a subset of the experimental space. Nonetheless, our modular implementation can be used for exhaustive evaluations, e.g., guided by evolutionary optimization approaches, see Section 3.5.

In the following section, we introduce our guiding research questions. They are the basis for our experimental plan, i.e., the specific experiment configurations to run, and give structure to our results and conclusions.

3.1 Research Questions

We structure our experimental study into three topics: "Effects of Hyperparameters and Metrics", "Data Set Differences", and "Dependencies between Evasion and Poisoning Robustness". This section introduces several research questions for each of these topics.

Next, there are two perspectives we consider in our experimental study: the one of the owner, i.e., the developer of the victim model, and the one of the adversary. Both evaluate results of attack and defense methods by different criteria, such as the computational budget or the success of an attack/defense, see Figure 1.10. We will show how these criteria are central to the interpretation of our experimental results, and how they help in answering our research questions.

Effects of Hyperparameters and Metrics. The success of evasion and poisoning attacks depends on the choice in hyperparameter values and metrics. This entails several challenges.

For one, every attack and defense method comes with a set of configurable hyperparameters, e.g., an attack budget and a constraint on the number of iterations for adversarial attacks, a poisoning ratio or a trigger size for poisoning attacks. Literature on adversarial machine learning often does not provide clear guidelines on how these hyperparameter choices affect the success of the respective method and on related metrics such as model accuracy.

Next, choosing a reasonable parameter of an attack or defense often depends on the data set the victim model is trained on. However, medical imaging data sets are not a focus in evasion and poisoning literature. So, suggestions on how to select a good configuration of hyperparameter values for medical imaging data are rare. Since the focus of our study is on the COVID-19 radiology database, one challenge will be the determination of reasonable hyperparameter values. Another challenge is that the selection of metrics in literature often is limited, e.g., the L^∞-norm to quantify the visibility of an adversarial attack. However, it is not clear whether a method also works well with a different choice of imperceptibility metrics. For example, L^∞-based adversarial training might be less successful when confronted with a L^2 adversarial attack.

In our experimental study, we assess the effects of hyperparameters and metrics with the following questions:

- *RQ 1.1:* How do parameter adaptations of poisoning attacks influence the performance of the machine learning model?
- *RQ 1.2:* Do certain adversarial defense methods transfer their success to metrics, which are not rooted in their respective approach, e.g., FGSM adversarial training paired with a L^∞-based attack?
- *RQ 1.3:* Which areas of the hyperparameter space yield efficient evasion attacks for the COVID-19 data set?

Data Set Differences. The medical imaging domain is the main use case in this experimental study. However, we also study the widely-used CIFAR-10 data set to compare our results with published results and to investigate if our novel insights transfer to other data sets as well. Images of the CIFAR-10 data set are significantly smaller in size than images of the COVID-19 data set ($32 \times 32 \times 3$ vs. 299×299, reshaped to 224×224 pixels). Although we limit our study to two data sets, we expect that the differences in data set size, image size, and intended classification task have an effect on the experimental results.

For one, it has long been a hypothesis in literature that data sets may contain robust and non-robust features, which adversaries can exploit with different rate of success. In our case, this begs the question whether adversaries are more or less successful on the COVID-19 data set compared to the well-studied CIFAR-10 data set. A significant difference in this context would strengthen the hypothesis that the nature of the data set is a central factor for the evaluation of security threats.

Further, CIFAR-10 contains approximately three times the number of images compared to the COVID-19 data set. This may have an impact on poisoning attacks which typically alter a certain percentage of data points during the training of the victim model. However, it is an open question if an absolute difference in the number of poisoned data points affects the success of an adversary using respective attacks.

Another data-set-related open research question is whether the concrete definition of the classification task plays a role in the robustness of the machine learning model. For example, the COVID-19 data set comes with four different labels, namely COVID-19 positive, lung opacity (non-COVID lung infection), viral pneumonia, and healthy (normal) patients. If one is interested in detecting COVID-19 cases, there is a choice to train a 4-class classifier, or a binary one-vs-all classifier for COVID-19 vs. non-COVID-19 patients. It is yet unclear how the nature of the classification task affects the success of adversarial and poisoning attacks or defenses.

Overall, this results in the following promising research questions:

- *RQ 2.1:* Is the success of the different poisoning and adversarial attacks / defenses influenced by the data set choice (CIFAR-10 vs. COVID-19 two-class vs. COVID-19 four-class)?
- *RQ 2.2:* Does data-set size play an important role for poisoning attacks and defenses?

Dependencies between Evasion and Poisoning Robustness. Poisoning and adversarial attacks share the same goal: in both cases, an adversary tries to force a victim model to misclassify some of the input data points. On the one hand, at run-time, adversarial attacks exploit characteristics of the decision boundary of an already trained machine learning model by adapting input data points in a direction that pushes them across the decision boundary. On the other hand, poisoning attacks alter the decision boundary of the victim model by influencing the training data of the victim model.

In spite of this common goal, both notions of robustness have mainly been studied separately, thus there is only a very limited understanding of the dependencies between evasion and poisoning robustness. However, the topic is broad, and we narrow down our efforts to a few well-defined research questions.

One of these questions is whether poisoned models are more or less affected by adversarial attacks. In other words, we ask whether poisoning and adversarial attacks have a reinforcing effect on each other. If there is a substantial difference in the reaction of poisoned models to adversarial attacks compared to unpoisoned models, this will have consequences for the optimal attack strategy of the adversary. At the same time, such insights may also help to detect poisoned models, or poisoned data points, by using adversarial attacks. This could lead to novel detection mechanisms preventing the deployment of poisoned models.

Another question is if a composition of poisoning attacks and adversarial defenses, e.g., adversarial training, have an interaction effect. Both classes of methods often interfere in the training process of the machine learning model. Thus, there is a risk that their effect changes when applied together.

In our experimental study, we address the following research questions:

- *RQ 3.1:* Are poisoned models more or less susceptible to adversarial attacks?
- *RQ 3.2:* Do adversarial defenses influence the success of poisoning attacks?

3.2 Technical Setup

Implementing an experimental framework for evaluating evasion and poisoning attacks is difficult. One reason is that methods often are designed for individual use, and their integration into larger experimental setups is not well defined. For instance, a combination of a poisoning attack and adversarial training yields the problem in which order to apply methods, i.e., first poisoning or first adversarial training. In other cases, like adaptations of evasion attacks, one has to modify the algorithmic implementations to allow for the extra steps such as “boosting”, i.e., a method

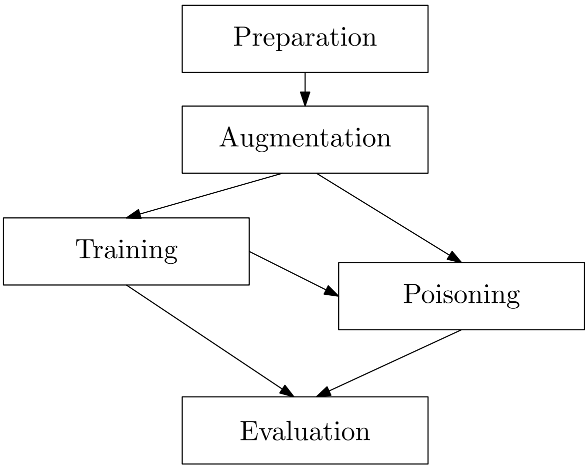


Figure 3.1: Stages in the experiment DAG. Arrows indicate the dependencies where the sink (arrow head) depends on the source.

that accumulates gradients to take steps in the same direction to avoid unnecessary backtracks. Another challenge is a large number of hyperparameters. Methods typically come with multiple hyperparameters for which one has to find fitting values. This requires frameworks to support for an easy configuration of hyperparameters. Finally, experiments are iterative and incremental. Based on the first results, one typically extends or modifies implementations. Here, long runtimes, e.g., for model training, quickly become prohibitive. Thus, one has to separate the experimental pipeline in stages that can be executed independently. This is, one must be able to run additional evaluations without re-training existing models. However, a re-training of models must invalidate existing evaluations and trigger new ones.

Implementation Details. We design our technical experimental framework in a modular and expandable way. Modular means that we define interfaces on the inputs and outputs for each type of method. So for instance, training a model requires a data set as an input and produces a trained model file as an output; attacking a model requires a data set and a model as an input and produces adversarial images as an output. One can extend the framework with new methods by implementing these input/output definitions. This results in a Directed Acyclic Graph (DAG) with explicit dependencies between methods, see Figure 3.1.

Experiment Stages:

- **Preparation**: downloading data and converting into required format
- **Augmentation**: standard augmentation techniques, such as random-crops and rotations
- **Training**: model training, including adversarial training and feature denoising as well as training of the auto-encoder for defense methods
- **Poisoning**: generation of poison examples and model retraining

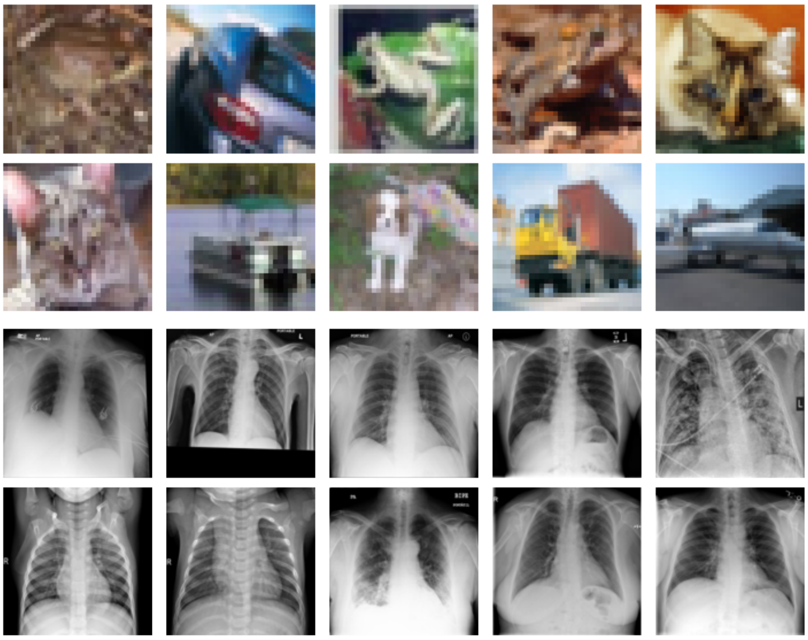


Figure 3.2: Sample images of the CIFAR-10 and COVID-19 data set. The CIFAR-10 training set is evenly distributed, i.e., every class makes up 10% of the overall data set. In the COVID-19 data set there is a clear imbalance towards non-COVID patients. In the case of the binary classification task, 74% of the training data belongs to the ”normal” class.

- **Evaluation**: execution and evaluation of evasion attacks with possible extension such as boosting

We implement the DAG with an open source experiment management software called Data Version Control (DVC)[1]. DVC allows to specify the DAG and hyperparameter choices as separate configuration files. This facilitates the definition and experiment execution using such configuration files. With an additional control over random seeds, we make our experiments reproducible up to low-level variations such as the choice of hardware. All our experiments are executed on AWS “g4dn.xlarge” instances that run on NVIDIA-T4-GPUs.

3.3 Models, Data & Methods

The goal of this coding project is a detailed analysis of a selection of evasion and poisoning robustness methods and their interaction effects. The results produced in this chapter are then used in later stages of this project to identify and shed light on open questions and vulnerabilities in adversarial machine learning. In the following, we elaborate on the selection of methods, i.e., we provide an overview of the implemented algorithms as well as used models and databases.

[1] https://dvc.org/

3.3.1 Models and Data

Throughout the experiments, we rely on one specific model architecture: the ResNeXt-50 32x4d model [490]. We use two data sets on which the model is trained, namely the CIFAR-10 and the COVID-19 data set (see Figure 3.2), see Section 3.1. In the case of the CIFAR-10 data set, we initialize the training with a ResNeXt-50 pretrained on ImageNet. This implies that the CIFAR-10 pixel values are in a $[0, 1]$ range and normalized according to the means and standard deviations calculated on the CIFAR-10 training data set. The grayscale images of the COVID-19 database, on the other hand, are loaded to a $[0, 1]$ range and then normalized with mean 0.5 and standard deviation 0.25. The ResNeXt-50 model is trained from scratch for the COVID-19 data set. Furthermore, we differentiate between two tasks for the COVID-19 data. For one, we train 4-class classifiers differentiating between healthy, COVID-19, viral pneumonia, and lung opacity patients. Additionally, we also train a binary one-vs-all classifiers for COVID-19 vs. Normal patients. For both data sets, we experiment with various data augmentation techniques (e.g., horizontal flip, vertical flip, and random crop), as well as with Automatic Mixed Precision (AMP), i.e., using half-precision during training where applicable, to find well-performing models as a basis for our robustness experiments.

3.3.2 Evasion Robustness

In the course of our literature review (Chapter 2), we have selected several state-of-the-art adversarial attacks and defenses. We have selected these methods with the goal of having a diverse set of strategies in our experiments. Another selection criterion has been the effectiveness of the methods as demonstrated in the experimental evaluations of the respective publications.
To answer questions on the transferability of methods (RQ1), we focus on attacks which allow for varying the imperceptibility metrics. This is, our experiments compare different imperceptibility metrics such as L_0 and L_2 in addition to the popular L^∞ norm.
We assume the strongest adversarial threat model in our experiments where an adversary has access to the gradients of a victim model. This gives way to apply white-box attacks, which generally are stronger than attacks that only have limited access to a model, e.g., only its inferences. The strong threat model allows us to strive for general statements about the robustness level of the victim model.
On the defense side, we investigate adversarial defenses from different categories. In particular, we look at representative methods from the model modification, model training and, add-on defense class.
The following methods are part of the experimental study:

Adversarial Attacks:

- FGSM [152]
- PGD [276]
- AutoAttack [97]

- Carlini & Wagner Attack [62]
- Orthogonal PGD [48]
- Blind-Spot Attack [523]

The PGD and the Orthogonal PGD attack are also combined with gradient boosting [108], a method that adds a momentum to every iteration step of the attack.

Adversarial Defenses:

- FFGSM [472]
- Matching Prediction Distributions [451]
- Denoising Blocks [488]
- Barrage of Random Transforms [350]

3.3.3 Poisoning Robustness

With poisoning robustness, we also focus on a restricted threat model. We assume that an adversary does not have control over the labeling function during the training of the victim. As a consequence, the attacker must use so-called clean-label attacks for the injection of backdoors into the model. Clean-label attacks require more sophisticated optimization strategies compared to the common pattern-key approaches, such as BadNets [155]. One implication of the restricted threat model is that an adversary requires significant knowledge of the victim model. However, it also leads to backdoors that are significantly harder to detect and defend against. As with adversarial attacks, this gives way to general statements on the robustness level of the victim model.

In our experiments, we investigate two poisoning defenses, one particularly designed for the threat imposed by clean-label attacks.

The following methods are part of the experimental study:

Poisoning Attacks:

- Bullseye Polytope [5]
- Poison Frogs [387]

Poisoning Defenses:

- Februus [105]
- Deep k-NN [337]

3.4 Results

We run a variety of experiments to answer the questions outlined in Section 3.1. Our results can serve as a foundation for future work, in particular the derivation of best-practices and the identification of unsolved problems. Every guiding research question is equipped with one or more experimental pipeline configurations. In this section, we summarize the configurations and describe the results.

3.4.1 Effects of Hyperparameters and Metrics

RQ 1.1: How do parameter adaptations of poisoning attacks influence the performance of the machine learning model?

Poisoning attacks alter the training data of the victim model. The adversary strives to inject a backdoor into the victim model, while staying inconspicuous. Here, we only consider clean-label backdoor attacks. This is more demanding for a victim to detect such backdoors compared to attacks that exploit the labeling function. However, the victim might still detect and dismiss a poisoned model based on unusual, unsatisfactory performance results after training. Thus, the adversary has to make sure that the poisoned data does not decrease the overall accuracy level of the victim model.
Existing poisoning attacks come with a variety of hyperparameters. A common hyperparameter is the "number of data points" to poison during training. Additionally, most clean-label attacks optimize the creation of poisoned data. These optimizations usually are iterative which results in an "iteration number" one has to specify. In this research question, we analyze how the choice of hyperparameter values impacts the success of the adversary.

Setup. In our experiments, we run the Bullseye Polytope and Poison Frogs poisoning attacks with varying parameter values on a COVID-19 detection model. The Bullseye Polytope alters a specified number of data points from the poison label class with an iterative gradient method. This perturbation process satisfies a given L^∞-constraint, which we set to $8/255$ – we leave further variations of the constraint metrics and values to future work. The goal of the poison perturbations is to create representations in the penultimate layer of the victim that are close to the ones of the desired target class. Ideally, this leads to the misclassification of the target class data points during deployment. The Poison Frogs poisoning attack is a targeted, clean-label attack strategy. Similar to the Bullseye Polytope poisoning attack, the poisoned data points are generated with the goal of creating similar representations for the poison and target data points in the penultimate layer. However, the Poison Frogs procedure additionally regularizes the L^2-distance between the benign and the poisoned data points. The two-part optimization objective is solved with a forward-backward-splitting iterative procedure. We evaluate a range of values for the number of poisoned data points, as well as a range of different iteration numbers for the optimization method of the Bullseye Polytope. We then assess the performance of the resulting poisoned model on the clean test data split. Here, the confusion matrix is useful to check if the backdoors indeed are successful in altering predictions in the desired way.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes)
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate, with poisoning attack
- **Poisoning Attacks**
  - Bullseye Polytope - $\{10, 20, 30...100\}$ iterations, $\{10, 20, 30, 40\}$ sample points from poison label ($= 1$) and target label ($= 0$), retraining on validation data (10 retraining epochs), $8/255$ as L^{∞}-norm constraint
  - Poison Frogs - $\{10, 20, 30...100\}$ iterations, $\{10, 20, 30\}$ sample points from poison label ($= 1$) and target label ($= 0$), retraining on validation data (10 retraining epochs), $8/255$ as L^{∞}-norm constraint
- **Defenses**
  - DeepKNN
  - Februus

Results. Throughout our experiments, we observe that the Bullseye Polytope attack does not reliably produce the desired misclassification. We evaluate the success of poisoning through the confusion matrix resulting from the victim model. In the case of a successful backdoor injection, there is an increase of predictions of the poison label class, accompanied by a decrease of target class predictions (see Table 3.1c). Poison Frogs is successful as well in changing the classification results in the desired way, see Table 3.2. However, Poison Frogs is less effective compared to a successful Bullseye Polytope.
However, in our experiments we also find poisoned models that have a bias towards the target class (see Table 3.1b). Another observation is that poisoning seems to also affect predictions on data points which do not belong to the poison or target class. This may decrease the victim model accuracy and may ease the detection of the attack. One reason for the instability of the Bullseye Polytope attack might be its implementation in the ART Toolbox. The publication of Bullseye Polytope [5] outlines different versions of the algorithm. The ART implementation bases on one of the simple versions, i.e., without using substitute models for the calculation of the data perturbations and using the same number of target and poison samples.
Figure 3.3a graphs the impact of the iteration number of the Bullseye Polytope attack on the accuracy of the poisoned victim model. As one can see, the iteration number does impact model accuracy. If the model accuracy is a key factor in the chosen approach to detect whether the model is poisoned, the adversary can optimize this parameter freely without increasing the risk

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	450	22	5	35	512
COVID	40	109	1	26	176
Viral Pneumonia	5	1	36	1	43
Lung Opacity	51	10	3	205	269
	546	142	45	267	1000

(a) Unpoisoned model. Weighted F1 score = 0.797

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	386	61	25	40	512
COVID	39	97	3	37	176
Viral Pneumonia	6	1	32	4	43
Lung Opacity	61	29	5	174	269
	492	188	65	255	1000

(b) Unsuccessful Bullseye. Iterations: 60; Samples: 10; Weighted F1 score = 0.6904

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	258	247	6	1	512
COVID	46	128	1	1	176
Viral Pneumonia	10	16	17	0	43
Lung Opacity	83	164	0	22	269
	397	555	24	24	1000

(c) Successful Bullseye. Iterations: 40; Samples: 10; Weighted F1 score = 0.4144

Table 3.1: Confusion matrices for different models. The poisoned models are attacked by the Bullseye Polytope attack with poison label "COVID" and target label "Normal".

of being detected. It is not overly surprising that the iteration number does not significantly impact the accuracy of the victim, given that the attack relies on an L^∞ constraint. We expect that the size of the ϵ-value of the constraint plays a more central role with respect to model performance, because it defines the degree of noise added to the poisoned images.

The number of poisoned samples, however, influences the model accuracy, see Figure 3.3b. This is intuitive, since one would expect a higher number of poisoned samples to also increase the attack strength.

The second poisoning attack, Poison Frogs, successfully increases the number of target classifications, see Table 3.2b. However, the it is slightly less successful than Bullseye Polytope with a good selection of hyperparameter values. Interestingly, neither the number of iterations (Figure 3.3a) nor the number of poisoned samples (Figure 3.3b) seem to have a significant effect on the accuracy of the poisoned model.

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	450	22	5	35	512
COVID	40	109	1	26	176
Viral Pneumonia	5	1	36	1	43
Lung Opacity	51	10	3	205	269
	546	142	45	267	1000

(a) Unpoisoned model. Weighted F1 score = 0.797

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	356	141	8	7	512
COVID	29	133	2	12	176
Viral Pneumonia	16	9	17	1	43
Lung Opacity	52	132	2	83	269
	453	415	29	103	1000

(b) Poison Frogs. Iterations: 20; Samples: 10. Weighted F1 Score: 0.68

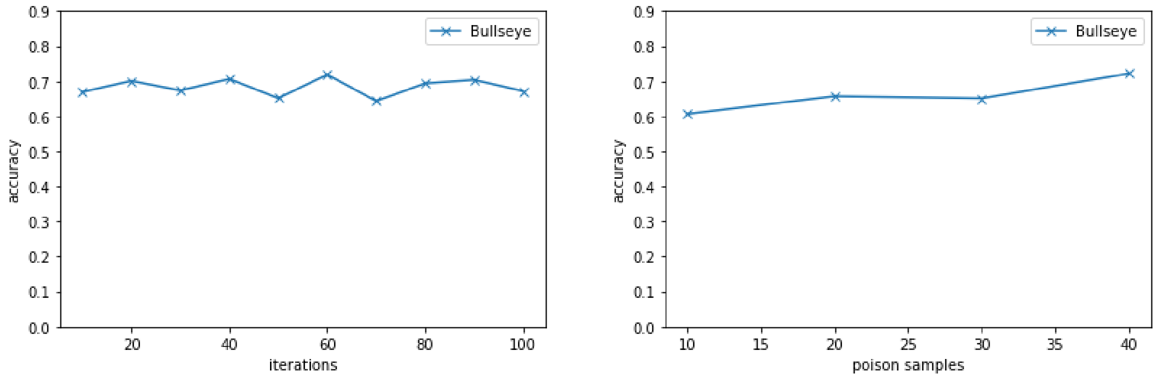
Table 3.2: Confusion matrices for different models. The poisoned models are attacked by the Poison Frogs attack with poison label "COVID" and target label "Normal".

We also evaluate if one can defend against poisoning with defense methods. Figure 3.5 shows the model accuracy of a base model, a model poisoned with Poison Frogs and two defenses on the three different data sets.[2] Here, we can see that poisoning indeed affects model accuracy. DeepKNN does help to mitigate this effect, in particular for Covid-2. Februus shows an adverse effect, it reduces model accuracy even further instead of mitigating the effect of poisoning. We expect the reason for this are the small differences between poisoned images and original images. In such cases, data cleaning defenses such as Februus seem to not be effective.

Finally, we plot some example images for visual assessment of the poisoning. Figure 3.6 shows the original images and predictions in the top row and the poisoned image with prediction of the poisoned model in the bottom row. All poisoned examples are successful, i.e., change the classification from class 0 (Normal) to class 1 (Covid). In all inspected cases, we found the poisoning to not alter the image in a way that is visible to a human observer, i.e., the L^∞ constraint with an epsilon of $8/255$ indeed leads to results that are imperceptible.

In summary, the results indicate that adversaries can indeed exploit hyperparameters to tune the effect of the poisoning. This gives way to choose a good trade-off between an attack-specific success metrics, e.g., the number of poison and target label predictions via the confusion matrix, and a detectability metric, e.g., the victim model accuracy.

[2] Because of poor initial results, we have decided to not run comprehensive experiments for Februus on CIFAR-10.



(a) Effect of the iteration number on the accuracy of the poisoned model.

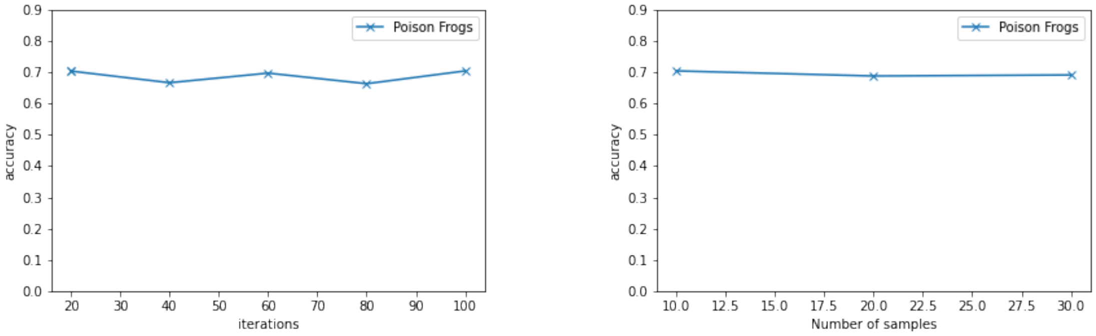
(b) Effect of the number of poisoned samples on the accuracy of the poisoned model.

Figure 3.3: Bullseye Polytope poisoning attack on the COVID-19 test data set

RQ 1.2: Can adversarial defense methods transfer their success to metrics, which are not rooted in their respective approach?

The motivation for this question is the contest between adversary and defender: an adversary tries to break existing defense mechanisms while the defender strives to be robust against variations of the attacks. One of the variations that an attacker might exploit is switching the imperceptibility metric used to craft adversarial examples. An imperceptibility is a constraint on the size of the perturbation measured by a metric such as the L_2 norm between pixel values of the original and perturbed image. Put differently, it is a mathematical quantification acting as a proxy to measure how well a perturbation can be perceived.
Ultimately though, if an adversarial perturbation is perceptible lies in the eye of a human observer. A human may perceive a high value under the L^∞ metric, i.e., the image contains at least one pixel with high perturbation. However, the observer may miss many small perturbations that have a low value in the L^∞ metric, but a high value for L^2. In a similar way, a victim that defends on attacks that exploit a specific imperceptibility metric may be weak in defending against attacks that rely on another one. Thus, an adversary may break a defense by switching the metric of the attack. Vice versa, defenses are more attractive if they are effective against a broad choice of imperceptibility metrics.

Setup. In our experiments, we evaluate the effectiveness of attacks and defenses with non-matching imperceptibility metrics. To this end, we compare three different models: A normal base model (trained without any adversarial defense), an adversarially trained model (trained with FFGSM) and a model equipped with a detection add-on that detects adversarial images by evaluating the reconstruction error for an autoencoder trained on the original training data (Matching Prediction Distributions method). For FFGSM, we fix the imperceptibility metric to L^∞ during the adversarial training. The Matching Prediction Distributions defense relies on a model-dependent version of the Kullback-Leibler distance for the training of the autoencoder.



(a) Effect of the iteration number on the accuracy of the poisoned model.

(b) Effect of the number of poisoned samples on the accuracy of the poisoned model.

Figure 3.4: Poison Frogs poisoning attack on the COVID-19 test data set

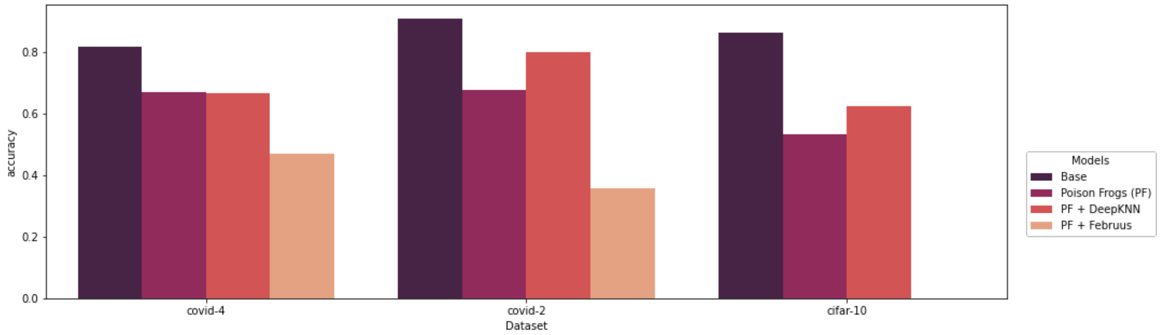


Figure 3.5: Comparison of poisoning attack success on different data sets and using Deep KNN and Februus defenses to mitigate the attack.

We found that training with AMP does not improve the prediction quality of the models, thus we only report results without AMP. We evaluate all adversarial attacks (FGSM, PGD and AutoAttack) with L^∞ and L^2-norm projections. In summary, the question is if the two selected adversarial defense methods can improve the robustness of the victim model, even when confronted with adversarial attacks that use distance measures which do not match the one used by the defense.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes)
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate, without AMP, with / without adversarial training, with / without detector

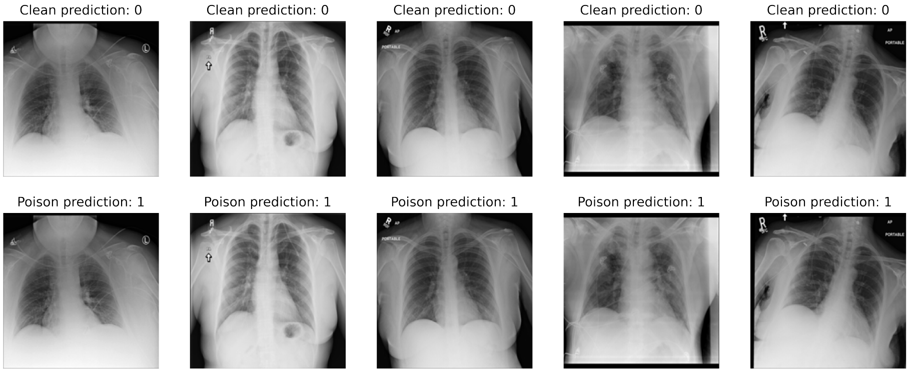
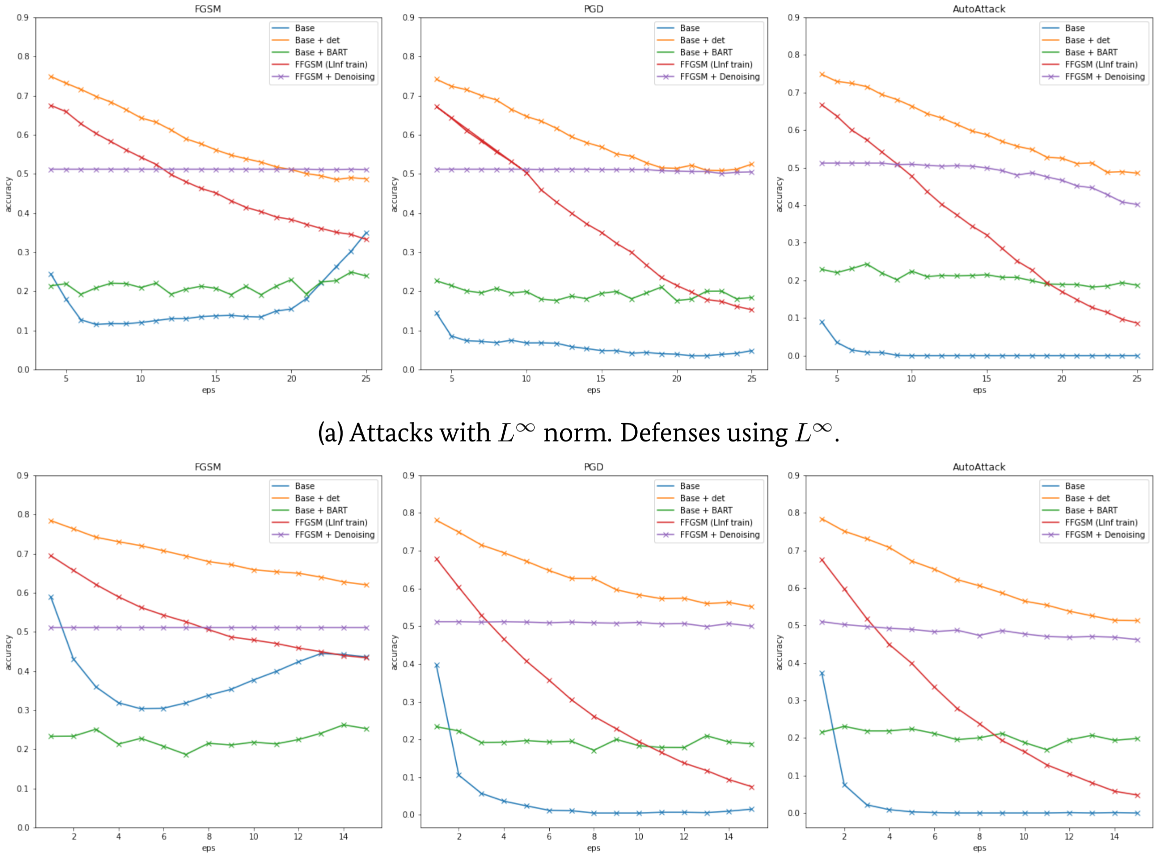


Figure 3.6: Visual inspection of poisoned images on the COVID-19 4 class data set.

- **Adversarial Defenses**
  - FFGSM - fix one ϵ-value for all models during training ($8/255$ as ϵ-value)
  - Matching Prediction Distribution with adaptive Kullback-Leibler distance
- **Adversarial Attacks**
  - FGSM, PGD and AutoAttack with L^∞ and L^2-norm constraint for varying ϵ-values

Results. Figure 3.7 shows the results for attacks using L^∞ (Figure 3.7a) and L^2 (Figure 3.7b) metrics. Overall, the base model without any defense shows a poor accuracy across all adversarial attacks. Even with non-matching imperceptibility metrics (Figure 3.7b), adversarial defenses seem to improve the robustness against adversarial attacks.
The Matching Prediction Distributions defense (Base + det) consistently outperforms FFGSM adversarial training. Even for high values of the imperceptibility constraint for the attack (high *eps* values on the x-axis), Matching Prediction Distributions still yields relatively high accuracy predictions. This even holds true for L^2-based adversarial attacks, although FFGSM utilizes L^∞-based adversarial examples during the training of the victim model. All this indicates that adversarial defenses can transfer their success to “unseen” distance measures in some cases. One can further observe that Barrage of Random Transforms (BART) has only a small effect as a defense plugin. Adding Feature Denoising to FFGSM (FFGSM + Denoising) does not improve the defense effect of FFGSM. This means that there can be cases where a combination of defense methods reduces the defense effect. Thus, one should be careful with combining defense methods.

Finally, we inspect example images for different imperceptibility constraints, see Figure 3.8. The six images have been perturbed by FGSM with an epsilon value increasing from 0 to 25. Below the image, we report the perturbation size by measuring the L^2 metric between the original



(a) Attacks with L^∞ norm. Defenses using L^∞.

(b) Attacks with L^2 norm. Defenses using L^∞.

Figure 3.7: Classification accuracy for adversarial test data generated by different attacks (FGSM, PGD, and AutoAttack) on different models (Base: no defense, Base + Detector [451], FFGSM [472]) for an increasing imperceptibility thresholds (eps).

and the perturbed image. As expected, the perturbation size is always lower than the epsilon bound. Even for an epsilon value of 4, one can already recognize artifacts in the image. So in this application, an epsilon of 4 would be an upper bound on the L^2 imperceptibility constraints.

RQ 1.3: Which areas of the hyperparameter space yield efficient evasion attacks for the COVID-19 data set?

Evasion attacks are optimization problems with perturbations of an image as the solution space and an objective that encodes the goals and constraints of an adversary. An adversarial attack is a custom-designed optimization method to find a solution to this constrained optimization problem. In most cases, one of the optimization constraints is an imperceptibility constraint, i.e., a measure that quantifies how strong the solution deviates from the original image. The

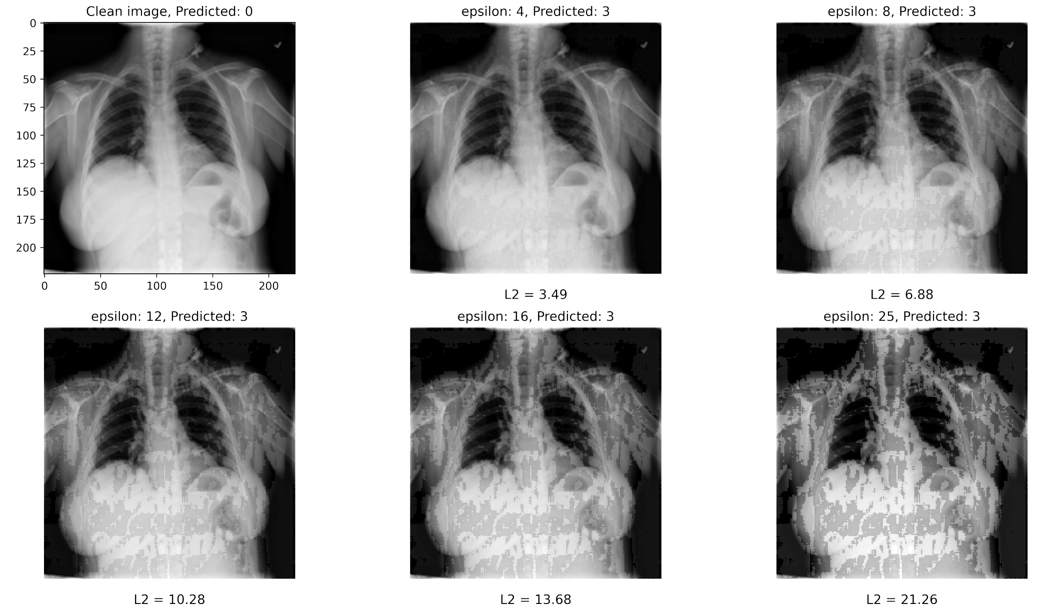


Figure 3.8: Visual inspection of imperceptibility constraint on COVID-19 4 class data set.

larger the deviation, the easier it is to perceive an adversarial image as such. Typically, one then correlates the imperceptibility measure, e.g., an L^2 distance, with an actual human perception to find a good cut-off value, i.e., the distance value where a human conceives an image as conspicuous. This value typically is application and data dependent, and finding the cutt-off value a laborious task.

Another dimension of interest are the solution quality, i.e., the success of a solution achieving the desired classification result. There are many conceivable success metrics one can use, e.g., a change count or a drop in accuracy compared to the original model evaluated on a test set. A third dimension is the runtime of an attack. Algorithmic runtimes are of practical relevance, since often the arms race between adversary and victim also is an economic question. If only resource-intensive attacks are successful, adversaries require a large computational budget which can be measured either in time to wait for an attack to complete or in money spend on computational resources. However, there are many moving parts when it comes to measuring computational budget, such as every-increasing hardware performance, and the difficulty of reliably estimating runtimes. Finding a good quantity to compare approaches often is difficult. Evasion attacks are framed as single objective optimization problems. However, the actual problem an adversary and a victim solve are multi-objective extensions of this problem: a search for an optimal trade-off between attack success, imperceptibility and computational budget. Adversaries and victims are interested in finding efficient solutions to this problem, i.e., a Pareto front. Which of the Pareto-optimal solutions they select is then up to subjective and economic objectives and constraints. For instance, one may constrain the search space to solutions that are imperceptible with respect to some cut-off evaluated by a human reference observer, and the computational budget is less than 5 minutes to complete an attack with a fixed hardware configuration.

A question remains: how does one find efficient attacks? Adversarial attacks typically come with a set of configurable parameters linked to the optimization method and the imperceptibility constraint. These parameters affect the resulting adversarial images and influence the runtime of an attack. One must vary these parameters to find pareto-fronts that are the basis for a decisions how to actually run or defend against an attack.

The adversarial machine learning research community has focused evaluating attacks on a few well-known benchmark data sets, e.g., MNIST, CIFAR-10, ImageNet. Thus, efficient configurations of adversarial attacks only are well-studied in relation to these few benchmarks. But insights can not be easily transferred to new data sets, like the COVID-19 chest-X-ray data. This is because visual cut-offs on the imperceptibility measure depend on the size and nature of the data set, and ultimately also on a human observer. Also the success of an attack and the effort required to find a good solution depends on data and model specifics. In this research question, we strive for an overview on how attack parameters influence COVID-19 detection results in all relevant dimensions: attack quality, imperceptibility and computational budget.

Setup. We study this question on the COVID-19, four-class model. We conduct two different experiments. In the first one, we set the imperceptibility constraint to $L^{inf} = 5$ and fix the number of iterations to 5 – values which we found to show decent attack success in previous experiments. We then measure the wall-clock time for the runtime of the attacks. We report results relative to the base model, i.e., an inference on the original image without running an attack. We also evaluate the attack success rate on the clean and perturbed images.

In the second one, we pick one of the attacks, AutoAttack, which often is seen as a default go-to attack because it has shown to perform well on a range of data sets and models. Here, we vary the type of Norm (L^∞ and L^2) as well as their ϵ thresholds. We run the attack for 1, 5 and 10 iterations and evaluate adversarial accuracy.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes)
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate without AMP
- **Runtime vs. Accuracy**
  - FGSM, PGD, PGD with Blind Spot, AutoAttack and Carlini Wagner with L^∞ norm with $\epsilon = 5$ and 5 iterations
- **AutoAttack Detail**
  - L^∞ with $\epsilon \in [4, 8, 15]$
  - L^2 with $\epsilon \in [1, 5, 10]$
  - Number of iterations in $[1, 5, 10]$

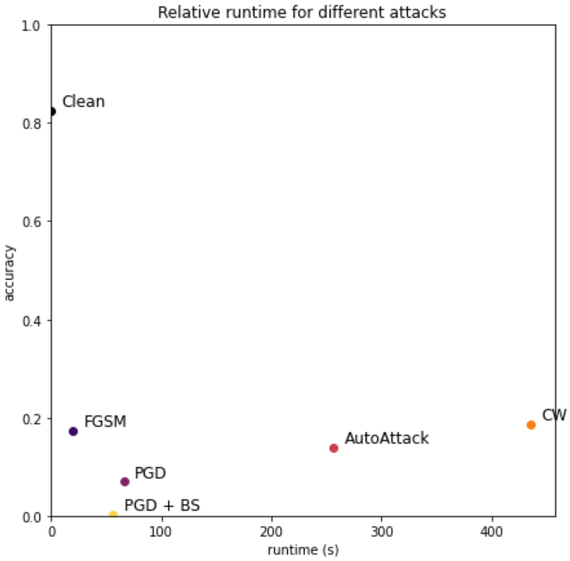


Figure 3.9: Runtime vs. accuracy for different attacks.

Results. Figure 3.9 graphs the runtime and accuracy of different attacks. One can see that all attacks are successful, they drop the predicted accuracy from above 0.8 to below 0.2. With respect to runtime, there are large differences. FGSM and PGD are quite fast compared to AutoAttack and Carlini Wagner. Also, neither AutoAttack or Carlini Wagner improve on the attack success. The Blind Spot decreases the accuracy of PGD further, the overall runtime even decreases slightly. The low complexity of the method, a small translation of the input image, and its low computational burden makes this a more efficient version of PGD. This means that in this experiment, only FGSM and PGD+BS are efficient; increasing the computational budget in form of longer runtimes does not pay off.

We now investigate the impact of the number of iterations on the adversarial accuracy for AutoAttack, see Figure 3.10. The figure shows the results for both L^∞ and L^2. A first observation is that both norms lead to successful results. For restrictive values ($L^\infty = 4$ and $L^2 = 1$), the adversarial accuracy still is quite high. For all other choices of metric values, we observe a significant drop for adversarial accuracy. After five iterations, all attacks yield perturbations where the adversarial accuracy is close to zero. One take-away from this experiment is that the number of iterations indeed has an effect on the adversarial accuracy. In fact, since the number of iterations is correlated with algorithmic runtimes, one can also infer that an increasing the computational budget auto-attack yields stronger attacks. However, this effect saturates quickly, after five iterations there is no further improvement. This is true for both restrictive imperceptibility constraints and loose ones.

Finally, one can also investigate the effect of an attack on an individual observation level. For this, we plot the distribution of attack success per observation as a boxplot. This is different to the evaluations before where we have only looked at the average success across observations.

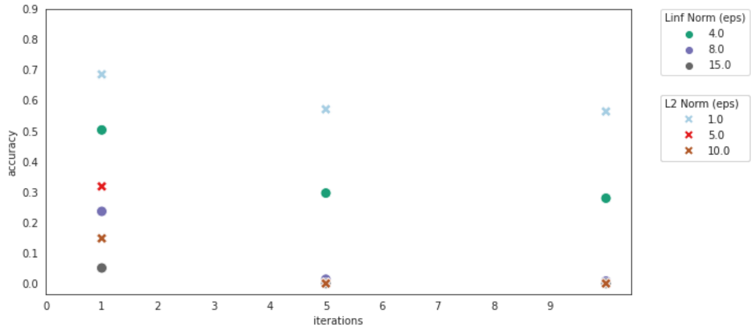


Figure 3.10: Impact of number of iterations on adversarial accuracy for AutoAttack.

Figure 3.11 shows the difference of the softmax score for the original and perturbed image for the class that was predicted for the original image. A high value on the y-axis means that the attack successfully changed the score for the original prediction. For instance, if the original prediction is the class "Covid" with a score of 0.9 on the original image and the prediction of the perturbed image reduces the score for the class "Covid" to 0.3, we have a difference in prediction score of 0.6. The plot also shows the worst case, i.e., the most unsuccessful attack per observation, and best case, i.e., the most successful attack per observation. One can see that PGD is producing attacks that are close to the best case. However, for all attacks, there are cases where the attacks fail to change the classification score, i.e., a difference close to zero. One take-away from this is that average values might not suffice as an evaluation if robustness is critical in all cases. One should evaluate attacks for individual observations and also visualize a distribution of the attack success over the data set.

3.4.2 Data Set Differences

RQ 2.1: Is the success of the different poisoning and adversarial attacks / defenses influenced by the data set choice (CIFAR-10 vs. COVID-19 two-class vs. COVID-19 four-class)?

One general question in the robustness domain is how attacks and defenses transfer between different data sets. Typically, one strives to correlate some data characteristics, such as data size and statistical properties of the features, to the attack success. If such correlations exist, one can choose appropriate attacks and defenses based on them. Correlations between data characteristics and attack success may also facilitate meta-learning to predict the outcome of an attack based on some historical executions without actually running it. However, this typically is a challenging endeavor since "data characteristics" are usually not well-defined and data sets are very diverse. Hence, we strive for some first indications on whether data sets actually play a role in the success of robustness methods.

Setup. In a first experiment, we compare the effect of defense methods on the accuracy of a model for different data sets. We train a base model which we poison with Bullseye Polytope

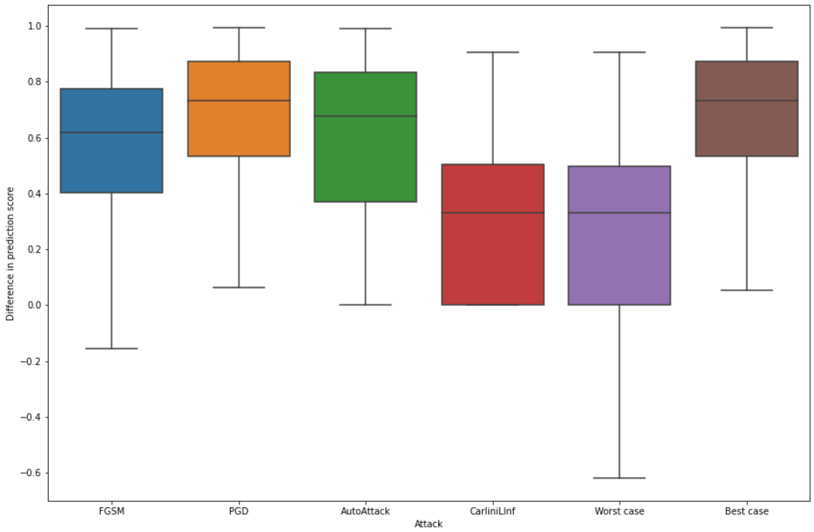


Figure 3.11: Difference of the classification softmax score for the original score and the adversarial score for the original predicted class (L^∞ constraint, epsilon 8/255 5 iterations.)

(base model), and two adapted version of the base model, one with a defense plugin (Matching Prediction Distributions) and one with adversarial training (FFGSM). We repeat this setup for all three data sets (COVID-19 with binary and with multi-class classification and CIFAR-10). We repeat each experiment five times with different random seeds account for non-determinism in the attack and defense methods.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes), COVID-19 (two classes), CIFAR-10
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate; without AMP for COVID, with AMP for CIFAR-10 with Imagenet weights
- **Poisoning Attacks**:
  - Bullseye Polytope
- **Defenses**:
  - FFGSM
  - Matching Prediction Distributions

Results. Figure 3.12 graphs the results. The result pattern is similar for all data sets: a defense plugin affects the model accuracy of a poisoned model much stronger than the adversarial retraining. This is not surprising, since the defense auto-encoder has been trained on the original

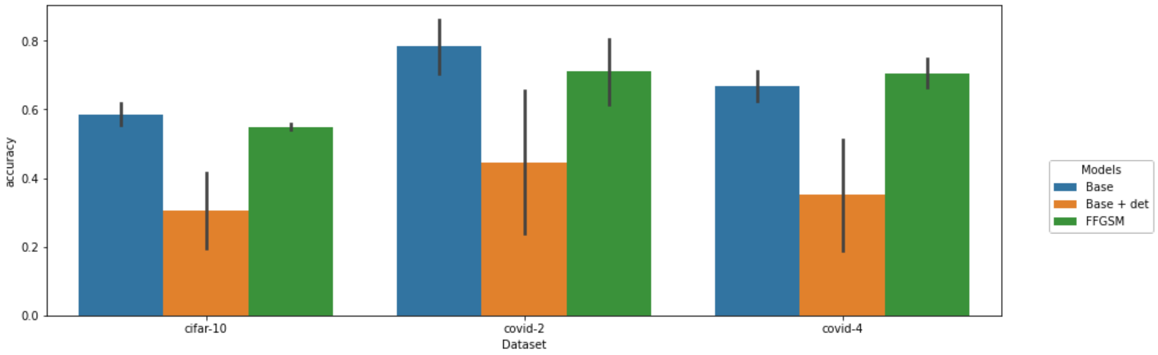


Figure 3.12: Comparison of average attack success on different data sets over five random repetitions. Error bars indicate one standard deviation.

model, which has a different distribution than the poisoned model. For the COVID four class model, the average accuracy even is slightly higher than without adversarial retraining. Further the defense plugin results in the largest standard deviation of all three model versions. A reason for this might be a high volatility of the auto-encoder that is trained as part of the defense. In summary, there are absolute differences in the model accuracy between data sets, but the different defense methods yield similar result patterns.

RQ 2.2: Does data set size play an important role for poisoning attacks and defenses?

An interesting data characteristic to look at is the size of a data set. One reason is that data size plays an important role in an iterative model development life cycle. Typically, models are trained on small development training sets since training times are faster and less resource intensive. Also, in many cases, collection of annotated data is laborious and expensive, so one strives to keep the data set as small as possible. This begs the question if results on a small subset of the data yields representative results.

Setup. For this question, we train models on different subsets of the COVID-19 data and look at how the success of evasion attack changes.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes)
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate without AMP
- **Adversarial Attacks**
  - FGSM
  - PGD

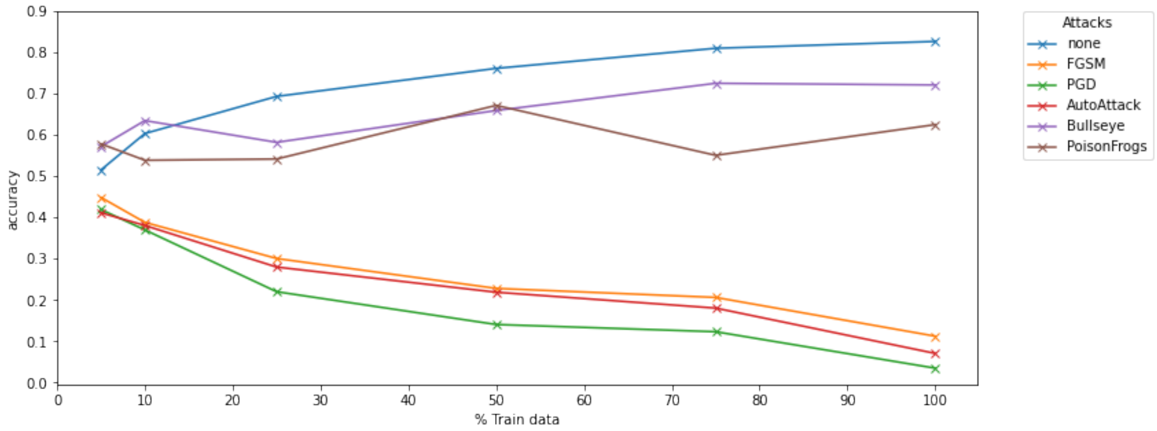


Figure 3.13: Comparison of model accuracy for different sizes of the training data set.

- AutoAttack
- Carlini Wagner
- Bullseye Polytope
- Poison Frogs

Results. Figure 3.13 graphs the development of accuracy with increasing training data size for different attacks. With a small training size of 5 %, the base model is around an accuracy of 0.5, and all evasion attacks slightly reduce the accuracy. With increasing training data size, poisoning and evasion attacks impact the accuracy differently. For a poisoned model, the accuracy increases with increasing data size, and the gap between the base model and the poisoned model stays almost the same. Here, the number of poisoned examples are fixed. This means that an increasing training data size does not mitigate the effect of a few poisoned examples. For evasion attacks, we observe that while the accuracy of the base model continuously increases, the accuracy of the attacked model decreases. The biggest gap is with the full training data. One conclusion from this result is that estimating the robustness of a model based on a small sample of the data might be misleading. Instead, one should evaluate adversarial accuracy on the full data set.

3.4.3 Dependencies between Evasion and Poisoning Robustness

RQ 3.1: Are poisoned models more or less susceptible to adversarial attacks?

Evasion and poisoning attacks both rely on exploiting the small distances between inputs and decision boundaries. However, for the intersection between both research areas, there has not been much research so far. There are many conceivable ways in which methods from both areas can interact. An example is to improve the effectiveness of generated backdoor triggers by employing a PGD attack on the generated poison example [322].

The focus of this research question is to combine the two notions of robustness, with respect to poisoning and evasion, to shed some light on the effects of these two attack vectors on each other. We expect insights on this questions to have an impact on the development of novel attack and defense strategies. For example, an adversary with access to the training data of the victim model might decide to use both poisoning and adversarial attacks if this increases the probability of a desired misclassification. Vice versa, if a joint application of adversarial and poisoning attacks decrease the attack success, an attacker will most likely separate the efforts into separate attack vectors. Finally, we believe that insights from both angles will be useful to the research area of adversarial machine learning. We expect combinations of methods to produce strong attacks or defense strategies which in turn will reveal existing vulnerabilities and lead to new research questions.

Setup. In our experiments we evaluate the success of L^∞-norm based adversarial attacks on a variety of COVID-19 detection models. The models include a choice of adversarial defenses and poisoning attacks during training. We choose FFGSM for adversarial training and Matching Prediction Distributions as an adversarial defense. For poisoning, we choose the Bullseye Polytope attack, a clean-label attack which tries to change the classification of images from a specific class, the poison class, to a desired target class. The basic idea is to perturb a number of images from the poison class such that their representation in the penultimate layer, the layer before softmax operation, matches the one of the desired target class. Similar to the evasion objective, perturbations should be imperceptible. In addition, the number of perturbed images should be low to minimize the chance of detecting the poisoning attack. In our experiments, we re-train the model on a poisoned version of the held-out validation split. We always apply the adversarial defense first, i.e., on unpoisoned data, and then execute the poisoning attack. Our choice is arbitrary, and we leave a combinatorial study on the impact of ordering evasion and poisoning methods to future work.
As before, we run a set of L^∞-norm based adversarial attacks on the unseen test data, in particular FGSM, PGD, AutoAttack and Carlini & Wagner. All attacks are executed with a fixed attack budget ($\epsilon = 8/255$). For Carlini & Wagner, we apply both a plain version of the attack and a combination with the Blind-Spot (BS) attack. In the combination, the Blind-Spot attack scales and shifts the original image before applying the Carlini & Wagner attack. Note that the resulting perturbed images are usually further away from the starting image than the maximal L^∞-norm distance given by the fixed ϵ-value.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes)
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate, with / without adversarial training, with / without detector, with / without poisoning attack
- **Adversarial Defenses**

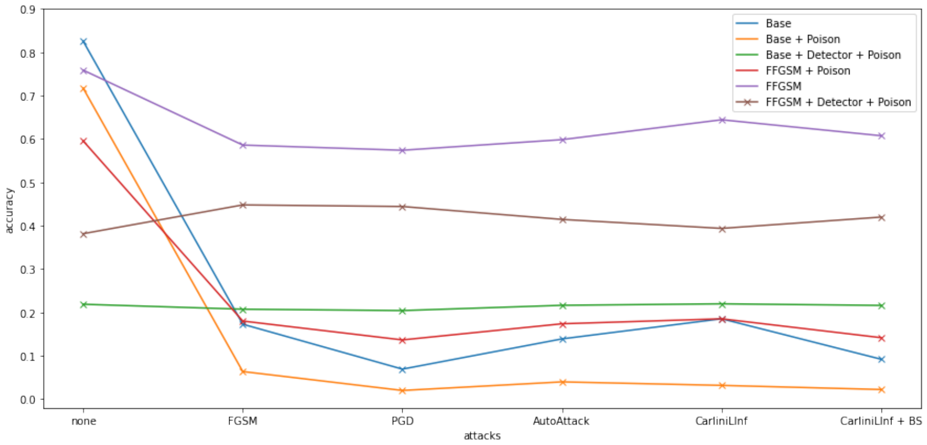


Figure 3.14: Classification accuracy for adversarial test data generated by different attacks (FGSM, PGD, AutoAttack, Carlini & Wagner, Carlini & Wagner combined with Blind-Spot) on different poisoned and non-poisoned models for a fixed L^∞-norm imperceptibility threshold ($\epsilon = 8/255$). Here, ”Poison” refers to the Bullseye Polytope attack [5], “Detector” is the Matching Prediction Distributions autoencoder [451] and “FFGSM” [152] is an adversarial training method.

- FFGSM - fix one ϵ-value for all models during training ($8/255$ as ϵ-value)
- Matching Prediction Distribution with adaptive Kullback-Leibler distance

- **Poisoning Attacks**
  - Bullseye Polytope - 60 iterations, 10 sample points from poison label ($= 1$) and target label ($= 0$), retraining on validation data (10 retraining epochs)
- **Adversarial Attacks**
  - FGSM, PGD, AutoAttack, Carlini & Wagner and Carlini & Wagner combined with Blind Spot method, all with L^∞-norm constraint for fixed ϵ-value ($8/255$ as ϵ-value)

Results. Figure 3.14 shows the prediction accuracy for different models and different attacks. A first observation is that all poison and defense methods reduce the accuracy compared to a non-attacked base model. Also the combination of the Matching Prediction Distributions (Detector) defense with the Bullseye Polytope (Poison) attack leads to very poor classification quality. Both variants, with and without FFGSM adversarial training, result in models of an accuracy below 40%. Independent FFGSM training yields the most robust model with an adversarial accuracy above 60% for all executed adversarial attacks.

However, when the FFGSM adversarially trained model is additionally attacked with the Bullseye Polytope attack, the vulnerability to adversarial attacks increases drastically. One can see this by comparing the “Base + Poison” with the “FFGSM + Poison”. The poisoned model (FFGSM

+ Poison) is only slightly more robust than the base model (Base + Poison) without any adversarial defense. The poisoning attack also decreases the accuracy of the base model (Base + Poison) without any attack. We conclude that the Bullseye Polytope attack does in fact i) reduce the accuracy of a non-attacked model and ii) decrease the robustness to adversarial examples.
Overall, our results suggest that poisoned models are more susceptible to adversarial attacks. Thus, a combination of poisoning and evasion attacks might bring up the effectiveness of the evasion over an isolated attack.

RQ 3.2: Do adversarial defenses influence the success of poisoning attacks?

One can also look at interaction effects from a defense perspective, i.e., if defense methods have an effect on poisoning attacks. The motivation is analog to the previous research question: experiments on the interaction might shed some light on which combinations of methods are beneficial for a defender or adversary.

Setup. The setup is similar to the experiments described in Section 3.4.1. We compare the confusion matrix of two models, both with FFGSM training and one a Bullseye attack. This time, however, we fix the Bullseye Polytope hyperparameters.

Experiment Setup:

- **Model**: ResNeXt-50
- **Data**: COVID-19 (four classes)
- **Training**: 10 training epochs, SGD optimizer and cyclic learning rate, with / without poisoning attack
- **Poisoning Attacks**:
  - Bullseye Polytope, epsilon=8/255, 60 iterations, 10 samples
  - Poison Frogs (Feature Collision), epsilon=8/255, 60 iterations, 10 samples
- **Adversarial Training**: FFGSM

Results. We compare the confusion matrix of the three models, see Table 3.3.[3] We can see that the Bullseye Polytope (Table 3.3b) indeed changes the classification in a few cases from 11 predicted COVID to 222 COVID predictions. However, also the distributions in the other classes change, e.g., the number of Lung Opacity reduces from 275 to 144. For Poison Frogs, we see a similar pattern, see Table 3.3c. Here, the number of COVID predictions increases from 11 to 255; other class distributions change as well. We can conclude that both Bullseye Polytope and Poison Frogs have an effect for models that use adversarial training. However, the Bullseye Polytope attack is not reliable, see Section 3.4.1. We found both, an example where a plain Bullseye

[3] For results without adversarial defense, see Table 3.1 and Table 3.2.

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	483	1	3	25	512
COVID	116	9	0	51	176
Viral Pneumonia	13	0	28	2	43
Lung Opacity	68	1	3	197	269
	680	11	34	275	1000

(a) FFGSM training. Weighted F1 Score = 0.657

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	446	36	19	11	512
COVID	39	121	1	15	176
Viral Pneumonia	4	5	34	0	43
Lung Opacity	81	60	10	118	269
	570	222	64	144	1000

(b) FFGSM training + Bullseye Polytope. Weighted F1 Score = 0.710

	Predicted				
	Normal	COVID	Viral Pneumonia	Lung Opacity	
Normal	357	86	9	60	512
COVID	20	122	1	33	176
Viral Pneumonia	2	7	27	7	43
Lung Opacity	32	40	5	192	269
	411	255	42	292	1000

(c) FFGSM training + Poison Frogs. Weighted F1 Score: 0.707

Table 3.3: Confusion matrices for the FFGSM adversarial training with Bullseye Polytope and Poison Frogs attack. The poisoning was done with poison label "COVID" and target label "Normal".

Polytope attack is more successful (cf. Table 3.1c) and one where it is less successful (cf. Table 3.1c) than the one with FFGSM training (Table 3.3b). So the overall impact of adversarial training on Bullseye Polytope remains inconclusive.

3.5 Discussion and Conclusions

Robustness of machine learning models is a competition between adversaries and defenders to attack and guard against vulnerabilities. Significant research has gone into developing methods on changing classification results under different threat models. However, most approaches are evaluated in an isolated way, which begs the question of how well they perform in practical settings where many different attack vectors are applicable. Such insights on interaction effects are useful not only to assess the capabilities of state-of-the-art methods, but more so to guide further research into stronger attacks and defenses by stacking existing methods.

In this scientific report, we focus on research questions on the effects of hyperparameter choices, data set characteristics, and combination of different methods on the robustness of classification models. Our experimental study comprises different data sets, classification models, and a variety of evasion and poisoning attacks as well as defenses. This results in a huge experimental space which is intractable to cover exhaustively with detailed results. Instead, we focus on areas which we deem interesting and fruitful to reveal previously unseen interaction effects between methods.

By hand-selecting interesting areas of the experimental space, one runs the danger of missing interesting aspects. So a different approach is to guide the search for efficient attacks and defenses by the dimensions that are relevant to attacker/defender: the attack success, the computational budget and imperceptibility constraints. Any attack/defense combination can be placed and compared with each other: a method A is better than method B if it is better in one of these dimensions and at least as good as B in all other dimensions. If method A has higher attack success rate and higher imperceptibility than method B, but requires larger computational budget, both A and B are Pareto optimal. One can then frame an optimal attack configuration as a multi-objective optimization. Attackers and victims may then choose a trade-off along the Pareto frontier that fits their constraints, e.g., an available computational budget. A multi-objective optimization problem also gives way to a guided search through the experimental space. One option would be the use of evolutionary algorithms to find a diverse set of solutions along the Pareto front. This leads to actionable outcomes such as: “The best model with respect to adversarial protection on data set X, one should use training procedure Y for the base model with parameters Φ^Y and apply defense method Z with parameters Φ^Z.”

A challenge that remains with the multi-objective approach is the exponential search space and the computational cost of an evolutionary optimization through the search space of potential models, attack and defense configurations. Another challenge is the decision of how to quantify the relevant dimensions. As an example, there exist multiple ways to quantify computational effort, e.g., by runtime measurements (CPU/GPU time, wall-clock time), counting the number of gradient operations or algorithm iterations. The same holds for measuring model quality and robustness. In our study, we focus on adversarial accuracy. However, adversarial accuracy assumes a static setting where an adversary has only one shot in creating adversarial examples. This means that it systematically underestimates an adversary that employs adaptive attacks.

Our experimental study shows that moving from single attacks to combinations across research areas can be fruitful. Approaches like AutoAttack [97] already take a step into this direction by replacing a single-attack perspective with an ensemble of parameter-free attacks. We expect that the exploration of interactions between methods and elaborate search approaches through the attack configuration space will play an important role in adversarial machine learning.

Summary and Conclusion

The field of adversarial ML is a fast developing and broad research branch. The amount of scientific publications in this area is growing exponentially (see Figure 3.15), which makes selecting suitable protection methods for AI systems a complex task.

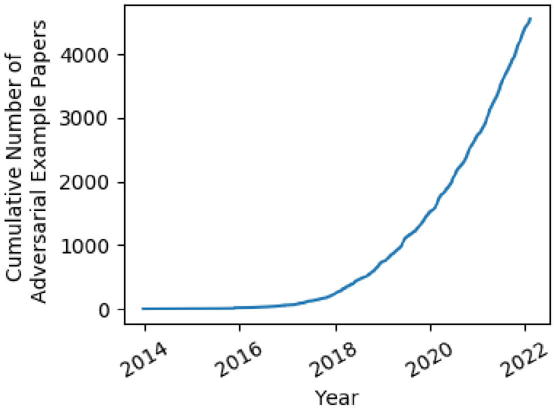


Figure 3.15: Amount of papers on adversarial research. Taken from `https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html`.

In particular, we identified the following challenges when developing resilient AI systems: the required choice of hyperparameters, the amount of existing as well as possible new attacks, and the difficulty of the transferability of methods between data sets and domains. Due to their deployment even in security-critical environments, methods providing trustworthy and resilient models are becoming more critical and obligatory even on a law level (e.g., according to GDPR). We find that verification and robustness certification methods are promising research directions to overcome the aforementioned problems with respect to evasion attacks. These approaches aim at providing a verified robust model without any further defenses and tuning. However, current methods are still limited in applicability and more research is needed to expand them to a broader range of models and to address the issue of certificate spoofing. Moreover, the ultimate goal is to obtain a certified model that is resilient also with respect to other types of attacks, which is not possible yet. Currently, the most feasible approach for defending the system fully is (i) to precisely analyze which possible threats are present in the particular use case (Section 1.1), and (ii) to have an up-to-date list of defenses and best practices (Section 1.2).

In this document, we provide such best practice guidelines based on an extensive adversarial ML literature review and our experimental framework. Our guidelines are intended to help navigate the vast field of adversarial ML. In particular, we raise awareness for threats along the AI life cycle and enable practitioners to select countermeasures that increase the resilience of their systems. This is a crucial step towards a more reliable and secure application of AI systems. As our work highlights possible limitations of the proposed defenses it allows for a realistic assessment of unmitigated risks. The application of state-of-the-art defenses as proposed here, together with adopting common IT security measures – while not guaranteeing full protection – will make the task of attackers significantly more complex and forces them to invest more resources. Thus, it helps to increase resilience of the models.

Bibliography

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In CCS (2016).

[2] Abbasi, M., and Gagne, C. Robustness to adversarial examples through an ensemble of specialists. In ICLR (Workshop Track) (2017).

[3] Adi, Y., Baum, C., Cisse, M., Pinkas, B., and Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In USENIX Security Symposium (2018).

[4] Agarwal, A., Singh, R., and Vatsa, M. The role of sign and direction of gradient on the performance of cnn. In CVPR Workshop (2020).

[5] Aghakhani, H., Meng, D., Wang, Y.-X., Kruegel, C., and Vigna, G. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In EuroS&P (2021).

[6] Aivodji, U., Bolot, A., and Gambs, S. Model extraction from counterfactual explanations. In arXiv (2020).

[7] Akintunde, M. E., Kevorchian, A., Lomuscio, A., and Pirovano, E. Verification of rnn-based neural agent-environment systems. In AAAI (2019).

[8] Al-Dujaili, A., and OReilly, U.-M. Sign bits are all you need for black-box attacks. In ICLR (2020).

[9] Alaifari, R., Alberti, G. S., and Gauksson, T. Adef: An iterative algorithm to construct adversarial deformations. In ICLR (2019).

[10] Amir, G., Wu, H., Barrett, C., and Katz, G. An smt-based approach for verifying binarized neural networks. In arXiv (2020).

[11] Anderson, G., Pailoor, S., Dillig, I., and Chaudhuri, S. Optimization and abstraction: a synergistic approach for analyzing neural network robustness. In ACM SIGPLAN (2019).

[12] Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In ECCV (2020).

[13] Ashtiani, H., Pathak, V., and Urner, R. Black-box certification and learning under adversarial perturbations. In ICML (2020).

[14] Ateniese, G., Felici, G., Mancini, L. V., Spognardi, A., Villani, A., and Vitali, D. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. In International Journal of Security and Networks 10.3 (2015).

[15] Athalye, A., and Carlini, N. On the robustness of the CVPR 2018 white-box adversarial example defenses. CoRR abs/1804.03286 (2018).

[16] Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In ICML (2018).

[17] Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In ICML (2018).

[18] Atli, B. G., Szyller, S., Juuti, M., Marchal, S., and Asokan, N. Extraction of complex dnn models: Real threat or boogeyman. In International Workshop on Engineering Dependable and Secure Machine Learning Systems (2020).

[19] Awasthi, P., Jain, H., Rawat, A. S., and Vijayaraghavan, A. Adversarial robustness via robust low rank representations. In NeurIPS (2020).

[20] Bafna, M., Murtagh, J., and Vyas, N. Thwarting adversarial examples: An l0-robust sparse fourier transform. In NeurIPS (2018).

[21] Bagdasaryan, Eugene, and Shmatikov, V. Blind backdoors in deep learning models. In arXiv (2020).

[22] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. In AISTATS (2020).

[23] Bai, T., Luo, J., and Zhao, J. Recent advances in understanding adversarial robustness of deep neural networks. In arXiv (2021).

[24] Bai, Y., Zeng, Y., Jiang, Y., Xia, S.-T., Ma, X., and Wang, Y. Improving adversarial robustness via channel-wise activation suppressing. In ICLR (2021).

[25] Baluja, S., and Fischer, I. Adversarial transformation networks: Learning to generate adversarial examples. In arXiv (2017).

[26] Balunovic, M., Baader, M., Singh, G., Gehr, T., and Vechev, M. Certifying geometric robustness of neural networks. In NeurIPS (2020).

[27] Barbalau, A., Cosma, A., Ionescu, R. T., and Popescu, M. Black-box ripper: Copying black-box models using generative evolutionary algorithms. In NeurIPS (2020).

[28] Barni, M., Kallas, K., and Tondi, B. A new backdoor attack in cnns by training set corruption without label poisoning. In ICIP (2019).

[29] Baruch, M., Baruch, G., and Goldberg, Y. A little is enough: Circumventing defenses for distributed learning. In NeurIPS (2019).

[30] Batina, L., Bhasin, S., Jap, D., and Picek, S. Csi nn: Reverse engineering of neural network architectures through electromagnetic side channel. In USENIX (2019).

[31] Berghoff, C., Bielik, P., Neu, M., Tsankov, P., and von Twickel, A. Robustness testing of AI systems: A case study for traffic sign recognition. CoRR abs/2108.06159 (2021).

[32] Berrada, L., Dathathri, S., Stanforth, R., Bunel, R., Uesato, J., Gowal, S., and Kumar, M. P. Verifying probabilistic specifications with functional lagrangians. In arXiv (2021).

[33] Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. Analyzing federated learning through an adversarial lens. In ICML (2019).

[34] Bhagoji, A. N., He, W., Li, B., and Song, D. Practical black-box attacks on deep neural networks using efficient query mechanisms. In ECCV (2018).

[35] Bhalerao, A., Kallas, K., Tondi, B., and Barni, M. Luminance-based video backdoor attack against anti-spoofing rebroadcast detection. In IEEE 21st International Workshop on Multimedia Signal Processing (MMSP) (2019).

[36] Bhattad, A., Chong, M. J., Liang, K., Li, B., and Forsyth, D. A. Unrestricted adversarial examples via semantic manipulation. In ICLR (2020).

[37] Biggio, B., and Rolia, F. Wild patterns: Ten years after the rise of adversarial machine learning. In Pattern Recognition (journal) (2018).

[38] Bonaert, G., Dimitrov, D. I., Baader, M., and Vechev, M. Fast and precise certification of transformers. In ACM SIGPLAN International Conference on Programming Language Design and Implementation (2021).

[39] Boopathy, A., Weng, T.-W., Chen, P.-Y., Liu, S., and Daniel, L. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In AAAI (2019).

[40] Borgnia, E., Cherepanova, V., Fowl, L., Ghiasi, A., Geiping, J., Goldblum, M., Goldstein, T., and Gupta, A. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In ICASSP (2021).

[41] Bose, Joey, A., and Aarabi, P. Adversarial attacks on face detectors using neural net based constrained optimization. In 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP) (2018).

[42] Botoeva, E., Kouvaros, P., Kronqvist, J., Lomuscio, A., and Misener, R. Efficient verification of relu-based neural networks via dependency analysis. In AAAI (2020).

[43] Bourse, F., Minelli, M., Minihold, M., and Paillier, P. Fast homomorphic evaluation of deep discretized neural networks. In CRYPTO (2018).

[44] Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In ICLR (2018).

[45] Brendel, W., Rauber, J., Kummerer, M., Ustyuzhaninov, I., and Bethge, M. Accurate, reliable and fast robustness evaluation. In NeurIPS (2019).

[46] Brown, T. B., Mane, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch. In NeurIPS Workshop 2017 (2017).

[47] Brunner, T., Diehl, F., Le, M. T., and Knoll, A. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In ICCV (2019).

[48] Bryniarski, O., Hingun, N., Pachuca, P., Wang, V., and Carlini, N. Evading adversarial example detection defenses with orthogonal projected gradient descent. In arXiv (2021).

[49] B.S., V., and Babu, R. V. Single-step adversarial training with dropout scheduling. In CVPR (2020).

[50] Buckman, J., Roy, A., Raffel, C., and Goodfellow, I. Thermometer encoding: One hot way to resist adversarial examples. In ICLR (2018).

[51] Bunel, R., Lu, J., Turkaslan, I., Torr, P. H., Kohli, P., and Kumar, M. P. Branch and bound for piecewise linear neural network verification. In arXiv (2019).

[52] Bunel, R., Palma, A. D., Desmaison, A., Dvijotham, K., Kohli, P., Torr, P., and Kumar, M. P. Lagrangian decomposition for neural network verification. In Conference on Uncertainty in Artificial Intelligence (2020).

[53] Bunel, R., Turkaslan, I., Torr, P. H., Kohli, P., and Kumar, M. P. A unified view of piecewise linear neural network verification. In NeurIPS (2018).

[54] Cai, Q.-Z., Du, M., Liu, C., and Song, D. Curriculum adversarial training. In IJCAI (2018).

[55] Cao, X., and Gong, N. Z. Mitigating evasion attacks to deep neural networks via region-based classification. In ACSAC (2017).

[56] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. On evaluating adversarial robustness. In arXiv (2019).

[57] Carlini, N., Deng, S., Garg, S., Jha, S., Mahloujifar, S., Mahmoody, M., Song, S., Thakurta, A., and Tramer, F. Is private learning possible with instance encoding. In S&P (2021).

[58] Carlini, N., Jagielski, M., and Mironov, I. Cryptanalytic extraction of neural network models. In CRYPTO (2020).

[59] Carlini, N., Katz, G., Barrett, C., and Dill, D. L. Provably minimally-distorted adversarial examples (previous name: Ground-truth adversarial examples). In arXiv (2018).

[60] Carlini, N., Liu, C., Erlingsson, U., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In USENIX (2019).

[61] Carlini, N., and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In AISec (2017).

[62] Carlini, N., and Wagner, D. Towards evaluating the robustness of neural networks. In S&P (2017).

[63] Carlini, N., and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. In SPW (IEEE Security and Privacy Workshops) (2018).

[64] Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., and Duchi, J. C. Unlabeled data improves adversarial robustness. In NeurIPS (2019).

[65] Chabanne, H., de Wargny, A., Milgram, J., Morel, C., and Prouff, E. Privacy-preserving classification on deep neural network. In IACR (2017).

[66] Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S., and Yan, S. Exploring connections between active learning and model extraction. In USENIX (2020).

[67] Chaubey, A., Agrawal, N., Barnwal, K., Guliani, K. K., and Mehta, P. Universal adversarial perturbations: A survey. In arXiv (2020).

[68] Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., and Srivastava, B. Detecting backdoor attacks on deep neural networks by activation clustering. In SafeAI@AAAI (2019).

[69] Chen, C.-L., Golubchik, L., and Paolieri, M. Backdoor attacks on federated meta-learning. In NeurIPS (2020).

[70] Chen, D., Yu, N., Zhang, Y., and Fritz, M. Gan-leaks: A taxonomy of membership inference attacks against generative models. In 2020 ACM SIGSAC (2020).

[71] Chen, H., Fu, C., Zhao, J., and Koushanfar, F. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In IJCAI (2019).

[72] Chen, H., Zhang, H., Chen, P.-Y., Yi, J., and Hsieh, C.-J. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In ACL (2018).

[73] Chen, J., Jordan, M. I., and Wainwright, M. J. Hopskipjumpattack: A query-efficient decision-based attack. In S&P (2020).

[74] Chen, P.-Y., Sharma, Y., Zhang, H., Y, J., and Hsieh, C.-J. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In AAAI (2018).

[75] Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In AISec (2017).

[76] Chen, S., Carlini, N., and Wagner, D. Stateful detection of black-box adversarial attacks. In SPAI (Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence at Asia-CCS) (2020).

[77] Chen, S., Wong, E., Kolter, J. Z., and Fazlyab, M. Deepsplit: Scalable verification of deep neural networks via operator splitting. In arXiv (2021).

[78] Chen, S.-T., Cornelius, C., Martin, J., and Chau, D. H. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In ECMLPKDD (2018).

[79] Chen, X., Salem, A., Backes, M., Ma, S., and Zhang, Y. Badnl: Backdoor attacks against nlp models. In arXiv (2020).

[80] Cheng, C.-H., Huang, C.-H., Brunner, T., and Hashemi, V. Towards safety verification of direct perception neural networks. In Design, Automation & Test in Europe Conference & Exhibition (DATE) (2020).

[81] Cheng, C.-H., Nuhrenberg, G., and Ruess, H. Maximum resilience of artificial neural networks. In ATVA (2017).

[82] Cheng, M., Le, T., Chen, P.-Y., Zhang, H., Yi, J., and Hsieh, C.-J. Query-efficient hard-label black-box attack: An optimization-based approach. In ICLR (2019).

[83] Cheng, S., Liu, Y., Ma, S., and Zhang, X. Deep feature space trojan attack of neural networks by controlled detoxification. In AAAI Conference on Artificial Intelligence (2021).

[84] Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Advances in Cryptology – ASIACRYPT 2016 (Berlin, Heidelberg, 2016), J. H. Cheon and T. Takagi, Eds., Springer Berlin Heidelberg, pp. 3–33.

[85] Chou, E., Tramer, F., and Pellegrino, G. Sentinet: Detecting localized universal attacks against deep learning systems. In IEEE Symposium on Security and Privacy Workshops (2020).

[86] Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. A., Reaz, M. B. I., and Islam, M. T. Can ai help in screening viral and covid-19 pneumonia? IEEE Access 8 (2020), 132665–132676.

[87] Cisse, M. M., Adi, Y., Neverova, N., and Keshet, J. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In NeurIPS (2017).

[88] Clements, J., and Lao, Y. Hardware trojan attacks on neural networks. In arXiv (2018).

[89] COMMISSION, E. On artificial intelligence - a european approach to excellence and trust. Whitepaper, Brussels, 2020.

[90] Commission, E., Directorate-General for Communications Networks, C., and Technology. Ethics guidelines for trustworthy AI. Publications Office, 2019.

[91] Correia-Silva, J. R., Berriel, R. F., Badue, C., de Souza, A. F., and Oliveira-Santos, T. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In IJCNN (2018).

[92] Costales, R., Mao, C., Norwitz, R., Kim, B., and Yang, J. Live trojan attacks on deep neural networks. In CVPR Workshop (2020).

[93] Croce, F., Andriushchenko, M., and Hein, M. Provable robustness of relu networks via maximization of linear regions. In AISTATS (2020).

[94] Croce, F., Andriushchenko, M., Singh, N. D., Flammarion, N., and Hein, M. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In arXiv (2020).

[95] Croce, F., and Hein, M. Sparse and imperceivable adversarial attacks. In ICCV (2019).

[96] Croce, F., and Hein, M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In ICML (2020).

[97] Croce, F., and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In ICML (2020).

[98] Croce, F., Rauber, J., and Hein, M. Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks. In IJCV (2019).

[99] Dai, J., and Chen, C. A backdoor attack against lstm-based text classification systems. In IEEE (2019).

[100] Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Li, S., Chen, L., Kounavis, M. E., and Chau, D. H. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In KDD (2018).

[101] Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R., and Shankar, S. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In arXiv (2020).

[102] Davaslioglu, K., and Sagduyu, Y. E. Trojan attacks on wireless signal classification with adversarial machine learning. In IEEE International Symposium on Dynamic Spectrum Access Networks (2019).

[103] Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. In ICLR (2018).

[104] Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. In ICLR (2020).

[105] Doan, B. G., Abbasnejad, E., and Ranasinghe, D. C. Februus: Input purification defense against trojan attacks on deep neural network systems. In ACSAC (2020).

[106] Dong, Y., Bao, F., Su, H., and Zhu, J. Towards interpretable deep neural networks by leveraging adversarial examples. In AAAI Workshop on Interpretability for Deep Learning (2019).

[107] Dong, Y., Deng, Z., Pang, T., Su, H., and Zhu, J. Adversarial distributional training for robust deep learning. In NeurIPS (2020).

[108] Dong, Y., Liao, F., Pang, T., Hu, X., and Zhu, J. Discovering adversarial examples with momentum. CVPR (2018).

[109] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In CVPR (2018).

[110] Dong, Y., Pang, T., Su, H., and Zhu, J. Evading defenses to transferable adversarial examples by translation-invariant attacks. In CVPR (2019).

[111] Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., and Zhu, J. Efficient decision-based black-box adversarial attacks on face recognition. In CVPR (2019).

[112] Du, T., Ji, S., Shen, L., Zhang, Y., Li, J., Shi, J., Fang, C., Yin, J., Beyah, R., and Wang, T. Cert-rnn: Towards certifying the robustness of recurrent neural networks. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (New York, NY, USA, 2021), CCS '21, Association for Computing Machinery, p. 516–534.

[113] Duddu, V., Samanta, D., Rao, D. V., and Balas, V. E. Stealing neural networks via timing side channels. In arXiv (2018).

[114] Dvijotham, K. D., Hayes, J., Balle, B., Kolter, J. Z., Qin, C., Gyorgy, A., Xiao, K., Gowal, S., and Kohli, P. A framework for robustness certification of smoothed classifiers using f-divergences. In ICLR (2020).

[115] Dvijotham, K. D., Stanforth, R., Gowal, S., Mann, T., and Kohli, P. A dual approach to scalable verification of deep networks. In arXiv (2018).

[116] Dvijotham, K. D., Stanforth, R., Gowal, S., Qin, C., De, S., and Kohli, P. Efficient neural network verification with exactness characterization. In UAI (2020).

[117] Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. Hotflip: White-box adversarial examples for text classification. In ACL (2018).

[118] Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. In arXiv (2017).

[119] Elboher, Y. Y., Gottschlich, J., and Katz, G. An abstraction-based framework for neural network verification. In International Conference on Computer Aided Verification (2020).

[120] Engstrom, L., Ilyas, A., and Athalye, A. Evaluating and understanding the robustness of adversarial logit pairing. In NeurIPS SECML (2018).

[121] Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In ICML (2019).

[122] Etmann, C., Lunz, S., Maass, P., and Schoenlieb, C. On the connection between adversarial robustness and saliency map interpretability. In ICML (2019).

[123] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., Kohno, T., and Song, D. Physical adversarial examples for object detectors. In 12th USENIX Workshop on Offensive Technologies (WOOT 18) (2018).

[124] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In CVPR (2018).

[125] Fang, M., Cao, X., Jia, J., and Gong, N. Z. Local model poisoning attacks to byzantine-robust federated learning. In USENIX Security Symposium (2020).

[126] Fawzi, A., and Frossard, P. Manitest: Are classifiers really invariant. In BMVC (British Machine Vision Conference) (2015).

[127] Fawzi, A., and Frossard, P. Measuring the effect of nuisance variables on classifiers. In BMVC (British Machine Vision Conference) (2016).

[128] Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. Empirical study of the topology and geometry of deep networks. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018), pp. 3762–3770.

[129] Fazlyab, M., Morari, M., and Pappas, G. J. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. In arXiv (2020).

[130] Feinman, R., Curtin, R. R., and Gardner, S. S. A. B. Detecting adversarial samples from artifacts. In - (2017).

[131] Fidel, G., Bitton, R., and Shabtai, A. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In IJCNN (2020).

[132] Fredrikson, M., Jha, S., and Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In ACM SIGSAC (2015).

[133] Freitas, S., Chen, S.-T., Wang, Z. J., and Chau, D. H. Unmask: Adversarial detection and defense through robust feature alignment. In IEEE Big Data (2020).

[134] Fromherz, A., Leino, K., Fredrikson, M., Parno, B., and Pasareanu, C. Fast geometric projections for local robustness certification. In ICLR (2021).

[135] Fung, Clement, Yoon, C. J., and Beschastnikh, I. Understanding and improving fast adversarial training. In NeurIPS (2020).

[136] Fung, C., Yoon, C. J. M., and Beschastnikh, I. The limitations of federated learning in sybil settings. In 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID) (2020).

[137] Gao, L., Zhang, Q., Song, J., Liu, X., and Shen, H. T. Patch-wise attack for fooling deep neural network. In ECCV (2020).

[138] Gao, Y., Doan, B. G., Zhang, Z., Ma, S., Zhang, J., Fu, A., Nepal, S., and Kim, H. Backdoor attacks and countermeasures on deeplearning: A comprehensive review. In arXiv (2020).

[139] Gao, Y., Xu, C., Wang, D., Chen, S., C.Ranasinghe, D., and Nepal, S. Strip: a defence against trojan attacks on deep neural networks. In ACSAC (2019).

[140] Garg, S., Kumar, A., Goel, V., and Liang, Y. Can adversarial weight perturbations inject neural backdoors. In CIKM (ACM International Conference on Information & Knowledge Management) (2020).

[141] Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. Ai: Safety and robustness certification of neural networks with abstract interpretation. In S&P (2018).

[142] Geiping, J., Fowl, L., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. Witches brew: Industrial scale data poisoning via gradient matching. In arXiv (2020).

[143] Geiping, J., Fowl, L., Somepalli, G., Goldblum, M., Moeller, M., and Goldstein, T. What doesnt kill you makes you robust(er): Adversarial training against poisons and backdoors. In arXiv (2021).

[144] Gentry, C. Fully homomorphic encryption using ideal lattices. In STOC (2009).

[145] Ghiasi, Amin, Shafahi, A., and Goldstein, T. Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates. In ICLR (2020).

[146] Ghodsi, Z., Gu, T., and Garg, S. Safetynet: Detecting and rejecting adversarial examples robustly. In NeurIPS (2017).

[147] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In ICML (2016).

[148] Gilmer, J., Adams, R. P., Goodfellow, I., Andersen, D., and Dahl, G. E. Motivating the rules of the game for adversarial example research. In arXiv (2018).

[149] Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., and Russell, S. Adversarial policies: Attacking deep reinforcement learning. In ICLR (2020).

[150] Gong, Zhenqiang, N., and Liu, B. You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors. In USENIX Security Symposium (2016).

[151] Gong, Z., Wang, W., and Ku, W. Adversarial and clean data are not twins. CoRR abs/1704.04960 (2017).

[152] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In ICLR (2015).

[153] Goswami, G., Ratha, N., Agarwal, A., Singh, R., and Vatsa, M. Unravelling robustness of deep learning based face recognition against adversarial attacks. In Proceedings of the AAAI Conference on Artificial Intelligence (2018).

[154] Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. In arXiv (2017).

[155] Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. Badnets: Evaluating backdooring attacks on deep neural networks. In IEEE Access (Journal) (2019).

[156] Guo, C., Gardner, J., You, Y., Wilson, A. G., and Weinberger, K. Simple black-box adversarial attacks. In ICML (2019).

[157] Guo, C., Rana, M., Cisse, M., and van der Maaten, L. Countering adversarial images using input transformations. In ICLR (2018).

[158] Guo, W., Wang, L., Xing, X., Du, M., and Song, D. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. In arXiv (2019).

[159] Hartl, A., Bachl, M., Fabini, J., and Zseby, T. Explainability and adversarial robustness for rnns. CoRR abs/1912.09855 (2019).

[160] Hayes, J. On visible adversarial perturbations & digital watermarking. In CVPR Workshop (2018).

[161] Hayes, J., Melis, L., Danezis, G., and Cristofaro, E. D. Logan: Membership inference attacks against generative models. In PoPETs (Proceedings on Privacy Enhancing Technologies) (2019).

[162] He, Zecheng, Zhang, T., and Lee., R. B. Model inversion attacks against collaborative inference. In ACSAC (2019).

[163] He, C., Xue, M., Wang, J., and Liu., W. Embedding backdoors as the facial features: Invisible backdoor attacks against face recognition systems. In ACM Turing Celebration Conference-China (2020).

[164] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. CoRR abs/1512.03385 (2015).

[165] He, W., Wei, J., Chen, X., Carlini, N., and Song, D. Adversarial example defense: Ensembles of weak defenses are not strong. In 11th USENIX Workshop on Offensive Technologies (2017).

[166] He, X., Lyu, L., Xu, Q., and Sun, L. Model extraction and adversarial transferability, your bert is vulnerable. In arXiv (2021).

[167] Hein, M., and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In NeurIPS (2017).

[168] Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. In NeurIPS (2019).

[169] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In arXiv (2020).

[170] Hesamifard, E., Takabi, H., and Ghasemi, M. Cryptodl: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189 (2017).

[171] Hilprecht, Benjamin, Harterich, M., and Bernau, D. Monte carlo and reconstruction membership inference attacks against generative models. In PoPETs (Proceedings on Privacy Enhancing Technologies) (2019).

[172] Hitaj, Briland, Ateniese, G., and Perez-Cruz, F. Deep models under the gan: information leakage from collaborative deep learning. In ACM SIGSAC (2017).

[173] Hojjat, A., Eisenhofer, T., Schonherr, L., Kolossa, D., Holz, T., Kruegel, C., and Vigna, G. Venomave: Clean-label poisoning against speech recognition. In arXiv (2020).

[174] Hong, S., Carlini, N., and Kurakin, A. Handcrafted backdoors in deep neural networks. In arXiv (2021).

[175] Hong, S., Chandrasekaran, V., Kaya, Y., Dumitras, T., and Papernot, N. On the effectiveness of mitigating data poisoning attacks with gradient shaping. In arXiv (2020).

[176] Hong, S., Davinroy, M., Kaya, Y., Dachman-Soled, D., and Dumitras, T. How to 0wn nas in your spare time. In arXiv (2020).

[177] Hong, S., Davinroy, M., Kaya, Y., Locke, S. N., Rackow, I., Kulda, K., Dachman-Soled, D., and Dumitras, T. Security analysis of deep neural networks operating in the presence of cache side-channel attacks. In arXiv (2018).

[178] Hosseini, H., Kannan, S., and Poovendran, R. Are odds really odd? bypassing statistical detection of adversarial examples. CoRR abs/1907.12138 (2019).

[179] Hosseini, H., and Poovendran, R. Semantic adversarial examples. In CVPR Workshop (2018).

[180] Hosseini, H., Xiao, B., Jaiswal, M., and Poovendran, R. On the limitation of convolutional neural networks in recognizing negative images. In ICMLA (2017).

[181] Hu, J. E., Swaminathan, A., Salman, H., and Yang, G. Improved image wasserstein attacks and defenses. In ICLR Workshop (2020).

[182] Hu, S., Yu, T., Guo, C., Chao, W.-L., and Weinberger, K. Q. A new defense against adversarial images: Turning a weakness into a strength. In NeurIPS (2019).

[183] Hu, W., and Tan, Y. Black-box attacks against rnn based malware detection algorithms. In arXiv (2017).

[184] Hu, W., and Tan, Y. Generating adversarial malware examples for black-box attacks based on gan. In arXiv (2017).

[185] Hu, X., Liang, L., Li, S., Deng, L., Zuo, P., Ji, Y., Xie, X., Ding, Y., Liu, C., Sherwood, T., and Xie, Y. Deepsniffer: A dnn model extraction framework based on learning architectural hints. In International Conference on Architectural Support for Programming Languages and Operating Systems (2020).

[186] Hu, X., Zhao, Y., Deng, L., Liang, L., Zuo, P., Ye, J., Lin, Y., and Xie, Y. Practical attacks on deep neural networks by memory trojaning. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2020).

[187] Huang, B., Wang, Y., and Wang, W. Model-agnostic adversarial detection by random perturbations. In IJCAI (2019).

[188] Huang, R., Xu, B., Schuurmans, D., and Szepesvari, C. Learning with a strong adversary. In arXiv (2015).

[189] Huang, W. R., Geiping, J., Fowl, L., Taylor, G., and Goldstein, T. Metapoison: Practical general-purpose clean-label data poisoning. In NeurIPS (2020).

[190] Huang, X., Alzantot, M., and Srivastava, M. Neuroninspect: Detecting backdoors in neural networks via output explanations. In arXiv (2019).

[191] Huang, X., Kwiatkowska, M., Wang, S., and Wu, M. Safety verification of deep neural networks. In CAV (2017).

[192] Huang, Y., Song, Z., Li, K., and Arora, S. Instahide: Instance-hiding schemes for private distributed learning. In ICML (2020).

[193] Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In ICML (2018).

[194] Ilyas, A., Engstrom, L., and Madry, A. Prior convictions: Black-box adversarial attacks with bandits and priors. In ICLR (2019).

[195] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In NeurIPS (2019).

[196] Jacobsen, J.-H., Behrmann, J., Zemel, R., and Bethge, M. Excessive invariance causes adversarial vulnerability. In ICLR (2019).

[197] Jacoby, Y., Barrett, C., and Katz, G. Verifying recurrent neural networks using invariant inference. In International Symposium on Automated Technology for Verification and Analysis (2020).

[198] Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., and Paperno, N. High accuracy and high fidelity extraction of neural networks. In USENIX (2020).

[199] Jang, Uyeong, Wu, X., and Jha, S. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In Computer Security Applications Conference (2017).

[200] Jang, Y., Zhao, T., Hong, S., and Lee, H. Adversarial defense via learning to generate diverse attacks. In ICCV (2019).

[201] Jayaraman, B., and Evans, D. Evaluating differentially private machine learning in practice. In USENIX (2019).

[202] Ji, Y., Zhang, X., and Wang, T. Backdoor attacks against learning systems. In IEEE Conference on Communications and Network Security (2017).

[203] Jia, Y., Lu, Y., Shen, J., Chen, Q. A., Chen, H., Zhong, Z., and Wei, T. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In ICLR (2020).

[204] Jiang, W., Li, H., Liu, S., Luo, X., and Lu, R. Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles. In IEEE transactions on vehicular technology (2020).

[205] Jin, G., Shen, S., Zhang, D., Dai, F., and Zhang, Y. Ape-gan: Adversarial perturbation elimination with gan. In ICASSP (2019).

[206] Jordan, M., Lewis, J., and Dimakis, A. G. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. In NeurIPS (2019).

[207] Joshi, A., Mukherjee, A., Sarkar, S., and Hegde, C. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In ICCV (2019).

[208] Juuti, M., Szyller, S., Marchal, S., and Asokan, N. Prada: Protecting against dnn model stealing attacks. In EuroS&P (2019).

[209] Kakizaki, K., and Yoshida, K. Adversarial image translation: Unrestricted adversarial examples in face recognition systems. In AAAI Workshop on Artificial Intelligence Safety (2020).

[210] Kanbak, C., Moosavi-Dezfooli, S.-M., and Frossard, P. Geometric robustness of deep networks: analysis and improvement. In CVPR (2018).

[211] Kannan, H., Kurakin, A., and Goodfellow, I. Adversarial logit pairing. In arXiv (2018).

[212] Karmon, D., Zoran, D., and Goldberg, Y. Lavan: Localized and visible adversarial noise. In ICML (2018).

[213] Karra, K., Ashcraft, C., and Fendley, N. The trojai software framework: An open source tool for embedding trojans into deep learning models. In arXiv (2020).

[214] Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In International Conference on Computer Aided Verification (2017).

[215] Katz, G., Huang, D. A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., and Shah, P. The marabou framework for verification and analysis of deep neural networks. In International Conference on Computer Aided Verification (2019).

[216] Kesarwani, M., Mukhoty, B., Arya, V., and Mehta, S. Model extraction warning in mlaas paradigm. In ACSAC (2018).

[217] Khalid, F., Hanif, M. A., Rehman, S., Ahmed, R., and Shafiqu, M. Trisec: Training data-unaware imperceptible security attacks on deep neural networks. In IEEE 25th International Symposium on On-Line Testing and Robust System Design (2019).

[218] Kim, H., Lee, W., and Lee, J. Understanding catastrophic overfitting in single-step adversarial training. In to appear in AAAI (2021).

[219] Ko, C.-Y., Lyu, Z., Weng, T.-W., Daniel, L., Wong, N., and Lin, D. Popqorn: Quantifying robustness of recurrent neural networks. In ICML (2019).

[220] Koh, P. W., and Liang., P. Understanding black-box predictions via influence functions. In ICML (2017).

[221] Kolouri, S., Saha, A., Pirsiavash, H., and Hoffmann, H. Universal litmus patterns: Revealing backdoor attacks in cnns. In CVPR (2020).

[222] Krishna, K., Tomar, G. S., Parikh, A. P., Papernot, N., and Iyyer., M. Thieves on sesamestreet model extraction of bert-based apis. In ICLR (2020).

[223] Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Tech. rep., 2009.

[224] Kumar, R. S. S., Nystrom, M., Lambert, J., Marshall, A., Goertzel, M., Comissoneru, A., Swann, M., and Xia, S. Adversarial machine learning-industry perspectives. In IEEE Security and Privacy Workshops (2020).

[225] Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. In ICLR Workshop (2017).

[226] Kwon, H., Yoon, H., and Park, K.-W. Friendnet backdoor: Indentifying backdoor attack that is safe for friendly deep neural network. In ICSIM (2020).

[227] Laidlaw, C., and Feizi, S. Functional adversarial attacks. In NeurIPS (2019).

[228] Laidlaw, C., Singla, S., and Feizi, S. Perceptual adversarial robustness: Defense against unseen threat models. In ICLR (2021).

[229] Lamb, A., Verma, V., Kannala, J., and Bengio, Y. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In AISec (2019).

[230] Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In IEEE Symposium on Security and Privacy (SP) (2019).

[231] Lee, G.-H., Yuan, Y., Chang, S., and Jaakkola, T. S. Tight certificates of adversarial robustness for randomly smoothed classifiers. In NeurIPS (2020).

[232] Lee, M., and Kolter, Z. On physical adversarial patches for object detection. In arXiv (2019).

[233] Lee, S., Lee, H., and Yoon, S. Adversarial vertex mixup: Toward better adversarially robust generalization. In CVPR (2020).

[234] Lee, T., Edwards, B., Molloy, I., and Su, D. Defending against neural network model stealing attacks using deceptive perturbations. In IEEE Symposium on Security and Privacy Workshops (2019).

[235] Levine, A., and Feizi, S. Deep partition aggregation: Provable defense against general poisoning attacks. CoRR abs/2006.14768 (2020).

[236] Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In NeurIPS (2019).

[237] Li, H., Wang, Y., Xie, X., Liu, Y., Wang, S., Wan, R., Chau, L.-P., and Kot, A. C. Light can hack your face black-box backdoor attack on face recognition systems. In arXiv (2020).

[238] Li, J., Du, T., Ji, S., Zhang, R., Lu, Q., Yang, M., and Wang, T. Textshield: Robust text classification based on multimodal embedding and neural machine translation. In USENIX (2020).

[239] Li, J. B., Qu, S., Li, X., Szurley, J., Kolter, J. Z., and Metze, F. Adversarial music: Real world audio adversary against wake-word detection system. In NeurIPS (2019).

[240] Li, L., Qi, X., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. In arXiv (2020).

[241] Li, P., Yi, J., Zhou, B., and Zhang, L. Improving the robustness of deep neural networks via adversarial training with triplet loss. In IJCAI (2019).

[242] Li, S., Zhao, B. Z. H., Yu, J., Xue, M., Kaafar, D., and Zhu, H. Invisible backdoor attacks against deep neural networks. In IEEE Transactions on Dependable and Secure Computing (2020).

[243] Li, W., Yu, J., Ning, X., Wang, P., Wei, Q., Wang, Y., and Yang, H. Hu-fu: Hardware and software collaborative attack framework against neural networks. In IEEE Computer Society Annual Symposium on VLSI (2018).

[244] Li, Y., Li, L., Wang, L., Zhang, T., and Gong, B. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In ICML (2019).

[245] Li, Y., Li, Y., Lv, Y., Wu, B., Jiang, Y., and Xia, S.-T. Hidden backdoor attack against semantic segmentation models. In ICLR Workshop on Security and Safety in Machine Learning Systems (2021).

[246] Li, Y., Li, Y., Wu, B., Li, L., He, R., and Lyu, S. Backdoor attack with sample-specific triggers. In arXiv (2020).

[247] Li, Y., Tian, D., Chang, M.-C., Bian, X., and Lyu, S. Robust adversarial perturbation on deep proposal-based models. In BMVC (British Machine Vision Conference) (2018).

[248] Li, Y., Zhai, T., Wu, B., Jiang, Y., Li, Z., and Xia, S. Rethinking the trigger of backdoor attack. In arXiv (2020).

[249] Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser. In CVPR (2018).

[250] Liu, A., Wang, J., Liu, X., Cao, B., Zhang, C., and Yu, H. Bias-based universal adversarial patch attack for automatic check-out. In ECCV (2020).

[251] Liu, H., Ji, R., Li, J., Zhang, B., Gao, Y., Wu, Y., and Huang, F. Universal adversarial perturbation via prior driven uncertainty approximation. In ICCV (2019).

[252] Liu, J., Zhang, W., Zhang, Y., Hou, D., Liu, Y., Zha, H., and Yu, N. Detection based defense against adversarial examples from the steganalysis point of view. In CVPR (2019).

[253] Liu, K., Dolan-Gavitt, B., and Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In RAID: International Symposium on Research in Attacks, Intrusions, and Defenses (2018).

[254] Liu, T., Wen, W., and Jin, Y. Sin 2: Stealth infection on neural networka low-cost agile neural trojan attack methodology. In IEEE International Symposium on Hardware Oriented Security and Trust (2018).

[255] Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In ECCV (2018).

[256] Liu, X., Si, S., Zhu, X., Li, Y., and Hsieh, C.-J. A unified framework for data poisoning attack to graph-based semi-supervised learning. In NeurIPS (2019).

[257] Liu, X., Xie, L., Wang, Y., Zou, J., Xiong, J., Ying, Z., and Vasilakos, A. V. Privacy and security issues in deep learning: A survey. In IEEE Access (Journal) (2020).

[258] Liu, X., Yang, H., Liu, Z., Song, L., Li, H., and Chen, Y. Dpatch: An adversarial patch attack on object detectors. In arXiv (2019).

[259] Liu, Y., Chen, X., Liu, C., and Song, D. Delving into transferable adversarial examples and black-box attacks. In ICLR (2017).

[260] Liu, Y., Lee, W.-C., Tao, G., Ma, S., Aafer, Y., and Zhang, X. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In CCS (2019).

[261] Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., and Zhang, X. Trojaning attack on neural networks. In NDSS (2018).

[262] Liu, Y., Ma, X., Bailey, J., and Lu, F. Reflection backdoor: A natural backdoor attack on deep neural networks. In ECCV (2020).

[263] Liu, Y., Wen, R., He, X., Salem, A., Zhang, Z., Backes, M., Cristofaro, E. D., Fritz, M., and Zhang, Y. Ml-doctor: Holistic risk assessment of inference attacks against machine learning models. In arXiv (2021).

[264] Liu, Y., Xie, Y., and Srivastava, A. Neural trojans. In ICCD (2017).

[265] Liu, Y., Yi, Z., and Chen, T. Backdoor attacks and defenses in feature-partitioned collaborative learning. In arXiv (2020).

[266] Long, Y., Bindschaedler, V., Wang, L., Bu, D., Wang, X., Tang, H., Gunter, C. A., and Chen, K. Understanding membership inferences on well-generalized learning models. In arXiv (2018).

[267] Lu, J., Sibai, H., and Fabry, E. Adversarial examples that fool detectors. In arXiv (2017).

[268] Lu, K., Nguyen, C. M., Xu, X., Chari, K., Goh, Y. J., and Foo, C.-S. Armoured: Adversarially robust models using unlabeled data by regularizing diversity. In ICLR (2021).

[269] Lust, J., and Condurache, A. P. Gran: An efficient gradient-norm based detector for adversarial and misclassified examples. In European Symposium on Artificial Neural Networks (2020).

[270] Lyu, Z., Ko, C.-Y., Kong, Z., Wong, N., Lin, D., and Daniel, L. Frown: Thightened neural network robustness certificates. In AAAI (2019).

[271] Ma, C., Chen, L., and Yong, J.-H. Simulating unknown target models for query-efficient black-box attacks. In CVPR (2021).

[272] Ma, S., Liu, Y., Tao, G., Lee, W.-C., and Zhang, X. Nic: Detecting adversarial samples with neural network invariant checking. In NDSS (2019).

[273] Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In ICLR (2018).

[274] Ma, Y., Zhu, X., and Hsu, J. Data poisoning against differentially-private learners: Attacks and defenses. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (2019), IJCAI'19, AAAI Press, p. 4732–4738.

[275] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1, 1 (2007), 3–es.

[276] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In ICLR (2018).

[277] Mahmood, K., Gurevin, D., van Dijk, M., and Nguyen, P. H. Beware the black-box: on the robustness of recent defenses to adversarial examples. In arXiv (2020).

[278] Maini, P., Wong, E., and Kolter, J. Z. Adversarial robustness against the union of multiple perturbation models. In ICML (2020).

[279] Mao, C., Zhong, Z., Yang, J., Vondrick, C., and Ray, B. Metric learning for adversarial robustness. In NeurIPS (2019).

[280] Matyasko, A., and Chau, L.-P. Improved network robustness with adversary critic. In NeurIPS (2019).

[281] Mehra, A., Kailkhura, B., Chen, P.-Y., and Hamm, J. Stealthy poisoning attack on certified robustness. In NeurIPS Workshop (2020).

[282] Melis, L., Song, C., Cristofaro, E. D., and Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In S&P (2019).

[283] Melis, M., Demontis, A., Biggio, B., Brown, G., Fumera, G., and Roli, F. Is deep learning safe for robot vision adversarial examples against the icub humanoid. In ICCV (2017).

[284] Meng, D., and Chen, H. Magnet: A two-pronged defense against adversarial examples. In CCS (2017).

[285] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. In ICLR (2017).

[286] Metzen, J. H., Kumar, M. C., Brox, T., and Fischer, V. Universal adversarial perturbations against semantic image segmentation. In ICCV (2017).

[287] Miller, D. J., Xiang, Z., and Kesidis, G. Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks. In Proceedings of the IEEE (Journal) (2021).

[288] Milli, S., Schmidt, L., Dragan, A. D., and Hardt, M. Model reconstruction from model explanations. In ACM Conference on Fairness, Accountability, and Transparency (2019).

[289] Mireshghallah, F., Taram, M., Vepakomma, P., Singh, A., Raskar, R., and Esmaeilzadeh, H. Privacy in deep learning: A survey. In arXiv (2020).

[290] Mirman, M., Gehr, T., and _x000D_, M. V. Differentiable abstract interpretation for provably robust neural networks. In ICML (2018).

[291] Mirman, M., Hagele, A., Bielik, P., Gehr, T., and Vechev, M. Robustness certification with generative models. In ACM SIGPLAN International Conference on Programming Language Design and Implementation (2021).

[292] Miyato, T., Dai, A. M., and Goodfellow, I. Adversarial training methods for semi-supervised text classification. In ICLR (2017).

[293] Miyato, T., ichi Maeda, S., Koyama, M., Nakae, K., and Ishii, S. Distributional smoothing with virtual adversarial training. In arXiv (2016).

[294] Modas, A., Moosavi-Dezfooli, S.-M., and Frossard, P. Sparsefool: a few pixels make a big difference. In CVPR (2019).

[295] Moon, S., An, G., and Song, H. O. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In ICML (2019).

[296] Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In CVPR (2017).

[297] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In CVPR (2016).

[298] Mopuri, K. R., Ojha, U., Garg, U., and Babu, R. V. Nag: Network for adversary generation. In CVPR (2018).

[299] Mothilal, R. K., Sharma, A., and Tan, C. Explaining machine learning classifiers through diverse counterfactual explanations. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (New York, NY, USA, 2020), FAT* '20, Association for Computing Machinery, p. 607–617.

[300] Mozaffari-Kermani, M., Sur-Kolay, S., Raghunathan, A., and Jha, N. K. Systematic poisoning attacks on and defenses for machine learning in healthcare. In IEEE Journal of Biomedical and Health Informatics (2015).

[301] Muller, C., Serre, F., Singh, G., Puschel, M., and Vechev, M. Scaling polyhedral neural network verification on gpus. In Proceedings of Machine Learning and Systems3 (2021).

[302] Muller, M. N., Makarchuk, G., Singh, G., Puschel, M., and Vechev, M. Prima: Precise and general neural networkcertification via multi-neuron convex relaxations. In arXiv (2021).

[303] Munoz-Gonzales, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., and Roli, F. Towards poisoning of deep learning algorithms with back-gradient optimization. In ACM Workshop on Artificial Intelligence and Security (2017).

[304] Munoz-Gonzalez, L., Pfitzner, B., Russo, M., Carnerero-Cano, J., and Lupu, E. C. Poisoning attacks with generative adversarial nets. In arXiv (2021).

[305] Murakonda, S. K., and Shokri, R. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. arXiv preprint arXiv:2007.09339 (2020).

[306] Mustafa, A., Khan, S. H., Hayat, M., and Shao, J. S. L. Image super-resolution as a defense against adversarial attacks. In IEEE Transactions on Image Processing (Journal) (2020).

[307] Na, T., Ko, J. H., and Mukhopadhyay, S. Cascade adversarial machine learning regularized with a unified embedding. In ICLR (2018).

[308] Narayanan, A., and Shmatikov, V. Robust de-anonymization of large sparse datasets. In 2008 IEEE Symposium on Security and Privacy (sp 2008) (2008), IEEE, pp. 111–125.

[309] Narodytska, N., and Kasiviswanathan, S. P. Simple black-box adversarial attacks on deep neural networks. In CVPR Workshop (2017).

[310] Naseer, M., Khan, S. H., Khan, H., Khan, F. S., and Porikli, F. Cross-domain transferability of adversarial perturbations. In NeurIPS (2019).

[311] Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacksagainst centralized and federated learning. In EuroS&P (2019).

[312] Nelson, B., Rubinstein, B. I. P., Huang, L., Joseph, A. D., Lee, S. J., Rao, S., and Tygar, J. D. Query strategies for evading convex-inducing classifiers. In JMLR (2012).

[313] Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In CVPR (2015).

[314] Nguyen, T. A., and Tran, T. A. Input-aware dynamic backdoor attack. In NeurIPS (2020).

[315] Oh, S. J., Augustin, M., Fritz, M., and Schiele, B. Towards reverse-engineering black-box neural networks. In ICLR (2018).

[316] Okada, R., Ishikura, Z., Shibahara, T., and Hasegawa, S. Special-purpose model extraction attacks: Stealing coarse model with fewer queries. In TrustCom (2020).

[317] Orekondy, T., Schiele, B., and Fritz, M. Knockoff nets: Stealing functionality of black-box models. In CVPR (2019).

[318] Pal, S., Gupta, Y., Shukla, A., Kanade, A., Shevade, S., and Ganapathy, V. A framework for the extraction of deep neural networks by leveraging public data. In arXiv (2019).

[319] Pal, S., Gupta, Y., Shukla, A., Kanade, A., Shevade, S., and Ganapathy, V. Activethief: Model extraction using active learning and unannotated public data. In AAAI (2020).

[320] Palma, A. D., Behl, H., Bunel, R. R., Torr, P., and Kumar, M. P. Scaling the convex barrier with active sets. In ICLR (2020).

[321] Palma, A. D., Behl, H. S., Bunel, R., Torr, P. H., and Kumar, M. P. Scaling the convex barrier with sparse dual algorithms. In arXiv (2021).

[322] Pang, R., Shen, H., Zhang, X., Ji, S., Vorobeychik, Y., Luo, X., Liu, A., and Wang, T. A tale of evil twins: Adversarial inputs versus poisoned models. In CCS (ACM SIGSAC Conference on Computer and Communications Security) (2020).

[323] Pang, R., Zhang, Z., Gao, X., Xi, Z., Ji, S., Cheng, P., and Wang, T. Trojanzoo: Everything you ever wanted to know about neural backdoors (but were afraid to ask). In arXiv (2020).

[324] Pang, T., Du, C., Dong, Y., and Zhu, J. Towards robust detection of adversarial examples. In NeurIPS (2018).

[325] Pang, T., Xu, K., Dong, Y., Du, C., Chen, N., and Zhu, J. Rethinking softmax cross-entropy loss for adversarial robustness. In ICLR (2020).

[326] Pang, T., Xu, K., Du, C., Chen, N., and Zhu, J. Improving adversarial robustness via promoting ensemble diversity. In ICML (2019).

[327] Pang, T., Xu, K., and Zhu, J. Mixup inference: Better exploiting mixup to defend adversarial attacks. In ICLR (2020).

[328] Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. Bag of tricks for adversarial training. In to appear in ICML (2021).

[329] Pang, T., Yang, X., Dong, Y., Xu, K., Zhu, J., and Su, H. Boosting adversarial training with hypersphere embedding. In NeurIPS (2020).

[330] Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. In ICLR (2017).

[331] Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. In arXiv (2016).

[332] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In Asia CCS (2017).

[333] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In EuroS&P (2016).

[334] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In S&P (2016).

[335] Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, U. Scalable private learning with pate. In ICLR (2018).

[336] Paulsen, B., Wang, J., and Wang, C. Reludiff: Differential verification of deep neural networks. In ICSE (2020).

[337] Peri, N., Gupta, N., Huang, W. R., Fowl, L., Zhu, C., Feizi, S., Goldstein, T., and Dickerson, J. P. Deep k-nn defense against clean-label data poisoning attacks. In ECCV (2020).

[338] Perolat, J., Malinowski, M., Piot, B., and Pietquin, O. Playing the game of universal adversarial perturbations. In arXiv (2018).

[339] Phan, N., Wu, X., Hu, H., and Dou, D. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In ICDM (2017).

[340] Phong, L. T., Aono, Y., Hayashi, T., Wang, L., and Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. In IEEE Transactions on Information Forensics and Security (2018).

[341] Picard, S., Chapdelaine, C., Cappi, C., Gardes, L., Jenn, E., Lefevre, B., and Soumarmon, T. Ensuring dataset quality for machine learning certification. In International Workshop on Software Certification (WoSoCer) (2020).

[342] Pintor, M., Demetrio, L., Sotgiu, A., Manca, G., Demontis, A., Carlini, N., Biggio, B., and Roli, F. Indicators of attack failure: Debugging and improving optimization of adversarial examples. In arXiv (2021).

[343] Poretschkin, M., Schmitz, A., Akila, M., Adilova, L., Becker, D., Cremers, A., Hecker, D., Houben, S., Mock, M., Rosenzweig, J., Sicking, J., Schulz, E., Voss, A., and Wrobel, S. Leitfaden zur Gestaltung vertrauenswürdiger Künstlicher Intelligenz - KI-Prüfkatalog. Tech. rep., Fraunhofer-Institut für Intelligente Analyse und Informationssysteme IAIS, Sankt Augustin, 2021.

[344] Poursaeed, O., Katsman, I., Gao, B., and Belongie, S. Generative adversarial perturbations. In CVPR (2018).

[345] Qiao, X., Yang, Y., and Li, H. Defending neural backdoors via generative distribution modeling. In NIPS (2019).

[346] Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial robustness through local linearization. In NeurIPS (2019).

[347] Qin, Y., Carlini, N., Cottrell, G., Goodfellow, I., and Raffel, C. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In ICML (2019).

[348] Qiu, H., Xiao, C., Yang, L., Yan, X., Lee, H., and Li, B. Semanticadv: Generating adversarial examples via attribute-conditional image editing. In ECCV (2020).

[349] Quiring, E., Arp, D., and Rieck, K. Forgotten siblings: Unifying attacks on machine learning and digital watermarking. In EuroS&P (2018).

[350] Raff, E., Sylvester, J., Forsyth, S., and McLean, M. Barrage of random transforms for adversarially robust defense. In CVPR (2019).

[351] Raghunathan, A., Steinhardt, J., and Liang, P. Semidefinite relaxations for certifying robustness to adversarial examples. In NeurIPS (2018).

[352] Rahman, M. A., Rahman, T., Laganiere, R., Mohammed, N., and Wang, Y. Membership inference attack against differentially private deep learning model. In Trans. Data Priv. 11.1 (2018).

[353] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S. B. A., Islam, M. T., Al Maadeed, S., Zughaier, S. M., Khan, M. S., et al. Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images. Computers in biology and medicine 132 (2021), 104319.

[354] Rahmati, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Dai, H. Geoda: a geometric framework for black-box adversarial attacks. In CVPR (2020).

[355] Ramakrishnan, G., and Albarghouthi, A. Backdoors in neural models of source code. In arXiv (2020).

[356] Ranjan, A., Janai, J., Geiger, A., and Black, M. J. Attacking optical flow. In ICCV (2019).

[357] Rao, S., Stutz, D., and Schiele, B. Adversarial training against location-optimized adversarial patches. In ECCV Workshops (2020).

[358] Ren, K., Zheng, T., Qin, Z., and Liu, X. Adversarial attacks and defenses in deep learning. In Engineering (Journal) (2020).

[359] Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In ICML (2020).

[360] Rigaki, Maria, and Garcia, S. A survey of privacy attacks in machine learning. In arXiv (2020).

[361] Roberts, N., Prabhu, V. U., and McAteer, M. Model weight theft with just noise inputs: The curious case of the petulant attacker. In arXiv (2019).

[362] Rolnick, D., and Kording, K. P. Reverse-engineering deep relu networks. In ICML (2020).

[363] Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In CVPR (2019).

[364] Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, J. Z. Certified robustness to label-flipping attacks via randomized smoothing. CoRR abs/2002.03018 (2020).

[365] Roth, K., Kilcher, Y., and Hofmann, T. The odds are odd: A statistical test for detecting adversarial examples. In ICML (2019).

[366] Rozsa, A., Rudd, E. M., and Boult, T. E. Adversarial diversity and hard positive generation. In CVPR Workshop (2016).

[367] Ru, B., Cobb, A. D., Blaas, A., and Gal, Y. Bayesopt adversarial attack. In ICLR (2020).

[368] Ruan, W., Huang, X., and Kwiatkowska, M. Reachability analysis of deep neural networks with provable guarantees. In arXiv (2018).

[369] Ruoss, A., Baader, M., Balunovic, M., and Vechev, M. Efficient certification of spatial robustness. In AAAI (2020).

[370] Ryou, W., Chen, J., Balunovic, M., Singh, G., Dan, A., and Vechev, M. Scalable polyhedral verification of recurrent neural networks. In CAV (2020).

[371] Sabour, S., Cao, Y., Faghri, F., and Fleet, D. J. Adversarial manipulation of deep representations. In ICLR (2016).

[372] Saha, A., Subramanya, A., and Pirsiavash, H. Hidden trigger backdoor attacks. In AAAI (2020).

[373] Sahay, R., Mahfuz, R., and Gamal, A. E. Combatting adversarial attacks through denoising and dimensionality reduction: A cascaded autoencoder approach. In CISS (2019).

[374] Salem, A., Backes, M., and Zhang, Y. Dont trigger me a triggerless backdoor attack against deep neural networks. In arXiv (2020).

[375] Salem, A., Sautter, Y., Backes, M., Humbert, M., and Zhang, Y. Baaan: Backdoor attacks against autoencoder and gan-based machine learning models. In arXiv (2020).

[376] Salem, A., Wen, R., Backes, M., Ma, S., and Zhang, Y. Dynamic backdoor attacks against machine learning models. In arXiv (2020).

[377] Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., and Backes, M. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In NDSS (2019).

[378] Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In ICLR (2018).

[379] Samarati, P., and Sweeney, L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.

[380] Sanyal, A., Kusner, M., Gascon, A., and Kanade, V. Tapas: Tricks to accelerate (encrypted) prediction as a service. In ICML (2018).

[381] Sarkar, E., Benkraouda, H., and Maniatakos, M. Facehack: Triggering backdoored facial recognition systems using facial characteristics. In ACM (2020).

[382] Sarkar, S., Bansal, A., Mahbub, U., and Chellappa, R. Upset and angri: Breaking high performance image classifiers. In arXiv (2017).

[383] Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In NeurIPS (2018).

[384] Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J. P., and Goldstein, T. Just how toxic is data poisoning a unified benchmark for backdoor and data poisoning attacks. In arXiv (2020).

[385] Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J. P., and Goldstein, T. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In Proceedings of the 38th International Conference on Machine Learning (18–24 Jul 2021), M. Meila and T. Zhang, Eds., vol. 139 of Proceedings of Machine Learning Research, PMLR, pp. 9389–9398.

[386] Sen, S., Ravindran, B., and Raghunathan, A. Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks. In ICLR (2020).

[387] Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs targeted clean-label poisoning attacks on neural networks. In NeurIPS (2018).

[388] Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free. In NeurIPS (2019).

[389] Shafahi, A., Najibi, M., Xu, Z., Dickerson, J., Davis, L. S., and Goldstein, T. Universal adversarial training. In AAAI (2020).

[390] Shafahi, A., Saadatpanah, P., Zhu, C., Ghiasi, A., Studer, C., Jacobs, D., and Goldstein, T. Adversarially robust transfer learning. In ICLR (2020).

[391] Shan, S., Wenger, E., Wang, B., Li, B., Zheng, H., and Zhao, B. Y. Gotta catchem all: Using honeypots to catch adversarial attacks on neural networks. In CCS (2020).

[392] Sharad, K., Marson, G. A., Truong, H. T. T., and Karame, G. On the security of randomized defenses against adversarial samples. In Asia CCS (2020).

[393] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In CCS (2016).

[394] Sharma, S., Henderson, J., and Ghosh, J. Certifai: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In AIES (2020).

[395] Shen, J., Zhu, X., and Ma, D. Tensorclog: An imperceptible poisoning attack on deep neural network applications. In IEEE Access (2019).

[396] Shi, Y., and Sagduyu, Y. E. Evasion and causative attacks with adversarial deep learning. In MILCOM (2017).

[397] Shi, Z., Wang, Y., Zhang, H., Yi, J., and Hsieh, C. Fast certified robust training via better initialization and shorter warmup. CoRR abs/2103.17268 (2021).

[398] Shokri, R., and Shmatikov, V. Privacy-preserving deep learning. In CCS (2015).

[399] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In S&P (2017).

[400] Shriver, D., Elbaum, S., and Dwyer, M. B. Dnnv: A framework for deep neural network verification. In arXiv (2021).

[401] Shumailov, I., Zhao, Y., Mullins, R., and Anderson, R. Towards certifiable adversarial sample detection. In AISec (2020).

[402] Silva, S. H., and Najafirad, P. Opportunities and challenges in deep learning adversarial robustness: A survey. In arXiv (2020).

[403] Singh, G., Ganvir, R., Puschel, M., and Vechev, M. Beyond the single neuron convex barrier for neural network certification. In NeurIPS (2019).

[404] Singh, G., Gehr, T., Mirman, M., Puschel, M., and Vechev, M. Fast and effective robustness certification. In NeurIPS (2018).

[405] Singh, G., Gehr, T., Puschel, M., and Vechev, M. An abstract domain for certifying neural networks. In ACM (2019).

[406] Singh, G., Gehr, T., Puschel, M., and Vechev, M. Boosting robustness certification of neural networks. In ICLR (2019).

[407] Singla, S., and Feizi, S. Second-order provable defenses against adversarial attacks. CoRR abs/2006.00731 (2020).

[408] Sitawarin, C., Bhagoji, A. N., Mosenia, A., Chiang, M., and Mittal, P. Darts: Deceiving autonomous cars with toxic signs. In arXiv (2018).

[409] Song, Congzheng, Ristenpart, T., and Shmatikov, V. Machine learning models that remember too much. In ACM SIGSAC (2017).

[410] Song, C., He, K., Lin, J., Wang, L., and Hopcroft, J. E. Robust local features for improving the generalization of adversarial training. In ICLR (2020).

[411] Song, C., He, K., Wang, L., and Hopcroft, J. E. Improving the generalization of adversarial training with domain adaptation. In ICLR (2019).

[412] Song, C., and Shmatikov, V. Auditing data provenance in text-generation models. In ACM SIGKDD (2019).

[413] Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In ICLR (2018).

[414] Song, Y., Shu, R., Kushman, N., and Ermon, S. Constructing unrestricted adversarial examples with generative models. In NeurIPS (2018).

[415] Sperl, P., Kao, C.-Y., Chen, P., Lei, X., and Bottinger, K. Dla: Dense-layer-analysis for adversarial example detection. In EuroS&P (2020).

[416] Stutz, D., Hein, M., and Schiele, B. Disentangling adversarial robustness and generalization. In CVPR (2019).

[417] Stutz, D., Hein, M., and Schiele, B. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In ICML (2020).

[418] Su, D., Zhang, H., Chen, H., Yi, J., Chen, P.-Y., and Gao, Y. Is robustness the cost of accuracy? – a comprehensive study on the robustness of 18 deep image classification models. In Proceedings of the European Conference on Computer Vision (ECCV) (September 2018).

[419] Su, J., Vargas, D. V., and Kouichi, S. One pixel attack for fooling deep neural networks. In IEEE Transactions on Evolutionary Computation (Journal) (2019).

[420] Suciu, O., Marginean, R., Kaya, Y., III, H. D., and Dumitras, T. When does machine learning fail generalized transferability for evasion and poisoning attacks. In USENIX Security Symposium (2018).

[421] Sun, B., Tsai, N.-H., Liu, F., Yu, R., and Su, H. Adversarial defense by stratified convolutional sparse coding. In CVPR (2019).

[422] Sun, X., Khedr, H., and Shoukr, Y. Formal verification of neural network controlled autonomous systems. In ACM International Conference on Hybrid Systems: Computation and Control (2019).

[423] Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. B. Can you really backdoor federated learning. In arXiv (2019).

[424] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In arXiv (2014).

[425] Szyller, S., Duddu, V., Grondahl, T., and Asokan, N. Good artists copy, great artists steal: Model extraction attacks against image translation generative adversarial networks. In arXiv (2021).

[426] Tabacof, P., and Valle, E. Exploring the space of adversarial images. In IJCNN (2016).

[427] Tan, T. J. L., and Shokri, R. Bypassing backdoor detection algorithms in deep learning. In IEEE European Symposium on Security and Privacy (2020).

[428] Tang, R., Du, M., Liu, N., Yang, F., and Hu, X. An embarrassingly simple approach for trojan attack in deep neural networks. In ACM SIGKDD (2020).

[429] Taran, O., Rezaeifar, S., Holotyak, T., and Voloshynovskiy, S. Defending against adversarial attacks by randomized diversification. In CVPR (2019).

[430] Teng, J., Lee, G.-H., and Yuan, Y. l1 adversarial robustness certificates: a randomized smoothing approach. In ICLR (2020).

[431] Theagarajan, R., Chen, M., Bhanu, B., and Zhang, J. Shieldnets: Defending against adversarial attacks using probabilistic adversarial robustness. In CVPR (2019).

[432] Thys, S., Ranst, W. V., and Goedeme, T. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In CVPR Workshop (2019).

[433] Tian, J., Zhou, J., Li, Y., and Duan, J. Detecting adversarial examples from sensitivity inconsistency of spatial-transform domain. In AAAI (2021).

[434] Tian, S., Yang, G., and Cai, Y. Detecting adversarial examples through image transformation. In AAAI (2018).

[435] Tjandraatmadja, C., Anderson, R., Huchette, J., Ma, W., Patel, K., and Vielma, J. P. The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification. In NeurIPS (2020).

[436] Tjeng, V., Xiao, K. Y., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In ICLR (2018).

[437] Tramer, F., and Boneh, D. Adversarial training and robustness for multiple perturbations. In NeurIPS (2019).

[438] Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. In NeurIPS (2020).

[439] Tramer, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: attacks and defenses. In ICLR (2018).

[440] Tramer, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction apis. In USENIX (2016).

[441] Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. The space of transferable adversarial examples, 2017.

[442] Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In NIPS (2018).

[443] Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. Demystifying membership inference attacks in machine learning as a service. In IEEE Transactions on Services Computing (2019).

[444] Truong, L., Jones, C., Hutchinson, B., August, A., Praggastis, B., Jasper, R., Nichols, N., and Tuor, A. Systematic evaluation of backdoor data poisoning attacks on image classifiers. In CVPR (2020).

[445] Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy, 2019.

[446] Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In AAAI (2019).

[447] Turner, A., Tsipras, D., and Madry, A. Clean-label backdoor attacks. In rejected at ICLR, not published (2018).

[448] Udeshi, S., Peng, S., Woo, G., Loh, L., Rawshan, L., and Chattopadhyay, S. Model agnostic defence against backdoor attacks in machine learning. In arXiv (2019).

[449] Uesato, J., Alayrac, J.-B., Huang, P.-S., Stanforth, R., Fawzi, A., and Kohli, P. Are labels required for improving adversarial robustness. In NeurIPS (2019).

[450] Uesato, J., ODonoghue, B., van den Oord, A., and Kohli, P. Adversarial risk and the dangers of evaluating against weak attacks. In ICML (2018).

[451] Vacanti, G., and Looveren, A. V. Adversarial detection and correction by matching prediction distributions. In arXiv (2020).

[452] Veldanda, A. K., Liu, K., Tan, B., Krishnamurthy, P., Khorrami, F., Karri, R., Dolan-Gavitt, B., and Garg, S. Nnoculation: Broad spectrum and targeted treatment of backdoored dnns. In arXiv (2020).

[453] Verma, G., and Swami, A. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. In NeurIPS (2019).

[454] Wang, B., and Gong, N. Z. Stealing hyperparameters in machine learning. In S&P (2018).

[455] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In S&P (2019).

[456] Wang, H., and Yu, C.-N. A direct approach to robust deep learning using adversarial networks. In ICLR (2019).

[457] Wang, J., and Zhang, H. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In ICCV (2019).

[458] Wang, R., Zhang, G., Liu, S., Chen, P.-Y., Xiong, J., and Wang, M. Practical detection of trojan neural networks: Data-limited and data-free cases. In ECCV (2020).

[459] Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. Efficient formal safety analysis of neural networks. In NeurIPS (2018).

[460] Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. Formal security analysis of neural networks using symbolic intervals. In USENIX (2018).

[461] Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., and Kolter, J. Z. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. In arXiv (2021).

[462] Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. On the convergence and robustness of adversarial training. In ICML (2019).

[463] Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In ICLR (2020).

[464] Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., and Qi, H. Beyond inferring class representatives: User-level privacy leakage from federated learning. In IEEE INFOCOM (2019).

[465] Webb, S., Rainforth, T., Teh, Y. W., and Kumar, M. P. Statistical verification of neural networks. In arXiv (2018).

[466] Weber, M., Xu, X., Karlaš, B., Zhang, C., and Li, B. Rab: Provable robustness against backdoor attacks. arXiv preprint arXiv:2003.08904 (2020).

[467] Wei, X., Liang, S., Chen, N., and Cao, X. Transferable adversarial attacks for image and video object detection. In IJCAI (2019).

[468] Weng, T.-W., Chen, P.-Y., Nguyen, L. M., Squillante, M. S., Oseledets, I., and Daniel, L. Proven: Verifying robustness of neural networks with a probabilistic approach. In ICML (2019).

[469] Weng, T.-W., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Boning, D., Dhillon, I. S., and Daniel, L. Towards fast computation of certified robustness for relu networks. In ICML (2018).

[470] Weng, T.-W., Zhang, H., Chen, P.-Y., Yi, J., Su, D., Gao, Y., Hsieh, C.-J., and Daniel, L. Evaluating the robustness of neural networks: An extreme value theory approach. In ICLR (2018).

[471] Wenger, E., Passananti, J., Yao, Y., Zheng, H., and Zhao, B. Y. Backdoor attacks on facial recognition in the physical world. In arXiv (2020).

[472] Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In ICLR (2020).

[473] Wong, E., Schmidt, F. R., and Kolter, J. Z. Wasserstein adversarial examples via projected sinkhorn iterations. In ICML (2019).

[474] Wu, B., Yang, X., Pan, S., and Yuan, X. Model extraction attacks on graph neural networks: Taxonomy and realization. In arXiv (2020).

[475] Wu, K., Wang, A., and Yu, Y. Stronger and faster Wasserstein adversarial attacks. In Proceedings of the 37th International Conference on Machine Learning (13–18 Jul 2020), H. D. III and A. Singh, Eds., vol. 119 of Proceedings of Machine Learning Research, PMLR, pp. 10377–10387.

[476] Wu, M., Wicker, M., Ruan, W., Huang, X., and Kwiatkowska, M. A game-based approximate verification of deep neural networks with provable guarantees. In Theoretical Computer Science (2020).

[477] Wu, T., Tong, L., and Vorobeychik, Y. Defending against physically realizable attacks on image classification. In ICLR (2020).

[478] X, C., C, L., B, L., K, L., and D, S. Targeted backdoor attacks on deep learningsystems using data poisoning. In arXiv (2017).

[479] Xiang, W., Tran, H.-D., and Johnson, T. T. Specification-guided safety verification for feed-forward neural networks. In arXiv (2018).

[480] Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. In IJCAI (2018).

[481] Xiao, C., Zhong, P., and Zheng, C. Enhancing adversarial defense by k-winners-take-all. In ICLR (2020).

[482] Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. Spatially transformed adversarial examples. In ICLR (2018).

[483] Xiao, X., and Tao, Y. M-invariance: towards privacy preserving re-publication of dynamic datasets. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (2007), pp. 689–700.

[484] Xie, C., Huang, K., Chen, P.-Y., and Li, B. Dba: Distributed backdoor attacks against federated learning. In ICLR (2019).

[485] Xie, C., Koyejo, S., and Gupta, I. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In Uncertainty in Artificial Intelligence, PMLR (2020).

[486] Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. In ICLR (2018).

[487] Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. Adversarial examples for semantic segmentation and object detection. In ICCV (2017).

[488] Xie, C., Wu, Y., van der Maaten, L., Yuille, A. L., and He, K. Feature denoising for improving adversarial robustness. In CVPR (2019).

[489] Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., and Yuille, A. Improving transferability of adversarial examples with input diversity. In CVPR (2019).

[490] Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. arXiv preprint (2017).

[491] Xu, C., Wang, J., Tang, Y., Guzman, F., Rubinstein, B. I., and Cohn., T. Targeted poisoning attacks on black-box neural machine translation. In WWW Conference (2021).

[492] Xu, K., Liu, S., Zhao, P., Chen, P.-Y., Zhang, H., Fan, Q., Erdogmus, D., Wang, Y., and Lin, X. Structured adversarial attack: Towards general implementation and better interpretability. In ICLR (2019).

[493] Xu, K., Shi, Z., Zhang, H., Huang, M., Chang, K., Kailkhura, B., Lin, X., and Hsieh, C. Automatic perturbation analysis on general computational graphs. CoRR abs/2002.12920 (2020).

[494] Xu, K., Zhang, G., Liu, S., Fan, Q., Sun, M., Chen, H., Chen, P.-Y., Wang, Y., and Lin, X. Adversarial t-shirt evading person detectors in a physical world. In ECCV (2020).

[495] Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. In NDSS (2018).

[496] Xu, X., Wang, Q., Li, H., Borisov, N., Gunter, C. A., and Li, B. Detecting ai trojans using meta neural analysis. In S&P (2021).

[497] Xue, M., He, C., Wang, J., and Liu, W. One-to-n & n-to-one: Two advanced backdoor attacks against deep learning models. In IEEE Transactions on Dependable and Secure Computing (2020).

[498] Yan, M., Fletcher, C. W., and Torrellas, J. Cache telepathy: Leveraging shared resource attacks to learn dnn architectures. In USENIX (2020).

[499] Yan, Z., Guo, Y., and Zhang, C. Deep defense: Training dnns with improved adversarial robustness. In NeurIPS (2018).

[500] Yang, C., Kortylewski, A., Xie, C., Cao, Y., and Yuille, A. Patchattack: A black-box texture-based attack with reinforcement learning. In ECCV (2020).

[501] Yang, C., Wu, Q., Li, H., and Chen, Y. Generative poisoning attack method against neural networks. In arXiv (2017).

[502] Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In ICML (2020).

[503] Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. I. Ml-loo: Detecting adversarial examples with feature attribution. In AAAI (2020).

[504] Yang, Y., Zhang, G., Katabi, D., and Xu, Z. Me-net: Towards effective adversarial robustness with matrix estimation. In ICML (2019).

[505] Yang, Z., Chang, E.-C., and Liang, Z. Neural network inversion in adversarial setting via background knowledge alignment. In ACM SIGSAC (2019).

[506] Yang, Z., Iyer, N., Reimann, J., and Virani, N. Design of intentional backdoors in sequential models. In arXiv (2019).

[507] Yang, Z., Iyer, N., Reimann, J., and Virani, N. Backdoor attacks in sequential decision-making agents. In AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applica-tins: Potentials, Theories, Practices, Tools, and Risk (2020).

[508] Yang, Z., Li, B., Chen, P.-Y., and Song, D. Characterizing audio adversarial examples using temporal dependency. In ICLR (2019).

[509] Yang, Z., Virani, N., and Iyer, N. S. Countermeasure against backdoor attacks using epistemic classifiers. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II (2020), T. Pham, L. Solomon, and K. Rainey, Eds., vol. 11413, International Society for Optics and Photonics, SPIE, pp. 195 – 202.

[510] Yao, Y., Li, H., Zheng, H., and Zhao, B. Y. Latent backdoor attacks on deep neural networks. In ACM SIGSAC (2019).

[511] YayuanXiong, Xu, F., Zhong, S., and Li, Q. Escaping backdoor attack detection of deep learning. In IFIP International Conference on ICT Systems Security and Privacy Protection (2020).

[512] Yin, D., Kannan, R., and Bartlett, P. Rademacher complexity for adversarially robust generalization. In ICML (2019).

[513] Yin, X., Kolouri, S., and Rohde, G. K. Gat: Generative adversarial training for adversarial example detection and robust classification. In ICLR (2020).

[514] Yu, H., Yang, K., Zhang, T., Tsai, Y.-Y., Ho, T.-Y., and Jin, Y. Cloudleak: Large-scale deep learning models stealing through adversarial examples. In NDSS (2020).

[515] Yuan, X., Ding, L., Zhang, L., Li, X., and Wu, D. Es attack: Model stealing against deep neural networks without data hurdles. In arXiv (2020).

[516] Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial examples: Attacks and defenses for deep learning. In IEEE Transactions on Neural Networks and Learning Systems (Journal) (2019).

[517] Zantedeschi, V., Nicolae, M.-I., and Rawat, A. Efficient defenses against adversarial attacks. In AISec (2017).

[518] Zeng, X., Liu, C., Wang, Y.-S., Qiu, W., Xie, L., Tai, Y.-W., Tang, C. K., and Yuille, A. L. Adversarial attacks beyond the image space. In CVPR (2019).

[519] Zhai, T., Li, Y., Zhang, Z., Wu, B., Jiang, Y., and Xia, S.-T. Backdoor attack against speaker verification. In arXiv (2020).

[520] Zhang, D., Ye, M., Gong, C., Zhu, Z., and Liu, Q. Black-box certification with randomized smoothing: A functional optimization based framework. In NeurIPS (2020).

[521] Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. You only propagate once: Accelerating adversarial training via maximal principle. In NeurIPS (2019).

[522] Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., and Xu, W. Dolphinatack: Inaudible voice commands. In ACM SIGSAC Conference on Computer and Communications Security (2017).

[523] Zhang, H., Chen, H., Song, Z., Boning, D., Dhillon, I. S., and Hsieh, C.-J. The limitations of adversarial training and the blind-spot attack. In ICLR (2019).

[524] Zhang, H., Chen, H., Xiao, C., Li, B., Boning, D. S., and Hsieh, C. Towards stable and efficient training of verifiably robust neural networks. CoRR abs/1906.06316 (2019).

[525] Zhang, H., and Wang, J. Defense against adversarial attacks using feature scattering-based adversarial training. In NeurIPS (2019).

[526] Zhang, H., Weng, T., Chen, P., Hsieh, C., and Daniel, L. Efficient neural network robustness certification with general activation functions. CoRR abs/1811.00866 (2018).

[527] Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In NIPS (2018).

[528] Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In ICML (2019).

[529] Zhang, H., Zhang, P., and Hsieh, C.-J. Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. In AAAI (2019).

[530] Zhang, J., Chen, B., Cheng, X., Binh, H. T. T., and Yu, S. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. In IEEE Internet of Things Journal (2020).

[531] Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In ICML (2020).

[532] Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M. Geometry-aware instance-reweighted adversarial training. In ICLR (2021).

[533] Zhang, R. Making convolutional networks shift-invariant again. In ICML (2019).

[534] Zhang, S., Chen, S., Liu, X., Hua, C., Wang, W., Chen, K., Zhang, J., and Wang, J. Detecting adversarial samples for deep learning models: A comparative study. In IEEE Transactions on Network Science and Engineering (2021).

[535] Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. The secret revealer: Generative model-inversion attacks against deep neural networks. In CVPR (2020).

[536] Zhang, Z., Xiao, G., Li, Y., Lv, T., Qi, F., Wang, Y., Jiang, X., Liu, Z., and Sun, M. Red alarm for pre-trained models: Universal vulnerabilities by neuron-level backdoor attacks. In arXiv (2021).

[537] Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., and Jiang, Y.-G. Clean-label backdoor attacks on video recognition models. In CVPR (2020).

[538] Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., and Chen, K. Seeing isnt believing: Towards more robust adversarial attack against real world object detectors. In ACM SIGSAC (2019).

[539] Zhao, Z., Dua, D., and Singh, S. Generating natural adversarial examples. In ICLR (2018).

[540] Zheng, Tianhang, Chen, C., and Ren, K. Distributionally adversarial attack. In AAAI (2019).

[541] Zheng, H., Zhang, Z., Gu, J., Lee, H., and Prakash, A. Efficient adversarial training with transferable adversarial examples. In CVPR (2020).

[542] Zhong, H., Liao, C., Squicciarini, A. C., Zhu, S., and Miller, D. Backdoor embedding in convolutional neural network models via invisible perturbation. In ACM Conference on Data and Application Security and Privacy (2020).

[543] Zhou, Q., Zhang, R., Wu, B., Li, W., and Mo, T. Detection by attack: Detecting adversarial samples by undercover attack. In ESORICS (2020).

[544] Zhou, Y., Kantarcioglu, M., and Xi, B. Breaking transferability of adversarial samples with randomness. In arXiv (2018).

[545] Zhu, C., Huang, W. R., Li, H., Taylor, G., Studer, C., and Goldstein, T. Transferable clean-label poisoning attacks on deep neural nets. In ICML (2019).

[546] Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. In NeurIPS (2019).

[547] Zhu, L., Ning, R., Wang, C., Xin, C., and Wu, H. Gangsweep: Sweep out neural backdoors by gan. In ACM MM (2020).

[548] Zhu, Y., Cheng, Y., Zhou, H., and Lu, Y. Hermes attack: Steal dnn models with lossless inference accuracy. In USENIX (2021).

[549] Zhu, Z.-A., Lu, Y.-Z., and Chiang, C.-K. Generating adversarial examples by makeup attacks on face recognition. In 2019 IEEE International Conference on Image Processing (ICIP) (2019).

[550] Zolna, K., Zajac, M., Rostamzadeh, N., and Pinheiro, P. O. Adversarial framing for image and video classification. In AAAI (2019).

Appendix A

Papers by category

A.1 Attacks on Deep Learning Systems

Evaluation Metrics

2018: Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach,

Backdoor

2020: TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask),

Semantic Attack

2016: Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition, Measuring the effect of nuisance variables on classifiers, **2017:** Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN, DolphinAtack: Inaudible Voice Commands, Black-Box Attacks against RNN based Malware Detection Algorithms, Adversarial Patch, On the Limitation of Convolutional Neural Networks in Recognizing Negative Images, **2018:** HotFlip: White-Box Adversarial Examples for Text Classification, Unravelling robustness of deep learning based face recognition against adversarial attacks, Generating Natural Adversarial Examples, Spatially Transformed Adversarial Examples, Constructing Unrestricted Adversarial Examples with Generative Models, Generative Adversarial Perturbations, Semantic Adversarial Examples, DARTS: Deceiving Autonomous Cars with Toxic Signs, Generating Adversarial Examples with Adversarial Networks, **2019:** Adversarial Defense via Learning to Generate Diverse Attacks, Functional Adversarial Attacks, The Limitations of Adversarial Training and the Blind-Spot Attack, Excessive Invariance Causes Adversarial Vulnerability, Generating Adversarial Examples By Makeup Attacks on Face Recognition, Adversarial Music: Real World Audio Adversary Against Wake-word Detection System, Structured Adversarial Attack: Towards General Implementation and Better Interpretability, Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers, Exploring the Landscape of Spatial Robustness, **2020:** SemanticAdv: Generating Adversarial Examples via Attribute-conditional Image Editing, Adversarial Policies: Attacking Deep Reinforcement Learning, Natural Adversarial Examples, PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning, Unrestricted Adversarial Examples via Semantic Manipulation, Adversarial Image Translation: Unrestricted Adversarial Examples in Face Recognition Systems, **2021:** Perceptual Adversarial Robustness: Defense Against Unseen Threat Models,

Perturbation-based

2018: Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models,

Loss Queries

2020: Sign Bits Are All You Need for Black-Box Attacks,

Targeted Attacks

2017: Generative poisoning attack method against neural networks, **2019:** Analyzing federated learning through an adversarial lens, Can you really backdoor federated learning, **2020:** The Limitations of Federated Learning in Sybil Settings,

Imperceptible

2019: Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, Sparse and Imperceivable Adversarial Attacks,

Meta Paper

2018: Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach, **2020:** On Adaptive Attacks to Adversarial Example Defenses,

Universal Perturbation

2017: Adversarial Patch, Universal Adversarial Perturbations, Universal adversarial perturbations against semantic image segmentation, UPSET and ANGRI: Breaking High Performance Image Classifiers, **2018:** Generative Adversarial Perturbations, NAG: Network for Adversary Generation, **2019:** Adversarial Framing for Image and Video Classification, Adversarial camera stickers: A physical camera-based attack on deep learning systems, Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation, **2020:** Sparse-RS: a versatile framework for query-efficient sparse black-box adversarial attacks,

Evasion Attack

2020: Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles, **2021:** Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent,

Natural

2015: Manitest: Are classifiers really invariant, **2018:** Unravelling robustness of deep learning based face recognition against adversarial attacks, Geometric robustness of deep networks: analysis and improvement, Generating Natural Adversarial Examples, **2019:** Exploring the Landscape of Spatial Robustness, **2020:** Natural Adversarial Examples,

Perceptible

2017: On the Limitation of Convolutional Neural Networks in Recognizing Negative Images, **2018:** Physical Adversarial Examples for Object Detectors, **2019:** Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, Sparse and Imperceivable Adversarial Attacks,

Defense Evaluation

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, On Adaptive Attacks to Adversarial Example Defenses,

Data-independent

2019: Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation,

Surrogate Model

2017: Black-Box Attacks against RNN based Malware Detection Algorithms, **2019:** Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks,

White Box

2019: Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks, Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, **2020:** Minimally distorted adversarial examples with a fast adaptive boundary attack,

Ensemble of Attacks

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,

Robustness

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,

Transferability

2016: Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, **2017:** Universal Adversarial Perturbations, Delving into Transferable Adversarial Examples and Black-box Attacks, **2018:** Boosting adversarial attacks with momentum, **2019:** Cross-Domain Transferability of Adversarial Perturbations, Evading defenses to transferable adversarial examples by translation-invariant attacks, Improving transferability of adversarial examples with input diversity,

Patch-based

2016: Measuring the effect of nuisance variables on classifiers, Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition, **2017:** Adversarial Patch, **2018:** Physical Adversarial Examples for Object Detectors, On Visible Adversarial Perturbations & Digital Watermarking, LaVAN: Localized and Visible Adversarial Noise, Robust Physical-World Attacks on Deep Learning Visual Classification, **2019:** Adversarial camera stickers: A physical camera-based attack on deep learning systems, Attacking Optical Flow, One Pixel Attack for Fooling Deep Neural Networks, Adversarial Framing for Image and Video Classification, DPatch: An Adversarial Patch Attack on Object Detectors, On Physical Adversarial Patches for Object Detection, Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection, **2020:** Adversarial T-shirt Evading Person Detectors in a Physical World, Patch-wise Attack for Fooling Deep Neural Network, Fooling detection alone is not enough: Adversarial attack against multiple object tracking, Square attack: a query-efficient black-box adversarial attack via random search, Bias-based Universal Adversarial Patch Attack for Automatic Check-out, PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning, Sparse-RS: a versatile framework for query-efficient sparse black-box adversarial attacks, Adversarial Training against Location-Optimized Adversarial Patches,

Generator-based

2018: Generative Adversarial Perturbations,

Semantic Attack with GAN

2019: Adversarial Defense via Learning to Generate Diverse Attacks,

Parameter Free

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,

Unrestricted Perturbation

2020: Sparse-RS: a versatile framework for query-efficient sparse black-box adversarial attacks,

Targeted Poisoning

2015: Systematic poisoning attacks on and defenses for machine learning in healthcare, **2017:** Understanding black-box predictions via influence functions, **2018:** When does machine learning FAIL generalized transferability for evasion and poisoning attacks, **2019:** DBA: Distributed backdoor attacks against federated learning, **2020:** Local model poisoning attacks to Byzantine-robust federated learning, Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation, How to backdoor federated learning, Backdoor attacks on federated meta-learning, Backdoor attacks and defenses in feature-partitioned collaborative learning,

Federated Learning

2019: DBA: Distributed backdoor attacks against federated learning, Analyzing federated learning through an adversarial lens, Can you really backdoor federated learning, **2020:** Local model poisoning attacks to Byzantine-robust federated learning, Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation, How to backdoor federated learning, Backdoor attacks on federated meta-learning, PoisonGAN: Generative Poisoning Attacks against Federated Learning in Edge Computing Systems, The Limitations of Federated Learning in Sybil Settings,

Alternative Optimization Problem

2019: Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach, Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization,

Defense against Patch-based

2018: On Visible Adversarial Perturbations & Digital Watermarking,

Black Box

2012: Query strategies for evading convex-inducing classifiers, **2016:** Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, **2017:** Adversarial machine learning at scale, On the Limitation of Convolutional Neural Networks in Recognizing Negative Images, UPSET and ANGRI: Breaking High Performance Image Classifiers, Black-Box

Attacks against RNN based Malware Detection Algorithms, Simple Black-Box Adversarial Attacks on Deep Neural Networks, Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN, Delving into Transferable Adversarial Examples and Black-box Attacks, **2018:** Boosting adversarial attacks with momentum, Generating Natural Adversarial Examples, Black-box Adversarial Attacks with Limited Queries and Information, Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms, Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models, **2019:** Sparse and Imperceivable Adversarial Attacks, NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks, Evading defenses to transferable adversarial examples by translation-invariant attacks, Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization, Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach, Cross-Domain Transferability of Adversarial Perturbations, AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks, Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors, Efficient decision-based black-box adversarial attacks on face recognition, Improving transferability of adversarial examples with input diversity, DPatch: An Adversarial Patch Attack on Object Detectors, Simple Black-box Adversarial Attacks, **2020:** GeoDA: a geometric framework for black-box adversarial attacks, HopSkipJumpAttack: A Query-Efficient Decision-Based Attack, PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning, Square attack: a query-efficient black-box adversarial attack via random search, Bayesopt adversarial attack, Sparse-RS: a versatile framework for query-efficient sparse black-box adversarial attacks, SemanticAdv: Generating Adversarial Examples via Attribute-conditional Image Editing, Sign Bits Are All You Need for Black-Box Attacks,

Gradient free

2019: Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks,

Epsilon Perturbation

2015: Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, Manitest: Are classifiers really invariant, **2016:** Exploring the space of adversarial images, The Limitations of Deep Learning in Adversarial Settings, Adversarial diversity and hard positive generation, DeepFool: a simple and accurate method to fool deep neural networks, Adversarial Manipulation of Deep Representations, **2017:** Adversarial machine learning at scale, Adversarial examples that fool detectors, UPSET and ANGRI: Breaking High Performance Image Classifiers, Simple Black-Box Adversarial Attacks on Deep Neural Networks, Universal adversarial perturbations against semantic image segmentation, Objective metrics and gradient descent algorithms for adversarial examples in machine learning, Adversarial Transformation Networks: Learning to Generate Adversarial Examples, Adversarial examples for semantic segmentation and object detection, Houdini: Fooling deep structured visual and speech recognition models with adversarial examples, Is Deep Learning Safe for Robot Vision Adversarial Examples against the iCub Humanoid, Towards Evaluating the Robustness of Neural Networks, Adversarial examples in the physical world, ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models, **2018:** Black-box Adversarial Attacks with Limited Queries and Information, Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms, EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples, Synthesizing Robust Adversarial Examples, Geometric robustness of deep networks: analysis and improvement, Robust Adversarial Perturbation on Deep Proposal-based Models, Adversarial attacks on face detectors using neural net based constrained optimization, Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning, Generative Adversarial Perturbations, Audio Adversarial Examples: Targeted Attacks on Speech-to-Text, Boosting adversarial attacks with momentum, Clean-label backdoor attacks, Adversarial Risk and the Dangers of Evaluating Against Weak Attacks, Provably Minimally-Distorted Adversarial Examples (previous name: Ground-truth adversarial examples), **2019:** Adversarial attacks beyond the image space, Structured Adversarial Attack: Towards General Implementation and Better Interpretability, Decoupling direction and norm for efficient gradient-based l_2 adversarial attacks and defenses, Wasserstein Adversarial Examples via Projected Sinkhorn Iterations, NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks, TrISec: Training data-unaware imperceptible security attacks on deep neural networks, Sparse and Imperceivable Adversarial Attacks, ADef: An Iterative Algorithm to Construct Adversarial Deformations, Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks, Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization, Sparsefool: a few pixels make a big difference, Accurate, reliable and fast robustness evaluation, Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach, Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks, Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors, Evading defenses to transferable adversarial examples by translation-invariant attacks, Improving transferability of adversarial examples with input diversity, Transferable Adversarial Attacks for Image and Video Object Detection, Distributionally adversarial attack, Efficient decision-based black-box adversarial attacks on face recognition, Adversarial Training and Robustness for Multiple Perturbations, **2020:** Bypassing backdoor detection algorithms in deep learning, A tale of evil twins: Adversarial inputs versus poisoned models, Patch-wise Attack for Fooling Deep Neural Network, Minimally distorted adversarial examples with a fast adaptive boundary attack, The Role of Sign and Direction of Gradient on the Performance of CNN, HopSkipJumpAttack: A Query-Efficient Decision-Based Attack, Improved Image Wasserstein Attacks and Defenses, Bayesopt adversarial attack, GeoDA: a geometric framework for black-box adversarial attacks,

Ensemble

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,

Adaptive Attacks

2020: On Adaptive Attacks to Adversarial Example Defenses,

General Finding

2021: Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent,

Certificate Spoofing

2020: Breaking Certified Defenses: Semantic Adversarial Examples with Spoofed Robustness Certificates,

Semi-supervised Learning

2016: Distributional Smoothing with Virtual Adversarial Training,

Score-based

2019: Simple Black-box Adversarial Attacks,

Gradient Estimation

2017: Simple Black-Box Adversarial Attacks on Deep Neural Networks, **2018:** Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms, **2019:** AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks, **2020:** HopSkipJumpAttack: A Query-Efficient Decision-Based Attack,

Real-world Attack

2016: Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition, **2017:** Adversarial examples in the physical world, Adversarial Patch, **2018:** Robust Physical-World Attacks on Deep Learning Visual Classification, DARTS: Deceiving Autonomous Cars with Toxic Signs, Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector, Physical Adversarial Examples for Object Detectors, **2019:** Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection, On Physical Adversarial Patches for Object Detection, Seeing isnt believing: Towards more robust adversarial attack against real world object detectors, Functional Adversarial Attacks, Adversarial camera stickers: A physical camera-based attack on deep learning systems, Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, **2020:** Adversarial T-shirt Evading Person Detectors in a Physical World, Bias-based Universal Adversarial Patch Attack for Automatic Check-out,

Gradient-based Attack

2016: Distributional Smoothing with Virtual Adversarial Training,

Multiple Threat Models

2021: Perceptual Adversarial Robustness: Defense Against Unseen Threat Models,

Untargeted Attacks

2017: Generative poisoning attack method against neural networks,

Adversarial Regularization

2016: Distributional Smoothing with Virtual Adversarial Training, **2019:** Improved Network Robustness with Adversary Critic,

Survey

2018: Wild patterns: Ten years after the rise of adversarial machine learning, Motivating the rules of the game for adversarial example research, **2020:** Universal Adversarial Perturbations: A Survey, Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey, Adversarial Attacks and Defenses in Deep Learning, Privacy in deep learning: A survey, Adversarial machine learning-industry perspectives,

Regularization

2016: Distributional Smoothing with Virtual Adversarial Training, **2019:** Improved Network Robustness with Adversary Critic,

Poisoning Attacks

2015: Systematic poisoning attacks on and defenses for machine learning in healthcare, **2017:** Backdoor attacks against learning systems, Understanding black-box predictions via influence functions, Generative poisoning attack method against neural networks, Evasion and causative attacks with adversarial deep learning, **2018:** Clean-label backdoor attacks, SIN 2: Stealth infection on neural networka low-cost agile neural trojan attack methodology, When does machine learning FAIL generalized transferability for evasion and poisoning attacks, Hardware trojan attacks on neural networks, Hu-Fu: Hardware and software collaborative attack framework against neural networks, Turning your weakness into a strength: Watermarking deep neural networks by backdooring, **2019:** Trojan attacks on wireless signal classification with adversarial machine learning, A backdoor attack against LSTM-based text classification systems, Luminance-based video backdoor attack against anti-spoofing rebroadcast detection, Can you really backdoor federated learning, Analyzing federated learning through an adversarial lens, TensorClog: An imperceptible poisoning attack on deep neural network applications, A little is enough: Circumventing defenses for distributed learning, A new backdoor attack in CNNs by training set corruption without label poisoning, DBA: Distributed backdoor attacks against federated learning, **2020:** Backdoor attacks and defenses in feature-partitioned collaborative learning, Can adversarial weight perturbations inject neural backdoors, Bypassing backdoor detection algorithms in deep learning, A tale of evil twins: Adversarial inputs versus poisoned models, Escaping Backdoor Attack Detection of Deep Learning, Local model poisoning attacks to Byzantine-robust federated learning, Input-aware dynamic backdoor attack, Blind backdoors in deep learning models, Backdoors in Neural Models of Source Code, Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation, Backdoor Attacks in Sequential Decision-Making Agents, Friendnet backdoor: Indentifying backdoor attack that is safe for friendly deep neural network, Backdoor embedding in convolutional neural network models via invisible perturbation, Reflection backdoor: A natural backdoor attack on deep neural networks, Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles, TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask), Practical attacks on deep neural networks by memory trojaning, How to backdoor federated learning, Stealthy Poisoning Attack on Certified Robustness, The Limitations of Federated Learning in Sybil Settings, Backdoor attacks on federated meta-learning, PoisonGAN: Generative Poisoning Attacks against Federated Learning in Edge Computing Systems, Live Trojan attacks on deep neural networks, **2021:** Handcrafted Backdoors in Deep Neural Networks, Deep Feature Space Trojan Attack of Neural Networks by Controlled Detoxification,

Adversarial Frame

2020: Sparse-RS: a versatile framework for query-efficient sparse black-box adversarial attacks,

Unsupervised

2019: Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation,

Fooling Pattern

2015: Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images,

GAN-based

2017: Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN, **2018:** Generating Adversarial Examples with Adversarial Networks, Generating Natural Adversarial Examples, NAG: Network for Adversary Generation, **2019:** Improved Network Robustness with Adversary Critic,

Distribution-based

2019: NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks,

Adversarial Training

2016: Distributional Smoothing with Virtual Adversarial Training, **2017:** Adversarial machine learning at scale, **2018:** Geometric robustness of deep networks: analysis and improvement, **2019:** Adversarial Training and Robustness for Multiple Perturbations, Adversarial Defense via Learning to Generate Diverse Attacks, Wasserstein Adversarial Examples via Projected Sinkhorn Iterations, Sparse and Imperceivable Adversarial Attacks, Improved Network Robustness with Adversary Critic, **2020:** Improved Image Wasserstein Attacks and Defenses, Adversarial Training against Location-Optimized Adversarial Patches, **2021:** Perceptual Adversarial Robustness: Defense Against Unseen Threat Models,

Defenses

2020: TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask),

Black Box Attack

2017: Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN,

Untargeted Poisoning

2017: Understanding black-box predictions via influence functions, **2020:** Local model poisoning attacks to Byzantine-robust federated learning, Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation,

Evasion Attacks

2020: Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles,

Sampling-based

2019: Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks,

Black Box

2019: Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks, **2020:** Bias-based Universal Adversarial Patch Attack for Automatic Check-out,

Decision-based

2020: GeoDA: a geometric framework for black-box adversarial attacks,

A.2 Certification and Verification Methods

Hybrid

2019: Boosting Robustness Certification of neural networks, **2020:** Safety Verification and Robustness Analysis of Neural Networks via Quadratic Constraints and Semidefinite Programming, Towards safety verification of direct perception neural networks, **2021:** Fast Geometric Projections for Local Robustness Certification, Beta-CROWN: Efficient Bound Propagation with Per-neuron Split Constraints for Complete and Incomplete Neural Network Verification,

Deterministic

2020: CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models,

Semidefinite Programming

2017: Safety Verification of Deep Neural Networks, **2018:** Semidefinite relaxations for certifying robustness to adversarial examples, **2020:** Efficient Neural Network Verification with Exactness Characterization, Adversarial robustness via robust low rank representations, Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming,

Linear Relaxation

2018: Fast and Effective Robustness Certification, Specification-guided safety verification for feedforward neural networks, Differentiable Abstract Interpretation for Provably Robust Neural Networks, Efficient Neural Network Robustness Certification with General Activation Functions, Towards fast computation of certified robustness for relu networks, A Dual Approach to Scalable Verification of Deep Networks, Formal Security Analysis of Neural Networks using Symbolic Intervals, Fast and Effective Robustness Certification, **2019:** CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks, RecurJac: An Efficient Recursive Algorithm for Bounding Jacobian Matrix of Neural Networks and Its Applications, Verification of RNN-Based Neural Agent-Environment Systems, Optimization and abstraction: a synergistic approach for analyzing neural network robustness, POPQORN: Quantifying Robustness of Recurrent Neural Networks, An Abstract Domain for Certifying Neural Networks, FROWN: Thightened Neural Network Robustness Certificates, Beyond the Single Neuron Convex Barrier for Neural Network Certification, **2020:** Scaling the Convex Barrier with Active Sets, Lagrangian decomposition for neural network verification, Scalable Polyhedral Verification of Recurrent Neural Networks, Verifying Recurrent Neural Networks using Invariant Inference, The Convex Relaxation Barrier, Revisited: Tightened Single-Neuron Relaxations for

Neural Network Verification, Black-box Certification and Learning under Adversarial Perturbations, ReluDiff: Differential Verification of Deep Neural Networks, Efficient Certification of Spatial Robustness, An Abstraction-Based Framework for Neural Network Verification, **2021:** DeepSplit: Scalable Verification of Deep Neural Networks via Operator Splitting, Fast and Precise Certification of Transformers, PRIMA: Precise and General Neural NetworkCertification via Multi-Neuron Convex Relaxations, Scaling Polyhedral Neural Network Verification on GPUs, Robustness Certification with Generative Models, DNNV: A Framework for Deep Neural Network Verification, Scaling the Convex Barrier with Sparse Dual Algorithms,

Survey

2020: Ensuring Dataset Quality for Machine Learning Certification, SoK: Certified Robustness for Deep Neural Networks, **2021:** Recent Advances in Understanding Adversarial Robustness of Deep Neural Networks,

Complete

2017: Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks, Maximum Resilience of Artificial Neural Networks, Reluplex: An efficient SMT solver for verifying deep neural networks, **2018:** Evaluating Robustness of Neural Networks with Mixed Integer Programming, A Unified View of Piecewise Linear Neural Network Verification, Efficient Formal Safety Analysis of Neural Networks, Reachability analysis of deep neural networks with provable guarantees, **2019:** Provable Certificates for Adversarial Examples: Fitting a Ball in the Union of Polytopes, Branch and Bound for Piecewise Linear Neural Network Verification, Formal verification of neural network controlled autonomous systems, The Marabou Framework for Verification and Analysis of Deep Neural Networks, **2020:** Efficient Verification of ReLU-based Neural Networks via Dependency Analysis, An SMT-Based Approach for Verifying Binarized Neural Networks,

Probabilistic Verification

2018: Statistical Verification of Neural Networks, **2021:** Verifying probabilistic specifications with functional lagrangians,

Lipschitz

2017: Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation, **2018:** Towards fast computation of certified robustness for relu networks, **2020:** Certifying Geometric Robustness of Neural Networks, A game-based approximate verification of deep neural networks with provable guarantees,

Linear Inequality Propagation

2020: Provable Robustness of ReLU networks via Maximization of Linear Regions,

Probabilistic

2018: Statistical Verification of Neural Networks, **2019:** PROVEN: Verifying Robustness of Neural Networks with a Probabilistic Approach, Certified Adversarial Robustness with Additive Noise, Certified Robustness to Adversarial Examples with Differential Privacy, **2020:** Black-Box Certification with Randomized Smoothing: A Functional Optimization Based Framework, A Framework for Robustness Certification of Smoothed Classifiers using f-Divergences, Tight Certificates of Adversarial Robustness for Randomly Smoothed Classifiers, l1 Adversarial Robustness Certificates: a Randomized Smoothing Approach, Randomized Smoothing of All Shapes and Sizes, **2021:** Verifying probabilistic specifications with functional lagrangians,

A.3 Defense Methods

Re-training

2017: Neural Trojans,

Differential Privacy (Objective Perturbation)

2017: Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning,

Generator Approach

2019: A Direct Approach to Robust Deep Learning Using Adversarial Networks,

Model Modifications - Ensemble Methods

2017: Robustness to adversarial examples through an ensemble of specialists,

Defenses against Poisoning/Backdoor Attacks (Detection)

2018: Spectral Signatures in Backdoor Attacks, **2019:** Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation, TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems, DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks, STRIP: a defence against trojan attacks on deep neural networks, NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations, Model Agnostic Defence against Backdoor Attacks in Machine Learning, **2020:** GangSweep: Sweep out Neural Backdoors by GAN, Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases, Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs, Deep k-NN Defense Against Clean-Label Data Poisoning Attacks, **2021:** Detecting AI Trojans Using Meta Neural Analysis,

Backdoor

2017: Neural Trojans, **2018:** Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, Spectral Signatures in Backdoor Attacks, **2019:** STRIP: a defence against trojan attacks on deep neural networks, DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks, Defending Neural Backdoors via Generative Distribution Modeling, Model Agnostic Defence against Backdoor Attacks in Machine Learning, TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems, ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations, **2020:** NNoculation: Broad Spectrum and Targeted Treatment of Backdoored DNNs, Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems, Deep k-NN Defense Against Clean-Label Data Poisoning Attacks, Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases, TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask), On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping, Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs, GangSweep: Sweep out Neural Backdoors by GAN, **2021:** What Doesnt Kill You Makes You Robust(er): Adversarial Training against Poisons and Backdoors, Strong Data Augmentation Sanitizes Poisoning and Backdoor Attacks Without an Accuracy Tradeoff,

Privacy Preserving Defense

2017: Membership Inference Attacks Against Machine Learning Models, **2020:** InstaHide: Instance-hiding Schemes for Private Distributed Learning, **2021:** Is Private Learning Possible with Instance Encoding,

Model Extraction

2016: Stealing Machine Learning Models via Prediction APIs, **2017:** Practical Black-Box Attacks against Machine Learning, **2018:** Model extraction warning in MLaaS paradigm, Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking, **2019:** PRADA: Protecting Against DNN Model Stealing Attacks, Defending Against Neural Network Model Stealing Attacks Using Deceptive Perturbations, **2020:** Special-purpose Model Extraction Attacks: Stealing Coarse Model with Fewer Queries, Exploring Connections Between Active Learning and Model Extraction, **2021:** Model Extraction and Adversarial Transferability, Your BERT is Vulnerable,

Semantic Attack

2018: HotFlip: White-Box Adversarial Examples for Text Classification, **2019:** Adversarial Defense via Learning to Generate Diverse Attacks, The Limitations of Adversarial Training and the Blind-Spot Attack, **2021:** Perceptual Adversarial Robustness: Defense Against Unseen Threat Models,

External Network Add-on (GAN-based)

2018: PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples, Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models, **2019:** APE-GAN: Adversarial Perturbation Elimination with GAN,

Train and Test Data Modification

2018: Thwarting adversarial examples: An l0-robust sparse Fourier transform, Countering adversarial images using input transformations, **2019:** ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation, **2020:** Mixup Inference: Better Exploiting Mixup to Defend Adversarial Attacks,

Training Data Modification

2017: Efficient Defenses Against Adversarial Attacks,

Epsilon Perturbation Attacks

2018: Thermometer Encoding: One Hot Way To Resist Adversarial Examples,

Imperceptible

2019: Sparse and Imperceivable Adversarial Attacks,

Model Modification

2016: Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks, **2017:** Robustness to adversarial examples through an ensemble of specialists, **2018:** Towards Robust Neural Networks via Random Self-ensemble, Stochastic Activation Pruning for Robust Adversarial Defense, Breaking Transferability of Adversarial Samples with Randomness, **2019:** Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks, Improving adversarial robustness via promoting ensemble diversity, Defending Against Adversarial Attacks by Randomized Diversification, Making Convolutional Networks Shift-Invariant Again, **2020:** Rethinking softmax cross-entropy loss for adversarial robustness, Enhancing Adversarial Defense by k-Winners-Take-All, EMPIR: Ensembles of Mixed Precision Deep Networks for Increased Robustness Against Adversarial Attacks,

Meta Paper

2018: Adversarially Robust Generalization Requires More Data, **2020:** Overfitting in adversarially robust deep learning, On Adaptive Attacks to Adversarial Example Defenses,

Train and Test Data Modification

2018: Thwarting adversarial examples: An l0-robust sparse Fourier transform, Countering adversarial images using input transformations, **2019:** ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation, Adversarial Defense by Stratified Convolutional Sparse Coding, **2020:** Mixup Inference: Better Exploiting Mixup to Defend Adversarial Attacks,

Universal Perturbation

2020: Universal Adversarial Training,

Evasion Attack

2021: Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent,

Natural

2018: Geometric robustness of deep networks: analysis and improvement,

Perceptible

2019: Sparse and Imperceivable Adversarial Attacks,

defense evaluation

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, On Adaptive Attacks to Adversarial Example Defenses,

Train and Test Data Modification (Random Input Transformation)

2018: Countering adversarial images using input transformations,

Robust Generalization

2019: Rademacher Complexity for Adversarially Robust Generalization, Disentangling Adversarial Robustness and Generalization, **2020:** Adversarial Vertex Mixup: Toward Better Adversarially Robust Generalization, Robust Local Features for Improving the Generalization of Adversarial Training, Overfitting in adversarially robust deep learning, **2021:** Bag of Tricks for Adversarial Training,

Test Data Modification (Random Input Transformation)

2017: Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification, **2018:** Mitigating Adversarial Effects Through Randomization, Countering adversarial images using input transformations, **2020:** On the Security of Randomized Defenses Against Adversarial Samples,

Ensemble of Attacks

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,

Adaptive Epsilon

2020: MMA Training: Direct Input Space Margin Maximization through Adversarial Training,

Robustness

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, **2021:** Indicators of Attack Failure: Debugging and Improving Optimization of Adversarial Examples,

Defenses against Poisoning/Backdoor Attacks (Pre-processing)

2020: Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems,

Model Inversion

2019: Deep leakage from gradients,

Data Stealing

2017: Practical Black-Box Attacks against Machine Learning, **2019:** Deep leakage from gradients,

Adversarial Training Against Patch-baseds

2020: Defending Against Physically Realizable Attacks on Image Classification,

Basic Privacy Preserving Defenses

2017: Membership Inference Attacks Against Machine Learning Models,

Patch-based

2018: On Visible Adversarial Perturbations & Digital Watermarking, **2020:** Defending Against Physically Realizable Attacks on Image Classification, Adversarial Training against Location-Optimized Adversarial Patches,

Differential Privacy (Gradient Perturbation)

2016: Deep Learning with Differential Privacy,

Semantic Attack with GAN

2019: Adversarial Defense via Learning to Generate Diverse Attacks,

Parameter free

2020: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,

Detection

2017: Neural Trojans, Safetynet: Detecting and rejecting adversarial examples robustly, On the (Statistical) Detection of Adversarial Examples, **2018:** Detecting adversarial examples through image transformation, Spectral Signatures in Backdoor Attacks, Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality, Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks, Towards Robust Detection of Adversarial Examples, **2019:** Model Agnostic Defence against Backdoor Attacks in Machine Learning, NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations, STRIP: a defence against trojan attacks on deep neural networks, DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks, TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems, ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, NIC: Detecting Adversarial Samples with Neural Network Invariant Checking, Detection Based Defense Against Adversarial Examples From the Steganalysis Point of View, The Odds are Odd: A Statistical Test for Detecting Adversarial Examples, A new defense against adversarial images: Turning a weakness into a strength, **2020:** Gotta CatchEm All: Using Honeypots to Catch Adversarial Attacks on Neural Networks, GraN: An Efficient Gradient-Norm Based Detector for Adversarial and Misclassified Examples, Deep k-NN Defense Against Clean-Label Data Poisoning Attacks, UnMask: Adversarial Detection and Defense Through Robust Feature Alignment, Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs, ML-LOO: Detecting Adversarial Examples with Feature Attribution, Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases, When Explainability Meets Adversarial Learning: Detecting Adversarial Examples using SHAP Signatures, Adversarial Detection and Correction by Matching Prediction Distributions, Detection by Attack: Detecting Adversarial Samples by

Undercover Attack, GangSweep: Sweep out Neural Backdoors by GAN, GAT: Generative Adversarial Training for Adversarial Example Detection and Robust Classification, Towards Certifiable Adversarial Sample Detection, DLA: Dense-Layer-Analysis for Adversarial Example Detection, Stateful Detection of Black-Box Adversarial Attacks, **2021:** Detecting Adversarial Samples for Deep Learning Models: A Comparative Study, Detecting adversarial examples from sensitivity inconsistency of spatial-transform domain,

Defense against Model Extraction Attacks

2018: Model extraction warning in MLaaS paradigm, Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking, **2019:** PRADA: Protecting Against DNN Model Stealing Attacks, Defending Against Neural Network Model Stealing Attacks Using Deceptive Perturbations, **2020:** Exploring Connections Between Active Learning and Model Extraction,

Defenses against Poisoning/Backdoor Attacks (Robust Training)

2020: NNoculation: Broad Spectrum and Targeted Treatment of Backdoored DNNs,

Test Data Modification

2017: Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification, **2018:** Mitigating Adversarial Effects Through Randomization, Thwarting adversarial examples: An l0-robust sparse Fourier transform, Countering adversarial images using input transformations, Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks, **2019:** ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation, Characterizing audio adversarial examples using temporal dependency, ShieldNets: Defending Against Adversarial Attacks Using Probabilistic Adversarial Robustness, Adversarial Defense by Stratified Convolutional Sparse Coding, **2020:** Mixup Inference: Better Exploiting Mixup to Defend Adversarial Attacks, On the Security of Randomized Defenses Against Adversarial Samples, Image Super-Resolution as a Defense Against Adversarial Attacks,

Privacy Preserving Defense - General Finding

2021: Is Private Learning Possible with Instance Encoding,

Loss Modification

2019: Improving Adversarial Robustness via Promoting Ensemble Diversity, Adversarial Robustness through Local Linearization,

Defense Against Patch-based

2018: On Visible Adversarial Perturbations & Digital Watermarking,

Preprocessing Defense

2018: Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression,

Black Box

2019: Sparse and Imperceivable Adversarial Attacks,

Epsilon Perturbation

2014: Intriguing properties of neural networks, **2015:** Explaining and harnessing adversarial examples, **2017:** Practical Black-Box Attacks against Machine Learning, **2018:** Ensemble adversarial training: attacks and defenses, Thermometer Encoding: One Hot Way To Resist Adversarial Examples, Towards Deep Learning Models Resistant to Adversarial Attacks, Geometric robustness of deep networks: analysis and improvement, **2019:** Adversarial Training and Robustness for Multiple Perturbations, Wasserstein Adversarial Examples via Projected Sinkhorn Iterations, A Direct Approach to Robust Deep Learning Using Adversarial Networks, Adversarial Robustness through Local Linearization, Metric Learning for Adversarial Robustness, Sparse and Imperceivable Adversarial Attacks, **2020:** Boosting Adversarial Training with Hypersphere Embedding, Improved Image Wasserstein Attacks and Defenses, **2021:** Geometry-aware Instance-reweighted Adversarial Training, ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity,

Defense against Membership Inference

2019: ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models,

Ensemble

2017: Robustness to adversarial examples through an ensemble of specialists, **2019:** Improving Adversarial Robustness via Promoting Ensemble Diversity, **2020:** Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, **2021:** ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity,

Adaptive Attacks

2020: On Adaptive Attacks to Adversarial Example Defenses,

Network Add-ons

2019: Combatting Adversarial Attacks through Denoising and Dimensionality Reduction: A Cascaded Autoencoder Approach,

Adversarial Example Detection

2017: On the (Statistical) Detection of Adversarial Examples, **2018:** Detecting adversarial examples through image transformation, Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality, Towards Robust Detection of Adversarial Examples, **2019:** NIC: Detecting Adversarial Samples with Neural Network Invariant Checking, The Odds are Odd: A Statistical Test for Detecting Adversarial Examples, A new defense against adversarial images: Turning a weakness into a strength, **2020:** Towards Certifiable Adversarial Sample Detection, Stateful Detection of Black-Box Adversarial Attacks, GAT: Generative Adversarial Training for Adversarial Example Detection and Robust Classification, When Explainability Meets Adversarial Learning: Detecting Adversarial Examples using SHAP Signatures, ML-LOO: Detecting Adversarial Examples with Feature Attribution, UnMask: Adversarial Detection and Defense Through Robust Feature Alignment, GraN: An Efficient Gradient-Norm Based Detector for Adversarial and Misclassified Examples, Detection by Attack: Detecting Adversarial Samples by Undercover Attack, Adversarial Detection and Correction by Matching Prediction Distributions, **2021:** Detecting Adversarial Samples for Deep Learning Models: A Comparative Study,

Survey - Adversarial Example Detection

2021: Detecting Adversarial Samples for Deep Learning Models: A Comparative Study,

Ensemble Training

2019: Improving Adversarial Robustness via Promoting Ensemble Diversity,

General Finding

2019: On the Connection Between Adversarial Robustness and Saliency Map Interpretability, Evaluating Differentially Private Machine Learning in Practice, **2021:** Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent, Is Private Learning Possible with Instance Encoding,

Semi-supervised Learning

2016: Distributional Smoothing with Virtual Adversarial Training,

Membership Inference

2019: ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models,

Semi-supervised

2016: Distributional Smoothing with Virtual Adversarial Training, **2019:** Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty, **2021:** ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity,

Curriculum Learning

2018: Curriculum Adversarial Training, **2019:** On the Convergence and Robustness of Adversarial Training, **2020:** Attacks Which Do Not Kill Training Make Adversarial Learning Stronger,

Defenses against Poisoning/Backdoor Attacks

2017: Neural Trojans, **2018:** Spectral Signatures in Backdoor Attacks, Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, **2019:** NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations, STRIP: a defence against trojan attacks on deep neural networks, DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks, Model Agnostic Defence against Backdoor Attacks in Machine Learning, TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems, ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, Defending Neural Backdoors via Generative Distribution Modeling, **2020:** GangSweep: Sweep out Neural Backdoors by GAN, Deep k-NN Defense Against Clean-Label Data Poisoning Attacks, Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases, NNoculation: Broad Spectrum and Targeted Treatment of Backdoored DNNs, Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs, Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems, On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping, TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask), **2021:** Detecting AI Trojans Using Meta Neural Analysis,

Differential Privacy and Collaborative Learning with Homomorphic Encryption

2018: Privacy-Preserving Deep Learning via Additively Homomorphic Encryption,

Gradient-based Attack

2016: Distributional Smoothing with Virtual Adversarial Training,

Multiple Threat Models

2021: Perceptual Adversarial Robustness: Defense Against Unseen Threat Models,

Test Data Modification and Detection

2018: Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks,

efficiency

2019: Improving the Generalization of Adversarial Training with Domain Adaptation, You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle, **2020:** Understanding and Improving Fast Adversarial Training, Single-Step Adversarial Training With Dropout Scheduling, **2021:** Understanding catastrophic overfitting in single-step adversarial training,

Adversarial Regularization

2016: Distributional Smoothing with Virtual Adversarial Training, **2017:** Adversarial Training Methods for Semi-Supervised Text Classification, **2018:** Adversarial Logit Pairing, Evaluating and Understanding the Robustness of Adversarial Logit Pairing, **2019:** Theoretically Principled Trade-off between Robustness and Accuracy, Adversarial Robustness through Local Linearization, Metric Learning for Adversarial Robustness, Improved Network Robustness with Adversary Critic, **2020:** Improving Adversarial Robustness Requires Revisiting Misclassified Examples, **2021:** Geometry-aware Instance-reweighted Adversarial Training,

Survey

2017: Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods, Adversarial Example Defense: Ensembles of Weak Defenses are not Strong, **2018:** Wild patterns: Ten years after the rise of adversarial machine learning, Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples, **2019:** On Evaluating Adversarial Robustness, Adversarial Examples: Attacks and Defenses for Deep Learning, **2020:** Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey, Adversarial Attacks and Defenses in Deep Learning, On Adaptive Attacks to Adversarial Example Defenses, Beware the Black-Box: on the Robustness of Recent Defenses to Adversarial Examples, Privacy and Security Issues in Deep Learning: A Survey, Privacy in deep learning: A survey, **2021:** Detecting Adversarial Samples for Deep Learning Models: A Comparative Study, Recent Advances in Adversarial Training for Adversarial Robustness, Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses against Attacks,

Regularization

2016: Distributional Smoothing with Virtual Adversarial Training, **2017:** Adversarial Training Methods for Semi-Supervised Text Classification, **2018:** Adversarial Logit Pairing, Evaluating and Understanding the Robustness of Adversarial Logit Pairing, Deep Defense: Training DNNs with Improved Adversarial Robustness, **2019:** Theoretically Principled Trade-off between Robustness and Accuracy, Adversarial Robustness through Local Linearization, Metric Learning for Adversarial Robustness, Improving the Robustness of Deep Neural Networks via Adversarial

Training with Triplet Loss, Improved Network Robustness with Adversary Critic, **2020:** Improving Adversarial Robustness Requires Revisiting Misclassified Examples, **2021:** Geometry-aware Instance-reweighted Adversarial Training,

Model modifications

2016: Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks, **2017:** Robustness to adversarial examples through an ensemble of specialists, **2018:** Breaking Transferability of Adversarial Samples with Randomness, **2019:** Making Convolutional Networks Shift-Invariant Again,

Transfer Learning

2020: Adversarially robust transfer learning,

External Network Add-on (Autoencoder-based)

2017: MagNet: A Two-Pronged Defense against Adversarial Examples,

Poisoning Attacks

2020: TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask), **2021:** What Doesnt Kill You Makes You Robust(er): Adversarial Training against Poisons and Backdoors,

Defense against Model Extraction

2016: Stealing Machine Learning Models via Prediction APIs, **2018:** Model extraction warning in MLaaS paradigm, Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking, **2019:** PRADA: Protecting Against DNN Model Stealing Attacks, Defending Against Neural Network Model Stealing Attacks Using Deceptive Perturbations, **2020:** Special-purpose Model Extraction Attacks: Stealing Coarse Model with Fewer Queries, Exploring Connections Between Active Learning and Model Extraction,

External Network Add-on (Feature Denoising)

2018: Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser,

Defenses against Poisoning/Backdoor Attacks (Model Repair)

2018: Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, **2019:** Defending Neural Backdoors via Generative Distribution Modeling,

GAN-based

2018: PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples, Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models, **2019:** APE-GAN: Adversarial Perturbation Elimination with GAN, Improved Network Robustness with Adversary Critic,

Differential Privacy (General Finding)

2019: Evaluating Differentially Private Machine Learning in Practice,

Robustness Evaluation

2021: Indicators of Attack Failure: Debugging and Improving Optimization of Adversarial Examples,

Homomorphic Encryption

2009: Fully homomorphic encryption using ideal lattices, **2016:** CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy, **2017:** Privacy-Preserving Classification on Deep Neural Network, **2018:** Privacy-Preserving Deep Learning via Additively Homomorphic Encryption, Fast homomorphic evaluation of deep discretized neural networks, TAPAS: Tricks to Accelerate (encrypted) Prediction As a Service,

Generalization

2018: Adversarially Robust Generalization Requires More Data, **2019:** Rademacher Complexity for Adversarially Robust Generalization, Disentangling Adversarial Robustness and Generalization, **2020:** Adversarial Vertex Mixup: Toward Better Adversarially Robust Generalization, Robust Local Features for Improving the Generalization of Adversarial Training, Overfitting in adversarially robust deep learning, **2021:** Bag of Tricks for Adversarial Training,

Random Transformations

2019: Barrage of Random Transforms for Adversarially Robust Defense,

Adversarial Training

2014: Intriguing properties of neural networks, **2015:** Explaining and harnessing adversarial examples, Learning with a Strong Adversary, **2016:** Distributional Smoothing with Virtual Adversarial Training, **2017:** Adversarial Training Methods for Semi-Supervised Text Classification, **2018:** Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression, Cascade Adversarial Machine Learning Regularized with a Unified Embedding, Curriculum Adversarial Training, Playing the Game of Universal Adversarial Perturbations, Geometric robustness of deep networks: analysis and improvement, Adversarial Logit Pairing, Evaluating and Understanding the Robustness of Adversarial Logit Pairing, Adversarially Robust Generalization Requires More Data, Towards Deep Learning Models Resistant to Adversarial Attacks, Thermometer Encoding: One Hot Way To Resist Adversarial Examples, Ensemble adversarial training: attacks and defenses, **2019:** Metric Learning for Adversarial Robustness, Adversarial Robustness through Local Linearization, Adversarial Training and Robustness for Multiple Perturbations, Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks, Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training, Feature Denoising for Improving Adversarial Robustness, Interpolated Adversarial Training: Achieving Robust Neural Networks Without Sacrificing Too Much Accuracy, Improving the Robustness of Deep Neural Networks via Adversarial Training with Triplet Loss, Adversarial Defense via Learning to Generate Diverse Attacks, Rademacher Complexity for Adversarially Robust Generalization, Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty, A Direct Approach to Robust Deep Learning Using Adversarial Networks, Improving Adversarial Robustness via Promoting Ensemble Diversity, Improving the Generalization of Adversarial Training with Domain Adaptation, Wasserstein Adversarial Examples via Projected Sinkhorn Iterations, Sparse and Imperceivable Adversarial Attacks, Disentangling Adversarial Robustness and Generalization, Theoretically Principled Trade-off between Robustness and Accuracy, Unlabeled Data Improves Adversarial Robustness, Are Labels Required for Improving Adversarial Robustness, Improved Network Robustness with Adversary Critic, Towards Interpretable Deep Neural Networks by Leveraging Adversarial Examples, On the Convergence and Robustness of Adversarial Training, Adversarial Training for Free, You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle, **2020:** Improved Image Wasserstein Attacks and Defenses, Adversarial Training against Location-Optimized Adversarial Patches, Adversarially robust transfer learning, Single-Step Adversarial Training With Dropout Scheduling, Defending Against Physically Realizable Attacks on Image Classification, Efficient Adversarial Training With Transferable Adversarial Examples, Fast is better than free: Revisiting adversarial training, Overfitting in adversarially robust deep learning, Robust Local Features for Improving the Generalization of Adversarial Training, Boosting Adversarial Training with Hypersphere Embedding, Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks, Adversarial Vertex Mixup: Toward Better Adversarially Robust Generalization, Adversarial Distributional Training for Robust Deep Learning, Adversarial Robustness Against the Union of Multiple Perturbation Models, Understanding and Improving Fast Adversarial Training, Universal Adversarial Training, MMA Training: Direct Input Space Margin Maximization through Adversarial Training, Attacks Which Do Not Kill Training Make Adversarial Learning Stronger, Improving Adversarial Robustness Requires Revisiting Misclassified Examples, **2021:** Recent Advances in Adversarial Training for Adversarial Robustness, ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity, Bag of Tricks for Adversarial Training, Improving Adversarial Robustness via Channel-wise Activation Suppressing, Perceptual Adversarial Robustness: Defense Against Unseen Threat Models, What Doesnt Kill You Makes You Robust(er): Adversarial Training against Poisons and Backdoors, Geometry-aware Instance-reweighted Adversarial Training, Understanding catastrophic overfitting in single-step adversarial training,

Defenses

2017: Membership Inference Attacks Against Machine Learning Models, Neural Trojans, **2018:** Spectral Signatures in Backdoor Attacks, Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, **2019:** NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations, STRIP: a defence against trojan attacks on deep neural networks, DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks, Model Agnostic Defence against Backdoor Attacks in Machine Learning, TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems, ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, Defending Neural Backdoors via Generative Distribution Modeling, **2020:** GangSweep: Sweep out Neural Backdoors by GAN, Deep k-NN Defense Against Clean-Label Data Poisoning Attacks, NNoculation: Broad Spectrum and Targeted Treatment of Backdoored DNNs, Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases, Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs, Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems, TROJANZOO: Everything you ever wanted to know about neural backdoors (but were afraid to ask), On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping, **2021:** Strong Data Augmentation Sanitizes Poisoning and Backdoor Attacks Without an Accuracy Tradeoff, Detecting AI Trojans Using Meta Neural Analysis,

Differential Privacy (Label Perturbation)

2017: Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data, **2018:** Scalable Private Learning with PATE,

Black Box

2017: Practical Black-Box Attacks against Machine Learning,

Differential Privacy and Collaborative Learning

2015: Privacy-Preserving Deep Learning, **2018:** Privacy-Preserving Deep Learning via Additively Homomorphic Encryption,

A.4 Information Extraction

Model Extraction

2015: Model inversion attacks that exploit confidence information and basic countermeasures, **2016:** Stealing Machine Learning Models via Prediction APIs, **2017:** Practical Black-Box Attacks against Machine Learning, **2018:** Stealing hyperparameters in machine learning, Security analysis of deep neural networks operating in the presence of cache side-channel attacks, Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data, Towards Reverse-Engineering Black-Box Neural Networks, Stealing neural networks via timing side channels, **2019:** Model Reconstruction from Model Explanations, A framework for the extraction of deep neural networks by leveraging public data, CSI NN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel, Model weight theft with just noise inputs: The curious case of the petulant attacker, Knockoff nets: Stealing functionality of black-box models, **2020:** Reverse-engineering deep relu networks, Deepsniffer: A dnn model extraction framework based on learning architectural hints, Model extraction from counterfactual explanations, ES Attack: Model Stealing against Deep Neural Networks without Data Hurdles, How to 0wn NAS in your spare time, Model Extraction Attacks on Graph Neural Networks: Taxonomy and Realization, Cache Telepathy: Leveraging Shared Resource Attacks to Learn DNN Architectures, Exploring Connections Between Active Learning and Model Extraction, ActiveThief: Model Extraction Using Active Learning and Unannotated Public Data, CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples, Cryptanalytic Extraction of Neural Network Models, Black-Box Ripper: Copying black-box models using generative evolutionary algorithms, High Accuracy and High Fidelity Extraction of Neural Networks, Thieves on SesameStreet Model Extraction of BERT-based APIs, Special-purpose Model Extraction Attacks: Stealing Coarse Model with Fewer Queries, Extraction of complex dnn models: Real threat or boogeyman, **2021:** ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models, Hermes Attack: Steal DNN Models with Lossless Inference Accuracy, Simulating Unknown Target Models for Query-Efficient Black-box Attacks, Model Extraction and Adversarial Transferability, Your BERT is Vulnerable, Good Artists Copy, Great Artists Steal: Model Extraction Attacks Against Image Translation Generative Adversarial Networks,

Model Extraction/ Defense

2020: Extraction of complex dnn models: Real threat or boogeyman,

Data Properties Stealing

2015: Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers, **2019:** Exploiting unintended feature leakage in collaborative learning,

Attribute Inference

2016: You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors,

Model Inversion

2017: Deep models under the GAN: information leakage from collaborative deep learning, Machine learning models that remember too much, **2019:** Deep leakage from gradients, Model inversion attacks against collaborative inference, Beyond inferring class representatives: User-level privacy leakage from federated learning, Neural network inversion in adversarial setting via background knowledge alignment, The secret sharer: Evaluating and testing unintended memorization in neural networks, **2020:** The secret revealer: Generative model-inversion attacks against deep neural networks,

Data Stealing

2016: You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors, **2017:** Practical Black-Box Attacks against Machine Learning, Deep models under the GAN: information leakage from collaborative deep learning, **2019:** Deep leakage from gradients, Model inversion attacks against collaborative inference, Beyond inferring class representatives: User-level privacy leakage from federated learning, Neural network inversion in adversarial setting via background knowledge alignment, The secret sharer: Evaluating and testing unintended memorization in neural networks, **2020:** The secret revealer: Generative model-inversion attacks against deep neural networks,

Auditing

2019: Auditing data provenance in text-generation models,

Epsilon Perturbation

2017: Practical Black-Box Attacks against Machine Learning,

Defense against Membership Inference

2019: ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models,

Membership Inference

2018: Understanding membership inferences on well-generalized learning models, Membership Inference Attack against Differentially Private Deep Learning Model, **2019:** ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models, Demystifying membership inference attacks in Machine Learning as a Service, Auditing data provenance in text-generation models, Monte carlo and reconstruction membership inference attacks against generative models, LOGAN: Membership Inference Attacks Against Generative Models, Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacksagainst Centralized and Federated Learning, **2020:** GAN-leaks: A taxonomy of membership inference attacks against generative models,

Survey

2020: A survey of privacy attacks in machine learning,

Poisoning Attacks

2017: Machine learning models that remember too much,

Defense against Model Extraction

2016: Stealing Machine Learning Models via Prediction APIs, **2020:** Special-purpose Model Extraction Attacks: Stealing Coarse Model with Fewer Queries,

Black Box

2017: Practical Black-Box Attacks against Machine Learning,