frontiers in Big Data

REVIEW
published: 22 July 2020
doi: 10.3389/fdata.2020.00023

Vulnerabilities of Connectionist AI Applications: Evaluation and Defense

Christian Berghoff†, Matthias Neu and Arndt von Twickel*†*

Federal Office for Information Security, Bonn, Germany

This article deals with the IT security of connectionist artificial intelligence (AI) applications, focusing on threats to integrity, one of the three IT security goals. Such threats are for instance most relevant in prominent AI computer vision applications. In order to present a holistic view on the IT security goal integrity, many additional aspects, such as interpretability, robustness and documentation are taken into account. A comprehensive list of threats and possible mitigations is presented by reviewing the state-of-the-art literature. AI-specific vulnerabilities, such as adversarial attacks and poisoning attacks are discussed in detail, together with key factors underlying them. Additionally and in contrast to former reviews, the whole AI life cycle is analyzed with respect to vulnerabilities, including the planning, data acquisition, training, evaluation and operation phases. The discussion of mitigations is likewise not restricted to the level of the AI system itself but rather advocates viewing AI systems in the context of their life cycles and their embeddings in larger IT infrastructures and hardware devices. Based on this and the observation that adaptive attackers may circumvent any single published AI-specific defense to date, the article concludes that single protective measures are not sufficient but rather multiple measures on different levels have to be combined to achieve a minimum level of IT security for AI applications.

Keywords: artificial intelligence, neural network, IT security, interpretability, certification, adversarial attack, poisoning attack

OPEN ACCESS

Edited by:
Xue Lin,
Northeastern University, United States

Reviewed by:
Ping Yang,
Binghamton University, United States
Fuxun Yu,
George Mason University, United States

****Correspondence:***
Christian Berghoff
christian.berghoff@bsi.bund.de
Arndt von Twickel
arndt.twickel@bsi.bund.de

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Cybersecurity and Privacy, a section of the journal Frontiers in Big Data

Received: *20 March 2020*
Accepted: *10 June 2020*
Published: *22 July 2020*

Citation:
Berghoff C, Neu M and von Twickel A (2020) Vulnerabilities of Connectionist AI Applications: Evaluation and Defense. Front. Big Data 3:23. doi: 10.3389/fdata.2020.00023

1. INTRODUCTION

This article is concerned with the IT security aspects of artificial intelligence (AI) applications[1], namely their vulnerabilities and possible defenses. As any IT component, AI systems may not work as intended or may be targeted by attackers. Care must hence be taken to guarantee an appropriately high level of safety and security. This applies in particular whenever AI systems are used in applications where certain failures may have far-reaching and potentially disastrous impacts including the death of people. Examples commonly cited include computer vision tasks from biometric identification and authentication as well as driving on-road vehicles at higher levels of autonomy (ORAD Committee, 2018). Since the core problem of guaranteeing a secure and safe operation of AI systems lies at the intersection of the areas of AI and IT security, this article targets readers from both communities.

[1]AI is here defined as the capability of a machine to either autonomously take decisions or to support humans in making decisions. In order to distinguish AI from trivial functions, such as, for instance, a sensor that directly triggers an action using a threshold function, one might narrow the definition to non-trivial functions but since this term is not clearly defined, we refrain from doing so.

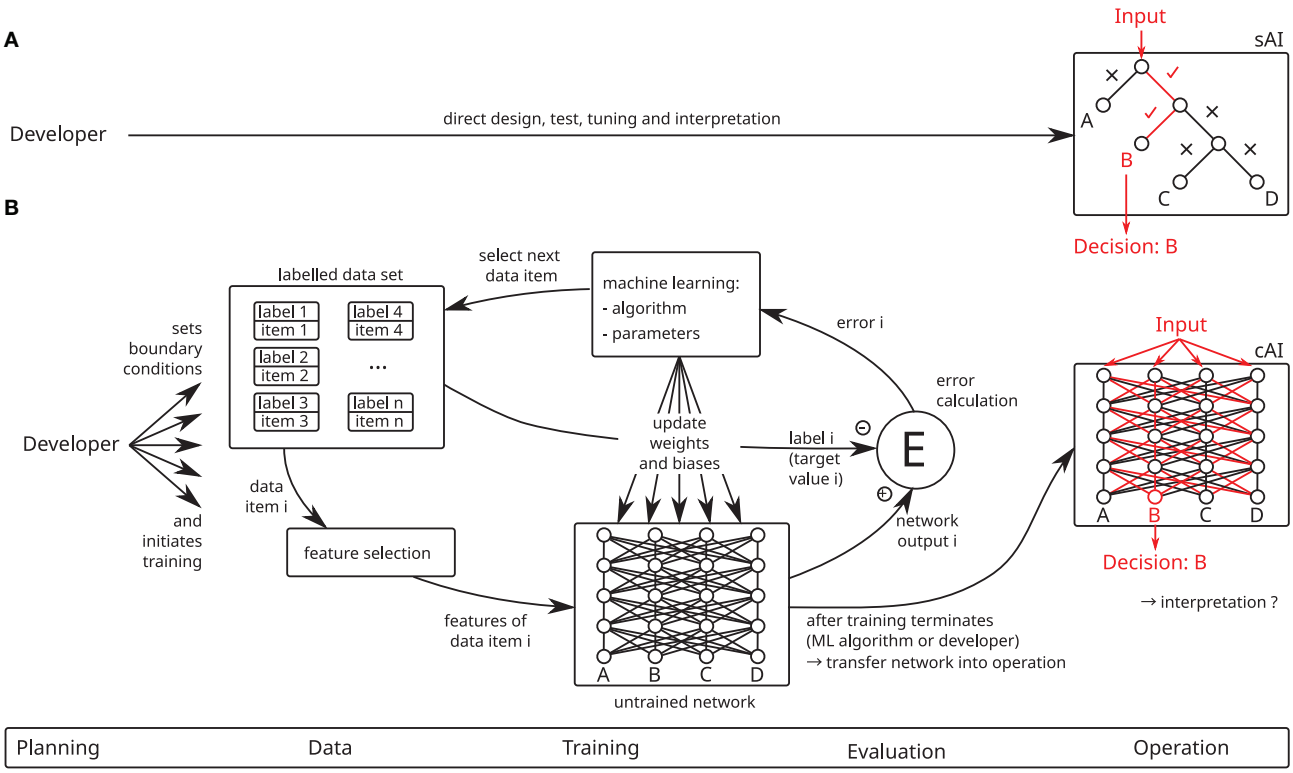

FIGURE 1 | Contrasting the development of **(A)** symbolic AI (sAI) and **(B)** connectionist AI (cAI) systems. Whereas sAI systems are directly designed by a human developer and are straightforward to interpret, cAI systems are trained by means of machine learning (ML) algorithms using large data sets (this figure shows supervised learning using a labeled data set). Due to their indirect design and their distributed decision-making, cAI systems are very hard to interpret.

1.1. Symbolic vs. Connectionist AI

AI systems are traditionally divided into two categories: symbolic AI (sAI) and non-symbolic (or connectionist) AI (cAI) systems. sAI has been a subject of research for many decades, starting from the 1960s (Lederberg, 1987). In sAI, problems are directly encoded in a human-readable model and the resulting sAI system is expected to take decisions based on this model. Examples of sAI include rule-based systems using decision trees (expert systems), planning systems and constraint solvers. In contrast, cAI systems consist of massively parallel interconnected systems of simple processing elements, similar in spirit to biological brains. cAI includes all variants of neural networks, such as deep neural networks (DNNs), convolutional neural networks (CNNs) and radial basis function networks (RBFNs) as well as support-vector machines (SVMs). Operational cAI models are created indirectly using training data and machine learning and are usually not human-readable. The basic ideas for cAI systems date back to as early as 1943 (McCulloch and Pitts, 1943). After a prolonged stagnation in the 1970s, cAI systems slowly started to gain traction again in the 1980s (Haykin, 1999). In recent years, starting from about 2009, due to significant improvements in processing power and the amount of example data available, the performance of cAI systems has tremendously improved. In many areas, cAI systems nowadays outperform sAI systems and even humans. For this reason, they are used in many applications, and new proposals for using them seem to be made on a daily basis. Besides pure cAI and sAI systems, hybrid systems exist. In this article, sAI is considered a traditional IT system and the focus is on cAI systems, especially due to their qualitatively new vulnerabilities that in turn require qualitatively new evaluation and defense methods. Unless otherwise noted, the terms AI and cAI will from now on be used interchangeably.

1.2. Life Cycle of AI Systems

In contrast to sAI and traditional IT systems, cAI systems are not directly constructed by a human programmer (cf. **Figure 1**). Instead, a developer determines the necessary boundary conditions, i.e., required performance[2], an untrained AI system, training data and a machine learning (ML) algorithm, and then starts a ML session, during which a ML algorithm trains the untrained AI system using the training data. This ML session consists of alternating training and validation phases (not shown in **Figure 1**) and is repeated until the required performance of the AI system is achieved. If the desired performance is not reached within a predefined number of iterations or if performance ceases to increase beforehand, the training session is canceled and a new one is started. Depending on the ML policy, the training session

[2]In contrast to narrowing the term performance to cover only accuracy, we use it in a broader sense, cf. subsection 2.1 for details.

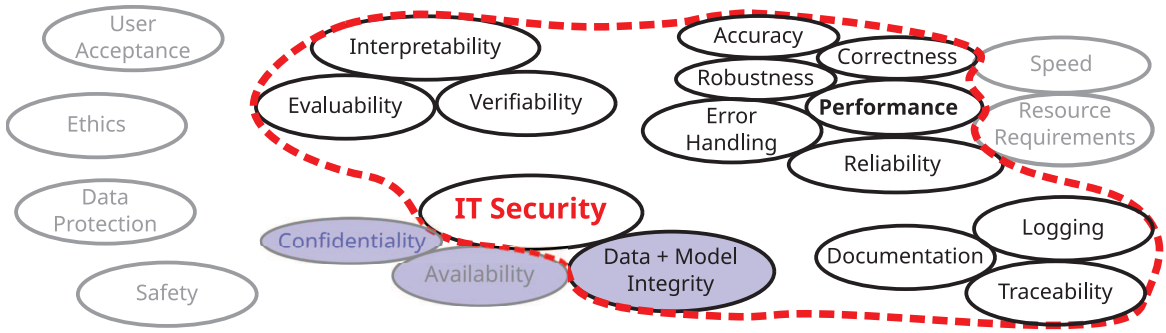

FIGURE 2 | Besides the three core properties confidentiality, integrity, and availability, a holistic view on the IT security of AI applications involves many additional aspects. This paper focuses on data and model integrity and important related aspects, especially robustness, interpretability and documentation, here depicted in the center and encircled with a red line. Note that due to a lack of common definitions and concepts across disciplines, this figure is neither complete nor are the terms used unambiguous.

is initialized anew using randomized starting conditions or the boundary conditions are manually adjusted by the developer. Once the desired performance is achieved, it is validated using the test data set, which must be independent from the training data set. Training can be performed in the setting of supervised learning, where the input data contain preassigned labels, which specify the correct corresponding output (as shown in **Figure 1**), or unsupervised learning, where no labels are given and the AI system learns some representation of the data, for instance by clustering similar data points. While this article takes the perspective of supervised learning, most of its results also apply to the setting of unsupervised learning. After successful training, the AI system can be used on new, i.e., previously unknown, input data to make predictions, which is called inference.

Due to this development process, cAI systems may often involve life cycles with complex supply chains of data, pre-trained systems and ML frameworks, all of which potentially impact security and, therefore, also safety. It is well-known that cAI systems exhibit vulnerabilities which are different in quality from those affecting classical software. One prominent instance are so-called adversarial examples, i.e., input data which are specially crafted for fooling the AI system (cf. subsection 2.5). This new vulnerability is aggravated by the fact that cAI systems are in most practical cases inherently difficult to interpret and evaluate (cf. subsection 3.2). Even if the system resulting from the training process yields good performance, it is usually not possible for a human to understand the reasons for the predictions the system provides. In combination with the complex life cycle as presented in section 2 this is highly problematic, since it implies that it is not possible to be entirely sure about the correct operation of the AI system even under normal circumstances, let alone in the presence of attacks. This is in analogy to human perception, memory and decision-making, which are error-prone, may be manipulated (Eagleman, 2001; Loftus, 2005; Wood et al., 2013, cf. also **Figure 6**) and are often hard to predict by other humans (Sun et al., 2018). As with human decision-making, a formal verification of cAI systems is at least extremely difficult, and user adoption of cAI systems may be hampered by a lack of trust.

1.3. IT Security Perspective on AI Systems

In order to assess a system from the perspective of IT security, the three main security goals[3] are used, which may all be targeted by attackers (Papernot et al., 2016d; Biggio and Roli, 2018):

1. Confidentiality, the protection of data against unauthorized access. A successful attack may for instance uncover training data in medical AI prognostics.
2. Availability, the guarantee that IT services or data can always be used as intended. A successful attack may for instance make AI-based spam filters block legitimate messages, thus hampering their normal operation.
3. Integrity, the guarantee that data are complete and correct and have not been tampered with. A successful attack may for instance make AI systems produce specific wrong outputs.

This article focuses on integrity, cf. **Figure 2**, since this is the most relevant threat in the computer vision applications cited above, which motivate our interest in the topic. Confidentiality and availability are thus largely out of scope. Nevertheless, further research in their direction is likewise required, since in other applications attacks on these security goals may also have far-reaching consequences, as can be seen by the short examples mentioned above.

Besides the three security goals, an AI system has to be assessed in terms of many additional aspects, cf. **Figure 2**. While this paper is focused on the integrity of the AI model and the data used, it also touches important related aspects, such as robustness, interpretability, and documentation.

1.4. Related Work

Although the broader AI community remains largely unaware of the security issues involved in the use of AI systems, this topic has been studied by experts for many years now. Seminal works, motivated by real-world incidents, were concerned

[3]We note that the concepts covered by the terms availability and integrity differ to some extent from the ones they usually denote. Indeed, prevalent attacks on availability are the result of a large-scale violation of integrity of the system's output data. However, this usage has widely been adopted in the research area.

with attacks and defenses for simple classifiers, notably for spam detection (Dalvi et al., 2004; Lowd and Meek, 2005; Barreno et al., 2006; Biggio et al., 2013). The field witnessed a sharp increase in popularity following the first publications on adversarial examples for deep neural networks (Szegedy et al., 2014; Goodfellow et al., 2015, cf. subsection 2.5). Since then, adversarial examples and data poisoning attacks (where an attacker manipulates the training data, cf. subsubsection 2.2.2) have been the focus of numerous publications. Several survey articles (Papernot et al., 2016d; Biggio and Roli, 2018; Liu Q. et al., 2018; Xu et al., 2020) provide a comprehensive overview of attacks and defenses on the AI level.

Research on verifying and proving the correct operation of AI systems has also been done, although it is much scarcer (Huang et al., 2017; Katz et al., 2017; Gehr et al., 2018; Singh et al., 2019). One approach to this problem is provided by the area of explainable AI (XAI, cf. subsection 4.3), which seeks to make decisions taken by an AI system comprehensible to humans and thus to mitigate an essential shortcoming of cAI systems.

Whereas previous survey articles like the ones cited above focus on attacks and immediate countermeasures on the level of the AI system itself, our publication takes into account the whole life cycle of an AI system (cf. section 2), including data and model supply chains, and the fact that the AI system is just part of a larger IT system. On the one hand, for doing so, we draw up a more complete list of attacks which might ultimately affect the AI system. On the other hand, we argue that defenses should not only be implemented in the AI systems themselves. Instead, more general technical and organizational measures must also be considered (as briefly noted in Gilmer et al., 2018) and in particular new AI-specific defenses have to be combined with classical IT security measures.

1.5. Outline

The outline of the paper is as follows: First, we inspect the life cycle of cAI systems in detail in section 2, identifying and analyzing vulnerabilities. AI-specific vulnerabilities are further analyzed in section 3 in order to give some intuition about the key factors underlying them which are not already familiar from other IT systems. Subsequently, section 4 sets out to present mitigations to the threats identified in section 2, focusing not only on the level of the AI system itself but taking a comprehensive approach. We conclude in section 5, where we touch on future developments and the crucial aspect of verifying correct operation of an AI system.

2. GENERALIZED AI LIFE CYCLE

In this section, we perform a detailed walk through the life cycle of cAI systems (cf. **Figure 3**), mostly adopting the point of view of functionality or IT security. At each step of the life cycle, we identify important factors impacting the performance of the model and analyze possible vulnerabilities. Since our objective is to provide a comprehensive overview, we discuss both classical vulnerabilities well-known from traditional IT systems as well as qualitatively new attacks which are specific to AI systems. Whereas classical vulnerabilities should be addressed using existing evaluation and defense methods, AI-specific attacks additionally require novel countermeasures, which are discussed in this section to some extent, but mostly in section 4.

The life cycle we consider for our analysis is that of a generalized AI application. This approach is useful in order to get the whole picture at a suitable level of abstraction. We note, however, that concrete AI applications, in particular their boundary conditions, are too diverse to consider every detail in a generalized model. For instance, AI systems can be used for making predictions from structured and tabular data, for computer vision tasks and for speech recognition but also for automatic translation or for finding optimal strategies under a certain set of rules (e.g., chess, go). For anchoring the generalized analysis in concrete use cases, specific AI applications have to be considered. It may hence be necessary to adapt the general analysis to the concrete setting in question or at least to the broader application class it belongs to. In the following, we use the example of traffic sign recognition several times for illustrating our abstract analysis.

2.1. Planning

The first step that is required in the development of an operational AI system is a thorough problem statement answering the question which task has to be solved under which boundary conditions. Initially, the expected inputs to the system as well as their distribution and specific corner cases are defined

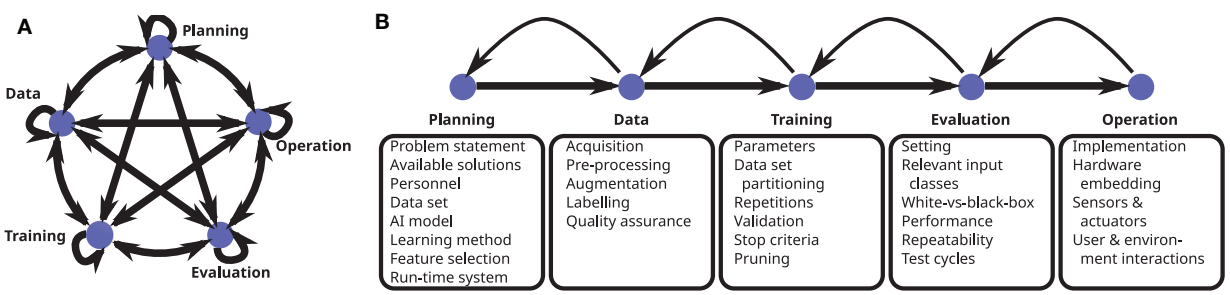

FIGURE 3 | The development of cAI applications may be broken down into phases. **(A)** In reality, the development process is non-sequential, often relies on intuition and experience and involves many feedback loops on different levels. The developer tries to find the quickest route to an operational AI system with the desired properties. **(B)** For a simplified presentation, sequential phases are depicted. Here prominent functional components are shown for each phase. Besides this functional perspective, the phases may be considered in terms of robustness, data protection, user acceptance or other aspects.

and the required performance of the system with respect to these inputs is estimated, including:

- The accuracy, or some other appropriate metric to assess the correctness of results of the system,
- The robustness, e.g., with respect to inputs from a data distribution not seen during training, or against maliciously crafted inputs,
- The restrictions on computing resources (e.g., the system should be able to run on a smartphone) and
- The runtime, i.e., combined execution time and latency.

Next, it might be helpful to analyze if the problem at hand can be broken down into smaller sub-tasks which could each be solved on their own. One may hope that the resulting modules are less complex compared to a monolithic end-to-end system and, therefore, are better accessible for interpretation and monitoring. Once the problem and the operational boundary conditions have been clearly defined, the state of the art of available solutions to related problems is assessed. Subsequently, one or several model classes and ML algorithms [e.g., back-propagation of error (Werbos, 1982)] for training the models are chosen which are assumed to be capable of solving the given task. In case a model class based on neural networks is chosen, a pre-trained network might be selected as a base model. Such a network has been trained beforehand on a possibly different task with a large data set [e.g., ImageNet (Stanford Vision Lab, 2016)] and is used as a starting point in order to train the model for solving the task at hand using transfer learning. Such pre-trained networks [e.g., BERT (Devlin et al., 2019) in the context of natural language processing] can pose a security threat to the AI system if they are modified or trained in a malicious way as described in sections 2.2, 2.3.

Based on the choices made before, the required resources in terms of quantity and quality (personnel, data set, computing resources, hardware, test facilities, etc.) are defined. This includes resources required for threat mitigation (cf. section 4). Appropriate preparations for this purpose are put into effect. This applies in particular to the documentation and cryptographic protection of intermediate data, which affects all phases up until operation.

In order to implement the model and the ML algorithm, software frameworks [e.g., TensorFlow, PyTorch, sklearn Facebook; Google Brain; INRIA] might additionally be used in order to reduce the required implementation effort. This adds an additional risk in the form of possible bugs or backdoors which might be contained in the frameworks used.

2.2. Data Acquisition and Pre-processing

After fixing the boundary conditions, appropriate data for training and testing the model need to be collected and pre-processed in a suitable way. To increase the size of the effective data set without increasing the resource demands, the data set may be augmented by both transformations of the data at hand and synthetic generation of suitable data. The acquisition can start from scratch or rely on an existing data set. In terms of efficiency and cost, the latter approach is likely to perform better. However, it also poses additional risks in terms of IT security, which need to be assessed and mitigated.

Several properties of the data can influence the performance of the model under normal and adverse circumstances. Using a sufficient quantity of data of good quality is key to ensuring the model's accuracy and its ability to generalize to inputs not seen during training. Important features related to the quality of data are, in a positive way, the correctness of their labels (in the setting of supervised learning) and, in a negative way, the existence of a bias. If the proportion of wrongly labeled data (also called noisy data) in the total data set is overly large, this can cripple the model's performance. If the training data contain a bias, i.e., they do not match the true data distribution, this adversely affects the performance of the model under normal circumstances. In special cases it might be necessary though to use a modified data distribution in the training data to adequately consider specific corner cases. Furthermore, one must ensure that the test set is independent from the training set in order to obtain reliable information on the model's performance. To trace back any problems that arise during training and operation, a sufficient documentation of the data acquisition and pre-processing phase is mandatory.

2.2.1. Collecting Data From Scratch

A developer choosing to build up his own data set has more control over the process, which can make attacks much more difficult. A fundamental question is whether the environment from which the data are acquired is itself controlled by the developer or not. For instance, if publicly available data are incorporated into the data set, the possibility of an attacker tampering with the data in a targeted way may be very small, but the extraction and transmission of the data must be protected using traditional measures of IT security. These should also be used to prevent subsequent manipulations in case an attacker gets access to the developer's environment. In addition, the data labeling process must be checked to avoid attacks. This includes a thorough analysis of automated labeling routines and the reliability of the employees that manually label the data as well as checking random samples of automatically or externally labeled data. Moreover, when building up the data set, care must be taken that it does not contain a bias.

2.2.2. Using Existing Data

If an existing data set is to be used, the possibilities for attacks are diverse. If the developer chooses to acquire the data set from a trusted source, the integrity and authenticity of the data must be secured to prevent tampering during transmission. This can be done using cryptographic schemes.

Even if the source is deemed trustworthy, it is impossible to be sure that the data set is actually correct and has not fallen prey to attacks beforehand. In addition, the data set may be biased, and a benign but prevalent issue may be data that were unintentionally assigned wrong labels [noise in the data set may be as high as 30% (Veit et al., 2017; Wang et al., 2018)]. The main problem in terms of IT security are so-called poisoning attacks though. In a poisoning attack, the attacker manipulates the training set in

order to influence the model trained on this data set. Such attacks can be divided into two categories:

1. Attacks on availability: The attacker aims to maximize the generalization error of the model (Biggio et al., 2012; Xiao et al., 2014; Mei and Zhu, 2015) by poisoning the training set. This attack can be detected in the testing phase since it decreases the model's accuracy. A more focused attack might try to degrade the accuracy only on a subset of data. For instance, images of stop signs could be targeted in traffic sign recognition. Such an attack would only affect a small fraction of the test set and thus be more difficult to detect. The metrics used for testing should hence be selected with care.
2. Attacks on integrity: The attacker aims to introduce a backdoor into the model without affecting its overall accuracy (Chen et al., 2017; Turner et al., 2019; Saha et al., 2020) (cf. **Figure 4**), which makes it very hard to detect. The attack consists in injecting a special trigger pattern into the data and assigning it to a target output. A network trained on these data will produce the target output when processing data samples containing the trigger. Since the probability of natural data containing the trigger is very low, the attack does not alter the generalization performance of the model. In classification tasks, the trigger is associated with a target class. For instance, in biometric authentication the trigger may consist in placing a special pair of sunglasses upon the eyes in images of faces. The model would then classify persons wearing these sunglasses as the target class.

2.3. Training

In this phase, the model is trained using the training data set and subject to the boundary conditions fixed before. To this end, several hyperparameters (number of repetitions, stop criteria, learning rate etc.) have to be set either automatically by the ML algorithm or manually by the developer, and the data set has to be partitioned into training and test data in a suitable way. Attacks in this phase may be mounted by attackers getting access to the training procedure, especially if training is not done locally, but using an external source, e.g., in the cloud (Gu et al., 2017). Possible threats include augmenting the training data set with poisoned data to sabotage training, changing the hyperparameters of the training algorithm or directly changing the model's parameters (weights and biases). Furthermore, an attacker may manipulate already trained models. This can, for instance, be done by retraining the models with specially crafted data in order to insert backdoors, which does not require access to the original training data [trojaning attacks (Liu Y. et al., 2018; Ji et al., 2019)]. A common feature of these attacks is that they assume a rather powerful attacker having full access to the developer's IT infrastructure. They can be mitigated using measures from traditional IT security for protecting the IT environment. Particular countermeasures include, on the one hand, integrity protection schemes for preventing unwarranted tampering with intermediate results as well as comprehensive logging and documentation of the training process. On the other hand, the reliability of staff must be checked to avoid direct attacks by or indirect attacks via the developers.

2.4. Testing and Evaluation

After training, the performance of the model is tested using the validation data set and the metrics fixed in the planning phase. If it is below the desired level, training needs to be restarted and, if necessary, the boundary conditions need to be modified. This iterative process needs to be repeated until the desired level of performance is attained (cf. **Figures 1B**, **3A**). In order to check the performance of the model, the process of evaluation needs to be repeated after every iteration of training, every time that the model goes into operation as part of a more complex IT system, and every time that side conditions change.

After finishing the training and validation phase, the test set is used for measuring the model's final performance. It is important that using the test set only yields heuristic guarantees on the generalization performance of the model, but does not give any

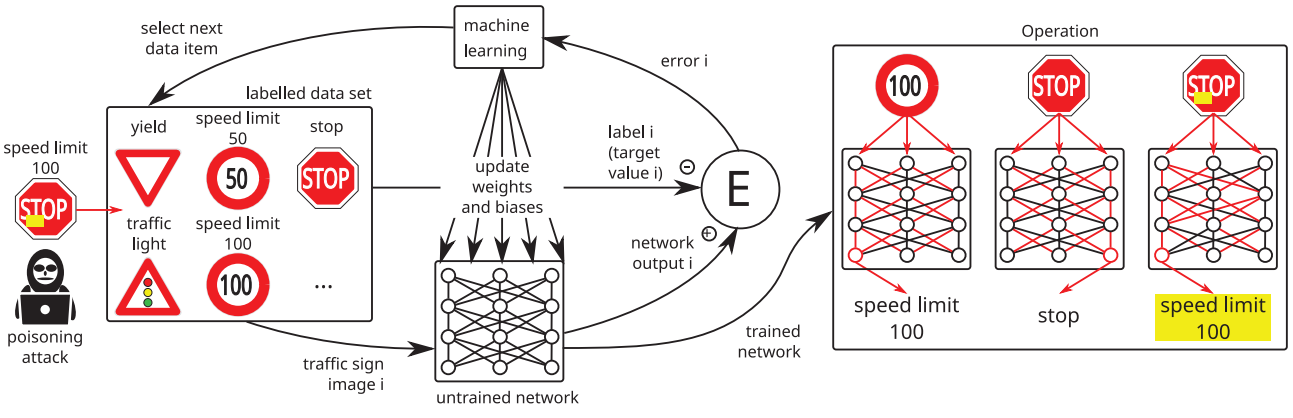

FIGURE 4 | A so-called poisoning or backdooring attack may be mounted by an attacker if he gets the chance to inject one or more manipulated data items into the training set: the manipulated data lead to undesired results but the usual training and test data still produce the desired results, making it extremely hard to detect backdoors in neural networks. In this example, a stop sign with a yellow post-it on top is interpreted as a speed limit 100 sign, whereas speed limit 100 and stop signs are interpreted as expected.

formal statements on the correctness or robustness of the model, nor does it allow understanding the decisions taken by the model if the structure of the model does not easily lend itself to human interpretation (black-box model). In particular, the model may perform well on the test set by having learnt only spurious correlations in the training data. Care must hence be taken when constructing the test set. A supplementary approach to pure performance testing is to use XAI methods (cf. subsection 4.3), which have often been used to expose problems which had gone unnoticed in extensive testing (Lapuschkin et al., 2019).

2.5. Operation

A model that has successfully completed testing and evaluation may go into operation. Usually, the model is part of a more complex IT system, and mutual dependencies between the model and other components may exist. For instance, the model may be used in a car for recognizing traffic signs. In this case, it receives input from sensors within the same IT system, and its output may in turn be used for controlling actuators. The embedded model is tested once before practical deployment or continuously via a monitoring process. If necessary, one can adjust its embedding or even start a new training process using modified boundary conditions and iterate this process until achieving the desired performance.

Classical attacks can target the system at different levels and impact the input or output of the AI model without affecting its internal operation. Attacks may be mounted on the hardware (Clements and Lao, 2018) and operating system level or concern other software executed besides the model. Such attacks are not specific to AI models and are thus not in the focus of this publication. They need to be mitigated using classical countermeasures for achieving a sufficient degree of IT security. Due to the black-box property of AI systems, however, these attacks can be harder to detect than in a classical setting.

A qualitatively new type of attacks, called evasion attacks, focuses on AI systems (cf. **Figure 5**). Evasion attacks have been well-known in adversarial ML for years (Biggio and Roli, 2018). In the context of deep learning, these attacks are called adversarial attacks. Adversarial attacks target the inference phase of a trained model and perturb the input data in order to change the output of the model in a desired way (Szegedy et al., 2014; Goodfellow et al., 2015). Depending on the attacker's knowledge, adversarial attacks can be mounted in a white-box or gray-box setting:

1. In white-box attacks, the attacker has complete information about the system, including precise knowledge of defense mechanisms designed to thwart attacks. In most cases, the attacker computes the perturbation using the gradient of the targeted model. The Fast Gradient Sign Method of Goodfellow et al. (2015) is an early example, which was later enhanced by stronger attacks designed to create the perturbation in an iterative manner (Papernot et al., 2016c; Carlini and Wagner, 2017c; Chen et al., 2018, 2020; Madry et al., 2018).
2. In gray-box attacks, the attacker does not have access to the internals of the model and might not even know the exact training set, although some general intuition about the design of the system and the type of training data needs to be present, as pointed out by Biggio and Roli (2018). In this case, the attacker trains a so-called surrogate model using data whose distribution is similar to the original training data and, if applicable, queries to the model under attack (Papernot et al., 2016b). If the training was successful, the surrogate model approximates the victim model sufficiently well to proceed to the next step. The attacker then creates an attack based on the surrogate model, which is likely to still perform well when applied to the targeted model, even if the model classes differ. This property of adversarial examples, which is very beneficial for attackers, has been termed transferability (Papernot et al., 2016a).

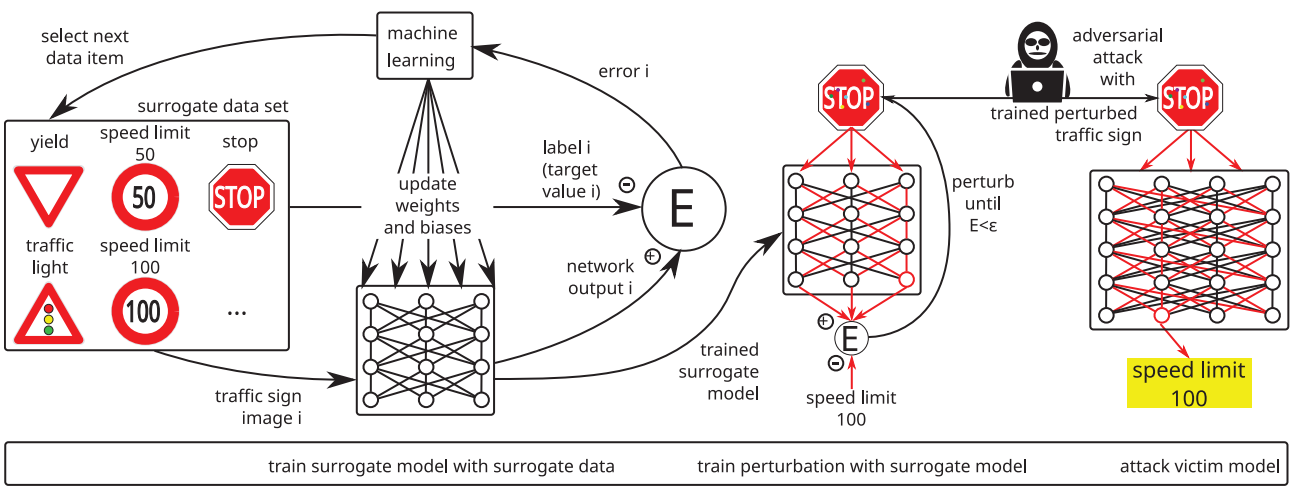

FIGURE 5 | Adversarial attacks may be conducted without white-box access to the victim model: First, a surrogate model is trained using a surrogate data set. Labels for this data set might optionally be obtained via queries to the victim model. Subsequently, the trained surrogate model is used to generate adversarial input examples. In many cases, these adversarial examples may then be used successfully for attacking the victim model.

Adversarial attacks usually choose the resulting data points to be close to the original ones in some metric, e.g., the Euclidean distance. This can make them indistinguishable from the original data points for human perception and thus impossible to detect by a human observer. However, some researchers have raised the question whether this restriction is really necessary and have argued that in many applications it may not be (Gilmer et al., 2018; Yakura et al., 2020). This applies in particular to applications where human inspection of data is highly unlikely and even blatant perturbations might well go unnoticed, as e.g., in the analysis of network traffic.

In most academic publications, creating and deploying adversarial attacks is a completely digital procedure. For situated systems acting in the sensory-motor loop, such as autonomous cars, this approach may serve as a starting point for investigating adversarial attacks but generally misses out on crucial aspects of physical instantiations of these attacks: First, it is impossible to foresee and correctly simulate all possible boundary conditions as e.g., viewing angles, sensor pollution and temperature. Second, sufficiently realistic simulations of the interaction effects between system modules and environment are hard to carry out. Third, this likewise applies to simulating individual characteristics of hardware components that influence the behavior of these components. This means the required effort for generating physical adversarial attacks that perform well is much larger as compared to their digital copies. For this reason, such attacks are less well-studied, but several publications have shown they can still work, in particular if attacks are optimized for high robustness to typically occurring transformations (e.g., rotation and translation in images) (Sharif et al., 2016; Brown et al., 2017; Evtimov et al., 2017; Eykholt et al., 2017; Athalye et al., 2018b; Song et al., 2018).

3. KEY FACTORS UNDERLYING AI-SPECIFIC VULNERABILITIES

As described in section 2, AI systems can be attacked on different levels. Whereas many of the vulnerabilities are just variants of more general problems in IT security, which affect not only AI systems, but also other IT solutions, two types of attacks are specific to AI, i.e., poisoning attacks and adversarial examples (also known as evasion attacks). This section aims to give a general intuition of the fundamental properties specific to AI which enable and facilitate these attacks, and to outline some general strategies for coping with them.

3.1. Huge Input and State Spaces and Approximate Decision Boundaries

Complex AI models contain many millions of parameters (weights and biases), which are updated during training in order to approximate a function for solving the problem at hand. As a result, the number of possible combinations of parameters is enormous and decision boundaries between input data where the models' outputs differ can only be approximate (Hornik et al., 1989; Blackmore et al., 2006; Montúfar et al., 2014) (cf. **Table 1**).

TABLE 1 | The size of the input and state spaces of commonly used architectures in the field of object recognition (LeNet-5, VGG-16, ResNet-152) and natural language processing (BERT) is extremely large.

Model	Number of distinct possible inputs	Input size (in bit)	Output size (in bit)	Number of parameters	Number of layers
LeNet-5 (LeCun et al., 1998)	2^{6272}	$28 \cdot 28 \cdot 8 = 6272$	$10 \cdot 32$	$\approx 60\,\text{K}$	7
VGG-16 (Simonyan and Zisserman, 2015)	$2^{1204224}$	$224 \cdot 224 \cdot 3 \cdot 8 = 1204224$	$1000 \cdot 32$	$\approx 135\,\text{M}$	16
ResNet-152 (He et al., 2016)	$2^{1204224}$	$224 \cdot 224 \cdot 3 \cdot 8 = 1204224$	$1000 \cdot 32$	$\approx 60\,\text{M}$	152
BERT (Devlin et al., 2019)	$\leq 2^{7680}$	$\leq 512 \cdot 15 = 7680$	$\leq 512 \cdot 1000 \cdot 32$	$\approx 345\,\text{M}$	24

Besides, due to the models' non-linearity small perturbations in input values may result in huge differences in the output (Pasemann, 2002; Goodfellow et al., 2015; Li, 2018).

In general, AI models are trained on the natural distribution of the data considered in the specific problem (e.g., the distribution of traffic sign images). This distribution, however, lies on a very low-dimensional manifold as compared to the complete input space (e.g., all possible images of the same resolution) (Tanay and Griffin, 2016; Balda et al., 2020), which is sometimes referred to as the "curse of dimensionality." **Table 1** shows that the size of the input space for some common tasks is extremely large. Even rather simple and academic AI models as e.g., LeNet-5 for handwritten digit recognition have a huge input space. As a consequence, most possible inputs are never considered during training.

On the one hand, this creates a safety risk if the model is exposed to benign inputs which sufficiently differ from those seen during training, such that the model is unable to generalize to these new inputs (Novak et al., 2018; Jakubovitz et al., 2019). The probability of this happening depends on many factors, including the model, the algorithm used and especially the quality of the training data (Chung et al., 2018; Zahavy et al., 2018).

On the other hand, what is much more worrying, inputs which reliably cause malfunctioning for a model under attack, i.e., adversarial examples, can be computed efficiently and in a targeted way (Athalye et al., 2018b; Yousefzadeh and O'Leary, 2019; Chen et al., 2020). Although much work has been invested in designing defenses since adversarial examples first surfaced in deep learning, as of now, no general defense method is known which can reliably withstand adaptive attackers (Carlini and Wagner, 2017a; Athalye et al., 2018a). That is, defenses may work if information about their mode of operation is kept secret from an attacker (Song et al., 2019). As soon as an attacker gains this information, which should in most cases be considered possible following Kerckhoffs's principle, he is able to overcome them.

Besides the arms race in practical attacks and defenses, adversarial attacks have also sparked interest from a theoretical perspective (Goodfellow et al., 2015; Tanay and Griffin, 2016;

Biggio and Roli, 2018; Khoury and Hadfield-Menell, 2018; Madry et al., 2018; Ilyas et al., 2019; Balda et al., 2020). Several publications deal with their essential characteristics. As pointed out by Biggio and Roli (2018), adversarial examples commonly lie in areas of negligible probability, blind spots where the model is unsure about its predictions. Furthermore, they arise by adding highly non-random noise to legitimate samples, thus violating the implicit assumption of statistical noise that is made during training. Khoury and Hadfield-Menell (2018) relates adversarial examples to the high dimension of the input space and the curse of dimensionality, which allows constructing adversarial examples in many directions off the manifold of proper input data. In Ilyas et al. (2019), the existence of adversarial examples is ascribed to so-called non-robust features in the training data, which would also provide an explanation for their transferability property. By practical experiments (Madry et al., 2018) demonstrate defenses from the point of view of robust optimization that show comparatively high robustness against strong adversarial attacks. Additionally and in contrast to most other publications, theses defenses provide some theoretical guarantee against a whole range of both static and adaptive attacks.

Figure 6 illustrates the problem of adversarial examples and its root cause and presents an analogy from human psychophysics. Decision-making in humans (Loftus, 2005) as well as in AI systems (Jakubovitz et al., 2019) is error-prone since theoretically ideal boundaries for decision-making (task decision boundaries) are in practice instantiated by approximations (model decision boundaries). Models are trained using data (AI and humans) and evolutionary processes (humans). In the trained model, small changes in either sensory input or other boundary conditions (e.g., internal state) may lead to state changes whereby decision boundaries are crossed in state space, i.e., small changes in input (e.g., sensory noise) may lead to large output changes (here a different output class). Model and task decision, therefore, may not always match. Adversarial examples are found in those regions in input space where task and model decision boundaries differ, as depicted in **Figure 6**:

- Part A shows an example for human perception of ambiguous images, namely the so-called Necker cube: sensory input (image, viewpoint, lightening, …), internal states (genetics, previous experience, alertness, mood, …) and chance (e.g., sensory noise) determine in which of two possible ways the Necker cube is perceived: (top) either the square on the left/top side or the square on the right/bottom side is perceived as the front surface of the cube, and this perception may spontaneously switch from one to the other (bistability). Besides internal human states that influence which of the two perceptions is more likely to occur (Ward and Scholl, 2015), the input image may be slightly manipulated such that either the left/top square (left) or the right/bottom square (right) is perceived as the front surface of the cube.
- Part B shows how all these effects are also observed in AI systems. This figure illustrates adversarial examples for a simplified two-dimensional projection of an input space with three decision boundaries forming the model decision boundary of class A (yellow) modeling the task decision boundary (blue): small modifications can shift (red arrows) input data from one model decision class to another, with (example on boundary 2 on the left) and without (example on boundary 3 on the right) changing the task decision class. Most data are far enough from the model decision boundaries to exhibit a certain amount of robustness (example on boundary 1 on the bottom). It is important to note that this illustration, depicting a two-dimensional projection of input space, does not reflect realistic systems with high-dimensional input space. In those systems, adversarial examples may almost always be found within a small distance from the point of departure (Szegedy et al., 2014; Goodfellow et al., 2015; Khoury and Hadfield-Menell, 2018). These adversarial examples rarely occur by pure chance but attackers may efficiently search for them.

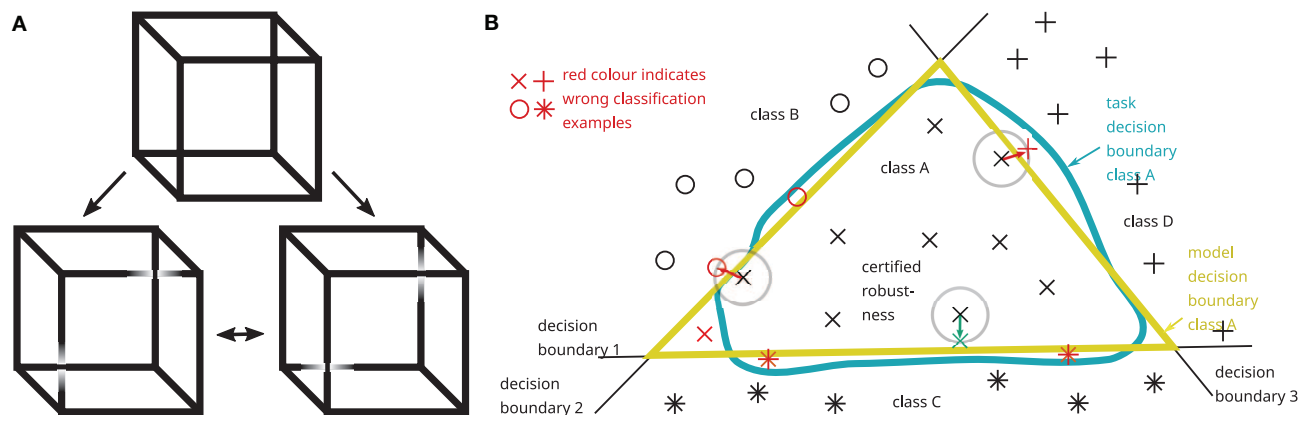

FIGURE 6 | Error-prone decision-making in humans **(A)** and AI systems **(B)** as exemplified by the Necker cube as an example of an ambiguous image **(A)** and a schematic depiction of adversarial examples in a 2D-projection of state space **(B)**. Task and model decision boundaries do not perfectly match and small changes in input may result in large changes in output. More details are given in the main text.

3.2. Black-Box Property and Lack of Interpretability

A major drawback of complex AI models like deep neural networks is their shortcoming in terms of interpretability and explainability (Rudin, 2019). Traditional computer programs solving a task are comprehensible and transparent at least to sufficiently knowledgeable programmers. Due to their huge parameter space as discussed in subsection 3.1, complex AI systems do not possess this property. In their case, a programmer can still understand the boundary conditions and the approach to the problem; however, it is infeasible for a human to directly convert the internal representation of a deep neural network to terms allowing him to understand how it operates. This is very dangerous from the perspective of IT security, since it means attacks can essentially only be detected from incorrect behavior of the model (which may in itself be hard to notice), but not by inspecting the model itself. In particular, after training is completed, the model's lack of transparency makes it very hard to detect poisoning and backdooring attacks on the training data. For this reason, such attacks should be addressed and mitigated by thorough documentation of the training and evaluation process and by protecting the integrity of intermediate results or alternatively by using training and test data that have been certified by a trustworthy party.

A straightforward solution to the black-box property of complex AI models would be to use a model which is inherently easier to interpret for a human, e.g., a decision tree or a rule list (Molnar, 2020). When considering applications based on tabular data, for instance in health care or finance, one finds that decision trees or rule lists even perform better than complex cAI models in most cases (Angelino et al., 2018; Rudin, 2019; Lundberg et al., 2020), besides exhibiting superior interpretability. However, in applications from computer vision, which are the focus of this paper, or speech recognition, sAI models cannot compete with complex models like deep neural networks, which are unfortunately very hard to interpret. For these applications, there is hence a trade-off between model interpretability and performance. A general rule of thumb for tackling the issue of interpretability would still consist in using the least complex model which is capable of solving a given problem sufficiently well. Another approach for gaining more insight into the operation of a black-box model is to use XAI methods that essentially aim to provide their users with a human-interpretable version of the model's internal representation. This is an active field of research, where many methods have been proposed in recent years (Gilpin et al., 2018; Samek et al., 2019; Molnar, 2020). Yet another approach is to use—where available—AI-systems which have been mathematically proven to be robust against attacks under the boundary conditions that apply for the specific use case (Huang et al., 2017; Katz et al., 2017; Gehr et al., 2018; Wong et al., 2018; Wong and Kolter, 2018; Singh et al., 2019). For more details, the reader is referred to subsection 4.3.

3.3. Dependence of Performance and Security on Training Data

The accuracy and robustness of an AI model is highly dependent on the quality and quantity of the training data (Zhu et al., 2016; Sun et al., 2017; Chung et al., 2018). In particular, the model can only achieve high overall performance if the training data are unbiased (Juba and Le, 2019; Kim et al., 2019). Despite their name, AI models currently used are not "intelligent," and hence they can only learn correlations from data but cannot by themselves differentiate spurious correlations from true causalities.

For economic reasons, it is quite common to outsource part of the supply chain of an AI model and obtain data and models for further training from sources which may not be trustworthy (cf. **Figure 7**). On the one hand, for lack of computational resources and professional expertise, developers of AI systems often use pre-trained networks provided by large international companies or even perform the whole training process in an environment not under their control. On the other hand, due to the efforts required in terms of funds and personnel for collecting training data from scratch as well as due to local data protection laws (e.g., the GDPR in the European Union), they often obtain whole data sets in other countries. This does not only apply to data sets containing real data, but also to data which are synthetically created (Gohorbani et al., 2019) in order to save costs. Besides synthetic data created from scratch, this especially concerns data obtained by augmenting an original data set, e.g., using transformations under which the model's output should remain invariant.

Both these facts are problematic in terms of IT security, since they carry the risk of dealing with biased or poor-quality data and of falling prey to poisoning attacks (cf. section 2), which are very hard to detect afterwards. The safest way to avoid these issues is not to rely on data or models furnished by other parties. If this is infeasible, at least a thorough documentation and cryptographic mechanisms for protecting the integrity and authenticity of such data and models should be applied throughout their whole supply chain (cf. subsection 4.2).

4. MITIGATION OF VULNERABILITIES OF AI SYSTEMS

4.1. Assessment of Attacks

A necessary condition for properly reasoning about attacks is to classify them using high-level criteria. The result of this classification will facilitate a discussion about defenses which are feasible and necessary. Such a classification is often referred to as a threat model or attacker model (Papernot et al., 2016d; Biggio and Roli, 2018).

An important criterion to consider is the **goal** of the attack. First, one needs to establish which security goal is affected. As already noted in section 1, attackers can target either integrity (by having the system make wrong predictions on specific input data), availability (by hindering legitimate users from properly using the system) or confidentiality (by extracting information without proper authorization). Besides, the scope of the attack may vary. An attacker may mount a targeted attack, which affects only certain data samples, or an indiscriminate one. In addition, the attacker may induce a specific or a general error. When considering AI classifiers, for instance, a specific error means that a sample is labeled as belonging to a target class of the

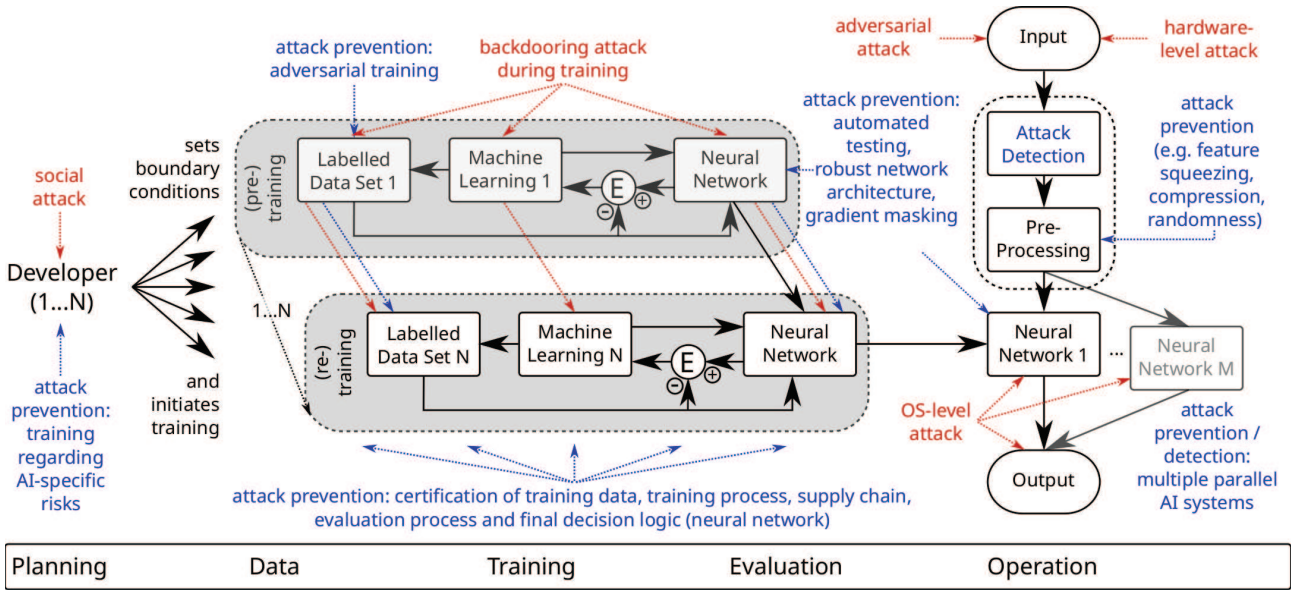

FIGURE 7 | Summary of possible attacks (red) on AI systems and defenses (blue) specific to AI systems depicted along the AI life cycle. Defenses not specific to AI systems, e.g., background checks of developers, hardware access control etc. are not shown here and should be adopted from classical IT security. Multiple AI training sessions with different data sets indicate the risk associated with pre-trained networks and externally acquired data.

attacker's choosing, whereas a general error only requires any incorrect label to be assigned to the sample. Furthermore, the ultimate objective of the attack must be considered. For example, this can be the unauthorized use of a passport (when attacking biometric authentication) or recognizing a wrong traffic sign (in autonomous driving applications). In order to properly assess the attack, it is necessary to measure its real-world impact. For lack of more precise metrics commonly agreed upon, as a first step one might resort to a general scale assessing the attack as having low, medium or high impact.

The **knowledge** needed to carry out an attack is another criterion to consider. As described in subsection 2.3, an attacker has full knowledge of the model and the data sets in the white-box case. In this scenario, the attacker is strongest, and an analysis assuming white-box access thus gives a worst-case estimate for security. As noted in Carlini et al. (2019), when performing such a white-box analysis, for the correct assessment of the vulnerabilities it is of paramount importance to use additional tests for checking whether the white-box attacks in question have been applied correctly, since mistakes in applying them have been observed many times and might yield wrong results.

In the case of a gray-box attack, conducting an analysis requires making precise assumptions on which information is assumed to be known to the attacker, and which is secret. Carlini et al. (2019) suggests that, in the same way as with cryptographic schemes, as little information as possible should be assumed to be secret when assessing the security of an AI system. For instance, the type of defense used in the system should be assumed to be known to the attacker.

The third criterion to be taken into account is the **efficiency** of the attack, which influences the capabilities and resources an attacker requires. We assume the cost of a successful attack to be the most important proxy metric from the attacker's point of view. This helps in judging whether an attack is realistic in a real-world setting. If an attacker is able to achieve his objective using a completely different attack which does not directly target the AI system and costs less, it seems highly probable a reasonable attacker will prefer this alternative (cf. the concise discussion in Gilmer et al., 2018). Possible alternatives may change over time though, and if effective defenses against them are put into place, the attacker will update his calculation and may likely turn to attack forms he originally disregarded, e.g., attacks on the AI system as discussed in this paper.

The cost of a successful attack is influenced by several factors. First, the general effort and scope of a successful attack have a direct influence. For instance, the fact whether manipulating only a few samples is sufficient for mounting a successful poisoning attack or whether many samples need to be affected can have a strong impact on the required cost, especially when taking into account additional measures for avoiding detection. Second, the degree of automation of the attack determines how much manual work and manpower is required. Third, the fact whether an attack requires physical presence or can be performed remotely is likewise important. For instance, an attack which allows only a low degree of automation and requires physical presence is much more costly to mount and especially to scale. Fourth, attacking in a real-world setting adds further complexity and might hence be more expensive than an attack in a laboratory setting, where all the side conditions are under control.

A fourth important criterion is the **availability of mitigations**, which may significantly increase the attacker's cost. However, mitigations must in turn be judged by the effort they require for the defender, their efficiency and effectiveness. In particular, non-adaptive defense mechanisms may provide a false sense of

security, since an attacker who gains sufficient knowledge can bypass them by modifying his attack appropriately. This is a serious problem pointed out in many publications (cf. Athalye et al., 2018a; Gilmer et al., 2018). As a rule, defense mechanisms should therefore respect Kerckhoffs's principle and must not rely on security by obscurity.

4.2. General Measures

A lot of research has been done on how to mitigate attacks on AI systems (Bethge, 2019; Carlini et al., 2019; Madry et al., 2019). However, almost all the literature so far focuses on mitigations inside the AI systems, neglecting other possible defensive measures, and does not take into account the complete AI life cycle when assessing attacks. Furthermore, although certain defenses like some variants of adversarial training (Tramèr et al., 2018; Salman et al., 2019) can increase robustness against special threat models, there is, as of now, no general defense mechanism which is applicable against all types of attacks. A significant problem of most published defenses consists in their lack of resilience against adaptive attackers (Carlini and Wagner, 2017a,b; Athalye et al., 2018a). As already stated, the defense mechanisms used should be assumed to be public. The resistance of a defense against attackers who adapt to it is hence extremely important. In this section, we argue that a broader array of measures need to be combined for increasing security, especially if one intends to certify the safe and secure operation of an AI system, as seems necessary in high-risk applications like autonomous driving. An overview of defenses and attacks is presented in **Figure 7**.

There is no compelling reason to focus solely on defending the AI system itself without taking into account additional measures which can hamper attacks by changing side conditions. This observation does not by any means imply that defenses inside the AI system are unimportant or not necessary but instead emphasizes that they constitute a last line of defense, which should be reinforced by other mechanisms.

Legal measures are most general. They cannot by themselves prevent attacks, but may serve as a deterrent to a certain extent, if properly implemented and enforced. Legal measures may include the adoption of new laws and regulation or specifying how existing laws apply to AI applications.

Organizational measures can influence the side conditions, making them less advantageous for an attacker. For instance, in biometric authentication systems at border control, a human monitoring several systems at once and checking for unusual behavior or appearance may prevent attacks which can fool the AI system but are obvious to a human observer or can easily be detected by him if he is properly trained in advance. Restricting access to the development and training of AI systems for sensitive use cases to personnel which has undergone a background check is another example of an organizational measure. Yet another example is properly checking the identity of key holders when using a public key infrastructure (PKI) for protecting the authenticity of data.

Technical measures outside the AI system can be applied to increase IT security. The whole supply chain of collecting and preprocessing data, aggregating and transmitting data sets, pre-training models which are used as a basis for further training, and the training procedure itself can be documented and secured using classic cryptographic schemes like hash functions and digital signatures to ensure integrity and authenticity (this ultimately requires a PKI), preventing tampering in the process and allowing reproducing results and tracing back problems (Berghoff, 2020). Depending on the targeted level of security and traceability, the information covered may include all the training and test data, all AI models, all ML algorithms, a detailed logging of the development process (e.g., hyperparameters set by the developer, pseudo-random seeds, intermediate results) and comments of the developers concisely explaining and justifying each step in the development process. If the source of the data used is itself trusted, such documentation and cryptographic protection can later be validated to prove (with high probability) that no data poisoning attacks have been carried out, provided the validating party gets access to at least a sample of the original data and can check the correctness of intermediate results. As a further external technical measure, the AI system can be enhanced by using additional information from other sources. For example, in biometric authentication, biometric fakes can be detected using additional sensors (Marcel et al., 2019).

In a somewhat similar vein, the **redundant operation of multiple AI systems** running in parallel may serve to increase robustness to attacks, while at the same time increasing the robustness on benign data not seen during training. These systems can be deployed in conjunction with each other and compare and verify each other's results, thus increasing redundancy. The final result might be derived by a simple majority vote (cf. **Figure 7**). Other strategies are conceivable though. For instance, in safety-critical environments an alarm could be triggered in case the final decision is not unanimous and, if applicable, the system could be transferred to a safe fall-back state pending closer inspection. Increasing the redundancy of a technical system is a well-known approach for reducing the probability of undesired behavior, whether due to benign reasons or induced by an attacker. However, the transferability property of adversarial examples (cf. subsection 2.5, Papernot et al., 2016a) implies that attacks may continue to work even in the presence of redundancy, although their probability of success should at least slightly diminish. As a result, when using redundancy, one should aim to use conceptually different models and train them using different training sets that all stem from the data distribution representing the problem at hand, but have been sampled independently or at least exhibit only small intersections. While this does not in principle resolve the challenges posed by transferability, our intuition is that it should help to further decrease an attacker's probability of success.

4.3. AI-Specific Measures

On the AI level, several measures can likewise be combined and used in conjunction with the general countermeasures presented above. First and foremost, appropriate state-of-the-art defenses from the literature can be implemented according to their security benefits and the application scenario. One common approach for thwarting adversarial attacks is to make

use of input compression (Dziugaite et al., 2016; Das et al., 2017), which removes high-frequency components from input data that are typical for adversarial examples. More prominent still is a technique called **adversarial training**, which consists in pre-computing adversarial examples using standard attack algorithms and incorporating them into the training process of the model, thus making it more robust and, in an ideal setting, immune to such attacks. State-of-the-art adversarial training methods may be identified using (Madry et al., 2018, 2019; Bethge, 2019). In general, when dealing with countermeasures against adversarial attacks, it is important to keep in mind that many proposed defenses have been broken in the past (Carlini and Wagner, 2017b; Athalye et al., 2018a), and that even the best defenses available and combinations thereof Carlini and Wagner (2017a) may not fully mitigate the problem of adversarial attacks.

In terms of **defenses against backdoor poisoning attacks** only a few promising proposals have been published in recent years (Tran et al., 2018; Chen et al., 2019; Wang et al., 2019). Their main idea lies in the creation of a method which proposes possibly malicious data samples of the training set for manual examination. Those methods use the fact that a neural network trained on such a compromised data set learns the false classification of backdoored samples as exceptions, which can be detected from the internal representation of the network. It needs to be kept in mind though that those defenses do not provide any formal guarantees and might be circumvented by an adaptive adversary.

As a first step, instead of preventing AI-specific attacks altogether, **reliably detecting** them might be a somewhat easier and hence more realistic task (Carlini and Wagner, 2017a). In case an attack is detected, the system might yield a special output corresponding to this situation, trigger an alarm and forward the apparently malicious input to another IT system or a human in the loop for further inspection. It depends on the application in question whether this approach is feasible. For instance, asking a human for feedback is incompatible by definition with fully autonomous driving at SAE level 5 (ORAD Committee, 2018).

A different approach lies in using methods from the area of **explainable AI (XAI)** to better understand the underlying reasons for the decisions which an AI system takes (cf. **Figure 8**). At the least, such methods may help to detect potential vulnerabilities and to develop more targeted defenses. One example is provided by Lapuschkin et al. (2019), which suggests a more diligent preprocessing of data for preventing the AI system from learning spurious correlations, which can easily be attacked. In principle, one can also hope that XAI methods will allow reasoning about the correctness of AI decisions under a certain range of circumstances. The field of XAI as focused on (deep) neural networks is quite young, and research has only started around 2015, although the general question of explaining decisions of AI systems dates back about 50 years (Samek et al.,

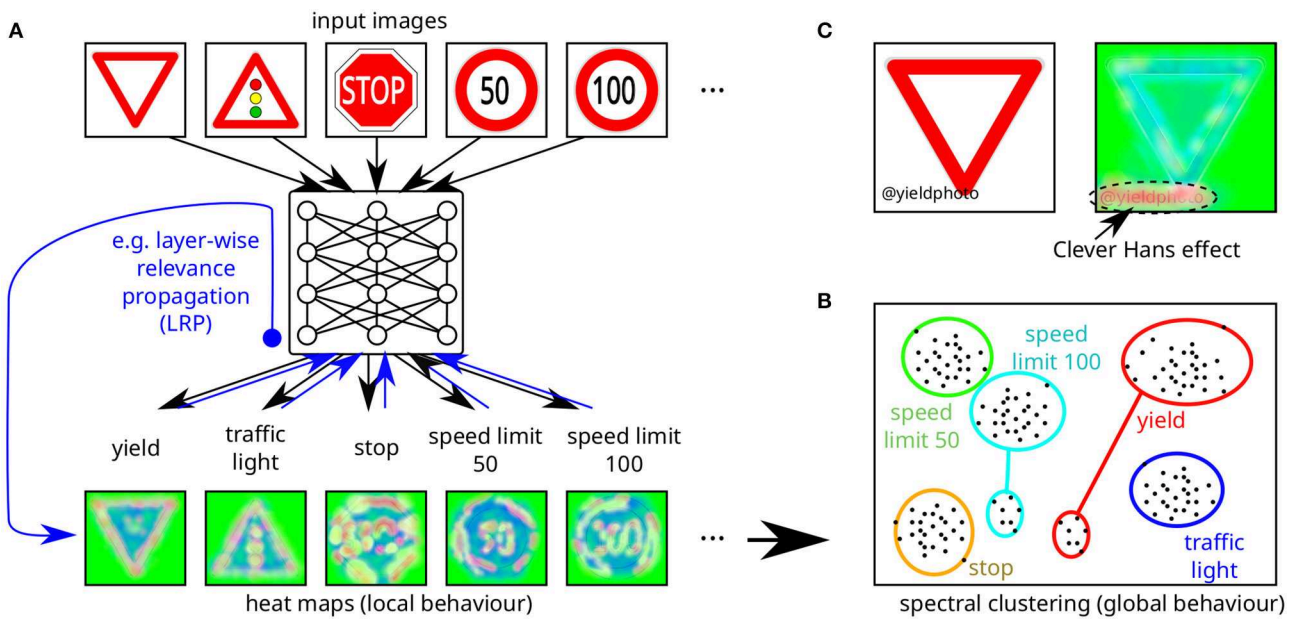

FIGURE 8 | Schematic illustration of the application of explainable AI (XAI) methods to deduce **(A)** local and **(B)** global model behavior of an AI system. **(A)** shows how heat maps are generated after labels were obtained for a specific input image, in this case using LRP (Samek et al., 2016), which assigns each input pixel a relative contribution to the output decision (green colors indicate lowest relevance, red colors highest relevance). **(B)** illustrates how many local model behavior explanations are combined to explain global model behavior, in this case using spectral analysis (cf. Lapuschkin et al., 2019). Here multiple topographically distinct clusters for individual labels shown in a 2D projection of input space indicate some kind of problem: the small cluster for *speed limit 100* represents the backdooring attack using modified stop signs (cf. **Figure 4**) and the small cluster for the yield sign represents the Clever Hans effect illustrated in detail in **(C)**, where specific image tags (here "@yieldphoto") correlate with specific input classes (here the yield sign) and the AI system focuses on these spurious correlations instead of causal correlations. Upon swapping the input data set (not containing any more spurious correlations of this kind), the AI model might show erroneous behavior.

2019, pp. 41–49). So far, it seems doubtful there will be a single method which will fit in every case. Rather, different conditions will require different approaches. On the one hand, the high-level use case has a strong impact on the applicable methods: When making predictions from structured data, probabilistic methods are considered promising (Molnar, 2020), whereas applications from computer vision rely on more advanced methods like layer-wise relevance propagation (LRP) (Bach et al., 2015; Samek et al., 2016; Montavon et al., 2017; Lapuschkin et al., 2019). On the other hand, some methods provide global explanations, while others explain individual (local) decisions. It should be noted that by using principles similar to adversarial examples, current XAI methods can themselves be efficiently attacked. Such attacks may either be performed as an enhancement to adversarial examples targeting the model (Zhang et al., 2018) or by completely altering the explanations provided while leaving model output unchanged (Dombrowski et al., 2019). Based on theoretical and practical observations, both Zhang et al. (2018) and Dombrowski et al. (2019) suggest countermeasures for thwarting the respective attacks.

A third line of research linked to both other approaches is concerned with **verifying and proving** the safety and security of AI systems. Owing to the much greater complexity of this problem, results in this area, especially practically usable ones, are scarce (Huang et al., 2017; Katz et al., 2017; Gehr et al., 2018; Wong et al., 2018; Wong and Kolter, 2018; Singh et al., 2019). A general idea for harnessing the potential of XAI and verification methods may be applied, provided one manages to make these methods work on moderately small models. In this case, it might be possible to **modularize** the AI system in question so that core functions are mapped to small AI models (Mascharka et al., 2018), which can then be checked and verified. From the perspective of data protection, this approach has the additional advantage that the use of specific data may be restricted to the training of specific modules. In contrast to monolithic models, this allows unlearning specific data by replacing the corresponding modules (Bourtoule et al., 2019).

5. CONCLUSION AND OUTLOOK

The life cycle of AI systems can give rise to malfunctions and is susceptible to targeted attacks at different levels. When facing naturally occurring circumstances and benign failures, i.e., in terms of safety, well-trained AI systems display robust performance in many cases. In practice, they may still show highly undesired behavior, as exemplified by several incidents involving Tesla cars (Wikipedia Contributors, 2020). The main problem in this respect is insufficient training data. The black-box property of the systems aggravates this issue, in particular when it comes to gaining user trust or establishing guarantees on correct behavior of the system under a range of circumstances.

The situation is much more problematic though when it comes to the robustness to attacks exhibited by the systems. Whereas a lot of attacks can be combated using traditional measures of IT security, the AI-specific vulnerabilities to poisoning and evasion attacks can have grave consequences and do not yet admit reliable mitigations. Considerable effort has been put into researching AI-specific vulnerabilities, yet more is needed, since defenses still need to become more resilient to attackers if they are to be used in safety-critical applications. In order to achieve this goal, it seems furthermore indispensable to combine defense measures at different levels and not only focus on the internals of the AI system.

Additional open questions concern the area of XAI, which is quite recent with respect to complex AI systems. The capabilities and limitations of existing methods need to be better understood, and reliable and sensible benchmarks need to be constructed to compare them (Osman et al., 2020). The topic of formal verification of the functionality of an AI system is an important enhancement that should further be studied. A general approach for obtaining better results from XAI and verification methods is to reduce complexity in the models to be analyzed. We argue that for safety-critical applications the size of AI systems used for certain tasks should be minimized subject to the desired performance. If possible, one might also envision using a modular system containing small modules, which lend themselves more easily to analysis. A thorough evaluation using suitable metrics should be considered a prerequisite for the deployment of any IT system and, therefore, of any AI system.

Thinking ahead, the issue of AI systems which are continuously being trained using fresh data (called continual learning, Parisi et al., 2019) also needs to be considered. This approach poses at least two difficulties as compared to the more static life cycle considered in this article. On the one hand, depending on how the training is done, an attacker might have a much better opportunity for poisoning training data. On the other hand, results on robustness, resilience to attacks or correctness guarantees will only be valid for a certain version of a model and may quickly become obsolete. This might be tackled by using regular checkpoints and repeating the countermeasures and evaluations, at potentially high costs.

Considering the current state of the art in the field of XAI and verification, it is unclear whether it will ever be possible to formally certify the correct operation of an arbitrary AI system and construct a system which is immune to the AI-specific attacks presented in this article. It is conceivable that both certification results and defenses will continue to only yield probabilistic guarantees on the overall robustness and correct operation of the system. If this assumption turns out true for the foreseeable future, its implications for safety-critical applications of AI systems need to be carefully considered and discussed without bias. For instance, it is important to discuss which level of residual risk, if any, one might be willing to accept in return for possible benefits of AI over traditional solutions, and in what way the conformance to a risk level might be tested and confirmed. For instance, humans are required to pass a driving test before obtaining their driver's license and being allowed to drive on their own. While a human having passed a driving test is not guaranteed to always respect the traffic rules, to behave correctly and to not cause any harm to other traffic participants, the test enforces a certain standard. In a similar vein, one might imagine a special test to be passed by an AI system for obtaining regulatory approval. In these cases the risks and

benefits of using an AI system and the boundary conditions for which the risk assessment is valid should be made transparent to the user. However, the use of any IT system that cannot be guaranteed to achieve the acceptable risk level as outlined above could in extreme cases be banned for particularly safety-critical applications. Specifically, such a ban could apply to pure AI systems, if they fail to achieve such guarantees.

AUTHOR CONTRIBUTIONS

CB, MN, and AT conceived the article and surveyed relevant publications. CB wrote the original draft of the manuscript, with some help by MN. AT designed and created all the figures and tables, reviewed and edited the manuscript, with help by CB. All authors contributed to the article and approved the submitted version.

FUNDING

The article was written as part of the authors' employment at Federal Office for Information Security, Bonn, Germany. The authors did not receive any other funding.

ACKNOWLEDGMENTS

We would like to thank Ute Gebhardt, Rainer Plaga, Markus Ullmann, and Wojciech Samek for carefully proofreading earlier versions of this document and providing valuable suggestions for improvement. Further we would like to thank Frank Pasemann, Petar Tsankov, Vasilios Danos, and the VdTÜV-BSI AI work group for fruitful discussions. We would also like to thank the reviewers for their helpful comments. This manuscript has been released as a preprint at arXiv:2003.08837.

REFERENCES

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., and Rudin, C. (2018). Learning certifiably optimal rule lists for categorical data. *J. Mach. Learn. Res.* 18, 1–78.

Athalye, A., Carlini, N., and Wagner, D. (2018a). "Obfuscated gradients give a false sense of security: circumventing Defenses to adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Volume 80 of Proceedings of Machine Learning Research*, eds J. G. Dy and A. Krause (Stockholm: PMLR), 274–283.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. (2018b). "Synthesizing robust and adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Volume 80 of Proceedings of Machine Learning Research*, eds J. G. Dy and A. Krause (Stockholm: PMLR), 284–293.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* 10:e0130140. doi: 10.1371/journal.pone.0130140

Balda, E. R., Behboodi, A., and Mathar, R. (2020). *Adversarial Examples in Deep Neural Networks: An Overview, Volume 865 of Studies in Computational Intelligence* (Cham: Springer), 31–65.

Barreno, M., Nelson, B., Sears, R., Joseph, A. D., and Tygar, J. D. (2006). "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006*, eds F. C. Lin, D. T. Lee, B. S. Paul Lin, S. Shieh, and S. Jajodia (Taipei: ACM), 16–25.

Berghoff, C. (2020). Protecting the integrity of the training procedure of neural networks. *arXiv:2005.06928*.

Bethge, A. G. (2019). *Robust Vision Benchmark*. Available online at: https://robust.vision (accessed March 3, 2020).

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., et al. (2013) "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases*, eds H. Blockeel, K. Kersting, S. Nijssen, and F. Železný (Berlin; Heidelberg: Springer), 387–402.

Biggio, B., Nelson, B., and Laskov, P. (2012). "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, eds J. Langford and J. Pineau (Omnipress), 1807–1814.

Biggio, B., and Roli, F. (2018). Wild patterns: ten years after the rise of adversarial machine learning. *Pattern Recogn.* 84, 317–331. doi: 10.1016/j.patcog.2018.07.023

Blackmore, K. L., Williamson, R. C., and Mareels, I. M. Y. (2006). Decision region approximation by polynomials or neural networks. *IEEE Trans. Inform. Theory* 43, 903–907. doi: 10.1109/18.568700

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C., Jia, H., Travers, A., Zhang, B., et al. (2019). Machine unlearning. *arXiv* abs/1912.03817.

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. *arXiv* abs/1712.09665.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., et al. (2019). On evaluating adversarial robustness. *arXiv* abs/1902.06705.

Carlini, N., and Wagner, D. (2017a). "Adversarial examples are not easily detected: bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec '17)*, eds B. M. Thuraisingham, B. Biggio, D. M. Freeman, B. Miller, and A. Sinha (New York, NY: Association for Computing Machinery), 3–14.

Carlini, N., and Wagner, D. (2017b). MagNet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv* abs/1711.08478.

Carlini, N., and Wagner, D. (2017c). "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP)* (San Jose, CA), 39–57.

Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., et al. (2019). "Detecting backdoor attacks on deep neural networks by activation clustering," in *Workshop on Artificial Intelligence Safety 2019 Co-located With the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Volume 2301 of CEUR Workshop Proceedings*, eds H. Espinoza, S. hÉigeartaigh, X. Huang, J. Hernández-Orallo, and M. Castillo-Effen (Honolulu, HI: CEUR-WS.org).

Chen, J., Zhou, D., Yi, J., and Gu, Q. (2020). "A Frank-Wolfe framework for efficient and effective adversarial attacks," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence 2020 (AAAI-20)* (New York, NY).

Chen, P. Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C. J. (2018). "EAD: elastic-net attacks to deep neural networks via adversarial examples," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, eds S. A. McIlraith and K. Q. Weinberger (New Orleans, LA: AAAI Press), 10–17.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. (2017). Targeted backdoor Attacks on deep learning systems using data poisoning. *arXiv* abs/1712.05526.

Chung, Y., Haas, P. J., Upfal, E., and Kraska, T. (2018). Unknown examples & machine learning model generalization. *arXiv* abs/1808.08294.

Clements, J., and Lao, Y. (2018). Hardware trojan attacks on neural networks. *arXiv* abs/1806.05768.

Dalvi, N. N., Domingos, P. M., Mausam, Sanghai, S. K., and Verma, D. (2004). "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel (Seattle, WA), 99–108.

Das, N., Shanbhogue, M., Chen, S. T., Hohman, F., Chen, L., Kounavis, M. E., et al. (2017). Keeping the bad guys out: protecting and vaccinating deep learning with JPEG compression. *arXiv* abs/1705.02900.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers), eds J. Burstein, C. Doran, and T. Solorio (Minneapolis, MN: Association for Computational Linguistics), 4171–4186.

Dombrowski, A. K., Alber, M., Anders, C. J., Ackermann, M., Müller, K. R., and Kessel, P. (2019). "Explanations can be manipulated and geometry is to blame," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, eds H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, (Vancouver, BC), 13567–13578.

Dziugaite, G. K., Ghahramani, Z., and Roy, D. M. (2016). A study of the effect of JPG compression on adversarial images. *arXiv* abs/1608.00853.

Eagleman, D. M. (2001). Visual illusions and neurobiology. *Nat. Rev. Neurosci.* 2, 920–926. doi: 10.1038/35104092

Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., et al. (2017). Robust physical-world attacks on machine learning models. *arXiv* abs/1707.08945.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Song, D., Kohno, T., et al. (2017). Note on attacking object detectors with adversarial stickers. *arXiv* abs/1712.08062.

Facebook. *PyTorch*. Available online at: https://pytorch.org (accessed March 17, 2020).

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. (2018). "AI2: safety and robustness certification of neural networks with abstract interpretation," in *IEEE Symposium on Security and Privacy (SP)* (San Francisco, CA), 3–18.

Gilmer, J., Adams, R. P., Goodfellow, I. J., Andersen, D., and Dahl, G. E. (2018). Motivating the rules of the game for adversarial example research. *arXiv* abs/1807.06732.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). "Explaining explanations: an overview of interpretability of machine learning," in *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018*, eds F. Bonchi, F. J. Provost, T. Eliassi-Rad, W. Wang, C. Cattuto, and R. Ghani (Turin: IEEE), 80–89.

Gohorbani, A., Natarajan, V., Coz, D. D., and Liu, Y. (2019). "DermGAN: synthetic generation of clinical skin images with pathology," in *Proceedings of Machine Learning for Health (ML4H) at NeurIPS 2019* (Vancouver, BC).

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*. Available online at: http://arxiv.org/abs/1412.6572

Google Brain. *TensorFlow*. Available online at: https://www.tensorflow.org (accessed March 17, 2020).

Gu, T., Dolan-Gavitt, B., and Garg, S. (2017). BadNets: identifying vulnerabilities in the machine learning model supply chain. *arXiv* abs/1708.06733.

Haykin, S. (1999). *Neural Networks, 2nd Edn.* Upper Saddle River, NJ: Prentice Hall.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016* (Las Vegas, NV: IEEE Computer Society), 770–778.

Hornik, K., Stinchcombe, M. B., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw.* 2, 359–366.

Huang, X., Kwiatkowska, M., Wang, S., and Wu, M. (2017). "Safety verification of deep neural networks," in *Computer Aided Verification–29th International Conference, CAV 2017, Proceedings, Part I, Volume 10426 of Lecture Notes in Computer Science*, eds R. Majumdar and V. Kuncak (Heidelberg: Springer), 3–29.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, eds H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett (Vancouver, BC, Canada), 125–136.

INRIA. *Scikit-Learn*. Available online at: https://scikit-learn.org/stable/ (accessed March 17, 2020).

Jakubovitz, D., Giryes, R., and Rodrigues, M. R. D. (2019). "Generalization error in deep learning," in *Compressed Sensing and Its Applications. Applied and Numerical Harmonic Analysis*, eds H. Boche, G. Caire, R. Calderbank, G. Kutyniok, R. Mathar, and P. Petersen (Cham: Birkhäuser). doi: 10.1007/978-3-319-73074-5_5

Ji, Y., Liu, Z., Hu, X., Wang, P., and Zhang, Y. (2019). Programmable neural network trojan for pre-trained feature extractor. *arXiv* abs/1901.07766.

Juba, B., and Le, H. S. (2019). "Precision-recall versus accuracy and the role of large data sets," in *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, Vol. 33 (Honolulu, HI).

Katz, G., Barrett, C. W., Dill, D. L., Julian, K., and Kochenderfer, M. J. (2017). "Reluplex: an efficient SMT solver for verifying deep neural networks," in *Computer Aided Verification–29th International Conference, CAV 2017, Proceedings, Part I, Volume 10426 of Lecture Notes in Computer Science*, eds R. Majumdar and V. Kuncak (Heidelberg: Springer), 97–117.

Khoury, M., and Hadfield-Menell, D. (2018). On the geometry of adversarial examples. *arXiv* abs/1811.00525.

Kim, B., Kim, H., Kim, K., Kim, S., and Kim, J. (2019). "Learning not to learn: training deep neural networks with biased data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA).

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K. R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nat. Commun.* 10, 1–8. doi: 10.1038/s41467-019-08987-4

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324.

Lederberg, J. (1987). "How DENDRAL was conceived and born," in *Proceedings of the ACM Conference on History of Medical Informatics*, ed B. I. Blum (Bethesda, MD: ACM), 5–19.

Li, H. (2018). Analysis on the nonlinear dynamics of deep neural networks: topological entropy and chaos. *arXiv* abs/1804.03987.

Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., and Leung, V. C. M. (2018). A survey on security threats and defensive techniques of machine learning: a data driven view. *IEEE Access* 6, 12103–12117. doi: 10.1109/ACCESS.2018.2805680

Liu, Y., Ma, S., Aafer, Y., Lee, W. C., Zhai, J., Wang, W., et al. (2018). "Trojaning attack on neural networks," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018* (San Diego, CA: The Internet Society).

Loftus, E. F. (2005). Planting misinformation in the human mind: a 30-year investigation of the malleability of memory. *Learn. Mem.* 12, 361–366. doi: 10.1101/lm.94705

Lowd, D., and Meek, C. (2005). "Adversarial learning," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds R. Grossman, R. J. Bayardo, and K. P. Bennett (Chicago, IL: ACM), 641–647.

Lundberg, S. M., Erion, G. G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., et al. (2020). Explainable AI for trees: from local explanations to global understanding. *Nat. Mach. Intell.* 2, 56–67. doi: 10.1038/s42256-019-0138-9

Madry, A., Athalye, A., Tsipras, D., and Engstrom, L. (2019). *RobustML*. Available online at: https://www.robust-ml.org/ (accessed March 17, 2020).

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). "Towards deep learning models resistant to adversarial attack," in *6th International Conference on Learning Representations* (Vancouver, BC). Available online at: http://arxiv.org/abs/1706.06083

Marcel, S., Nixon, M. S., and Fierrez, J. (Eds.). (2019). *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection*. Advances in Computer Vision and Pattern Recognition (Basel: Springer International Publishing).

Mascharka, D., Tran, P., Soklaski, R., and Majumdar, A. (2018). "Transparency by design: closing the gap between performance and interpretability in visual reasoning," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018* (Salt Lake City, UT: IEEE Computer Society), 4942–4950.

McCulloch, W., and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bull. Math. Biophys.* 5, 115–133.

Mei, S., and Zhu, X. (2015). "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, eds B. Bonet and S. Koenig (Austin, TX: AAAI Press), 2871–2877.

Molnar, C. (2020). *Interpretable Machine Learning–A Guide for Making Black Box Models Explainable*. Available online at: https://christophm.github.io/interpretable-ml-book/ (accessed March 17, 2020).

Montavon, G., Bach, S., Binder, A., Samek, W., and Müller, K. R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recogn.* 65, 211–222. doi: 10.1016/j.patcog.2016.11.008

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). "On the number of linear regions of deep neural networks," in *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2 (Montreal, QC), 2924–2932.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018). "Sensitivity and generalization in neural networks: an empirical study," in *International Conference on Learning Representations* (Vancouver, BC).

On-Road Automated Driving (ORAD) Committee (2018). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016_201806*. Technical report, SAE International.

Osman, A., Arras, L., and Samek, W. (2020). Towards ground truth evaluation of visual explanations. *arXiv* abs/2003.07258.

Papernot, N., McDaniel, P. D., and Goodfellow, I. J. (2016a). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv* abs/1605.07277.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. (2016b). Practical black-box attacks against deep learning systems using adversarial examples. *arXiv* abs/1602.02697.

Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016c). "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy, EuroS&P 2016* (Saarbrücken), 372–387.

Papernot, N., McDaniel, P. D., Sinha, A., and Wellman, M. P. (2016d). "SoK: security and privacy in machine learning," in *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018* (London: IEEE), 399–414.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Netw.* 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Pasemann, F. (2002). Complex dynamics and the structure of small neural networks. *Netw. Comput. Neural Syst.* 13, 195–216. doi: 10.1080/net.13.2.195.216

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* 1, 206–215. doi: 10.1038/s42256-019-0048-x

Saha, A., Subramanya, A., and Pirsiavash, H. (2020). "Hidden trigger backdoor attacks," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence 2020 (AAAI-20)* (New York City, NY).

Salman, H., Li, J., Razenshteyn, I. P., Zhang, P., Zhang, H., Bubeck, S., et al. (2019). "Provably robust deep learning via adversarially trained smoothed classifiers," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, eds H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett (Vancouver, BC), 11289–11300.

Samek, W., Montavon, G., Binder, A., Lapuschkin, S., and Müller, K. R. (2016). Interpreting the predictions of complex ML models by layer-wise relevance propagation. *arXiv* abs/1611.08191.

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K. R. (eds.). (2019). *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham: Springer.

Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. (2016). "Accessorize to a crime," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, eds E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi (Vienna: ACM), 1528–1540.

Simonyan, K., and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations*, eds Y. Bengio and Y. LeCun (San Diego). Available online at: http://arxiv.org/abs/1409.1556

Singh, G., Gehr, T., Püschel, M., and Vechev, M. (2019). "An abstract domain for certifying neural networks," in *Proceedings of the ACM Symposium on Principles of Programming Languages 2019*, Vol. 3 (Cascais), 1–30.

Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., et al. (2018). "Physical adversarial examples for object detectors," in *12th USENIX Workshop on Offensive Technologies, WOOT 2018*, eds C. Rossow and Y. Younan (Baltimore, MD: USENIX Association).

Song, Q., Yan, Z., and Tan, R. (2019). Moving target defense for deep visual sensing against adversarial examples. *arXiv* abs/1905.13148.

Stanford Vision Lab. (2016). *ImageNet*. available online at: http://image-net.org/index (accessed March 17, 2020).

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). "Revisiting unreasonable effectiveness of data in deep learning era," in *IEEE International Conference on Computer Vision, ICCV 2017* (Venice: IEEE Computer Society), 843–852.

Sun, Q., Zhang, H., Zhang, J., and Zhang, X. (2018). Why can't we accurately predict others' decisions? Prediction discrepancy in risky decision-making. *Front. Psychol.* 9:2190. doi: 10.3389/fpsyg.2018.02190

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., et al. (2014). "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*, eds Y. Bengio and Y. LeCun (Banff, AB).

Tanay, T., and Griffin, L. D. (2016). A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv* abs/1608.07690.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. (2018). "Ensemble adversarial training: attacks and defenses," in *Proceedings of the 6th International Conference on Learning Representations* (Vancouver). Available online at: https://arxiv.org/abs/1705.07204

Tran, B., Li, J., and Madry, A. (2018). "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, eds S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Montréal, QC), 8011–8021.

Turner, A., Tsipras, D., and Madry, A. (2019). Label-consistent backdoor attacks. *arXiv* abs/1912.02771.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. J. (2017). "Learning from noisy large-scale datasets with minimal supervision," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (Honolulu, HI: IEEE Computer Society), 6575–6583.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., et al. (2019). "Neural cleanse: identifying and mitigating backdoor attacks in neural networks," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)* (San Francisco, CA), 707–723.

Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., et al. (2018). "The devil of face recognition is in the noise," in *Computer Vision–ECCV 2018*, eds V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Cham: Springer International Publishing), 780–795.

Ward, E., and Scholl, B. (2015). Stochastic or systematic? Seemingly random perceptual switching in bistable events triggered by transient unconscious cues. *J. Exp. Psychol. Hum. Percept. Perform.* 41, 929–939. doi: 10.1037/a0038709

Werbos, P. (1982). "Applications of advances in nonlinear sensitivity analysis," in *System Modeling and Optimization. Lecture Notes in Control and Information Sciences*, Vol. 38, eds R. F. Drenick and F. Kozin (Berlin; Heidelberg; New York, NY: Springer), 762–770.

Wikipedia Contributors (2020). *Tesla Autopilot—Wikipedia, The Free Encyclopedia*. San Francisco, CA: Wikimedia Foundation, Inc.

Wong, E., and Kolter, J. Z. (2018). "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proceedings of the 35th International Conference on Machine Learning, PMLR* (Stockholm), 5286–5295.

Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. (2018). "Scaling provable adversarial defenses," in *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, QC), 8410–8419.

Wood, G., Vine, S., and Wilson, M. (2013). The impact of visual illusions on perception, action planning, and motor performance. *Atten. Percept. Psychophys.* 75, 830–834. doi: 10.3758/s13414-013-0489-y

Xiao, H., Biggio, B., Nelson, B., Xiao, H., Eckert, C., and Roli, F. (2014). Support vector machines under adversarial label contamination . *J. Neurocomput. Spec. Issue Adv. Learn. Label Noise* 160, 53–62. doi: 10.1016/j.neucom.2014.08.081

Xu, H., Ma, Y., Liu, H. C., Deb, D., Liu, H., Tang, J. L., et al. (2020). Adversarial attacks and defenses in images, graphs and text: a review. *Int. J. Autom. Comput.* 17, 151–178. doi: 10.1007/s11633-019-1211-x

Yakura, H., Akimoto, Y., and Sakuma, J. (2020). "Generate (non-software) bugs to fool classifiers," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence 2020 (AAAI-20)* (New York, NY).

Yousefzadeh, R., and O'Leary, D. P. (2019). Investigating decision boundaries of trained neural networks. *arXiv* abs/1908.02802.

Zahavy, T., Kang, B., Sivak, A., Feng, J., Xu, H., and Mannor, S. (2018). "Ensemble robustness and generalization of stochastic deep learning algorithms," in *International Conference on Learning Representations Workshop (ICLRW'18)* (Vancouver, BC).

Zhang, X., Wang, N., Ji, S., Shen, H., and Wang, T. (2018). Interpretable deep learning under fire. *arXiv* abs/1812.00891.

Zhu, X., Vondrick, C., Fowlkes, C. C., and Ramanan, D. (2016). Do we need more training data? *Int. J. Comput. Vis.* 119, 76–92. doi: 10.1007/s11263-015-0812-2

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.