Ruprecht-Karls-Universität Heidelberg

Fakultät für Mathematik und Informatik

Institut für Informatik

Bachelorarbeit

Adversarial Machine Learning

Name: Frank Walter

Matrikelnummer: 3244418

Betreuer: Prof. Frederik Armknecht (Universität Mannheim),
Dr. Wolfgang Merkle (Universität Heidelberg)

Datum der Abgabe: 22.05.2020

English Abstract

Machine learning algorithms play a central role in many modern software applications. They are being used for computer vision, robotics, language, finance and security applications and many more.
As attackers try to exploit software systems or bypass security measures which involve machine learning algorithms, there is a need to assess and address these dangers to machine learning algorithms.
This is being done by the field of adversarial machine learning. In this work the author will provide an overview on some of the most important adversarial attack and defense techniques as well as a number of real life attacks.
A theoretical framework for developing and testing adversarial defenses based on a threat model will be given and the need for more research in this field of study stressed, as none of the defenses mentioned in this work are robust enough to deal with the smart employment of state-of-the-art attacks.

German Abstract

Machine learning Algorithmen spielen eine zentrale Rolle in vielen modernen Softwareanwendungen. Sie werden in Bildverarbeitungs-, Robotik-, Sprach-, Finanzen- und Sicherheitsanwendungen, etc. verwendet.
Durch die konstante Gefahr eines Angriffs auf solche Systeme müssen die daraus resultierenden Gefahren festgestellt und ihnen entgegnet werden.
Dies fällt dem Forschungsfeld von adversarial machine learning zu. In dieser Arbeit wird der Author einige der wichtigsten Angriffs- und Verteidigungstechniken davon aufzeigen, wie auch tatsächlich vorgekommene oder drohende Angriffe in der Praxis darstellen und die Gegenmaßnahmen zu ihnen beschreiben.
Ein Modell zur Entwicklung und Testung von adversarial machine learning Verteidigungsmaßnahmen basierend auf einem threat model wird gegeben, als auch die Notwendigkeit für weitere Forschung auf dem Gebiet bekräftigt, da keine der in der Arbeit vorgebrachten Verteidigungsmaßnahmen das System vor einem kompetenten Angreifer mit state-of-the-art Angriffen schützen kann.

Table of contents

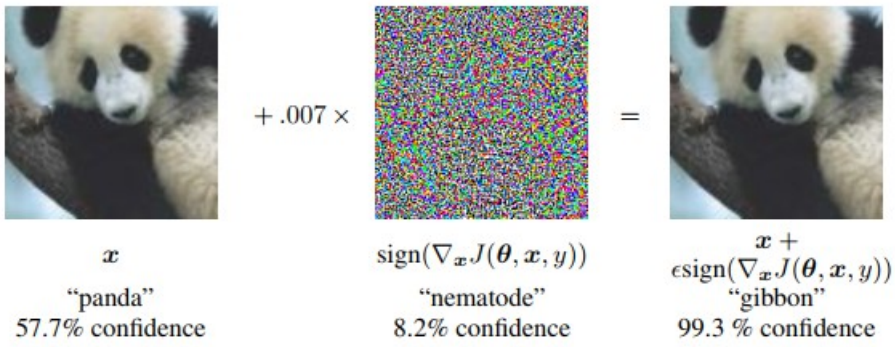

Figure 1[1]: One of the first published adversarial attacks on deep learning algorithms back in 2014 could achieve over 99% misclassification confidence in a false category.

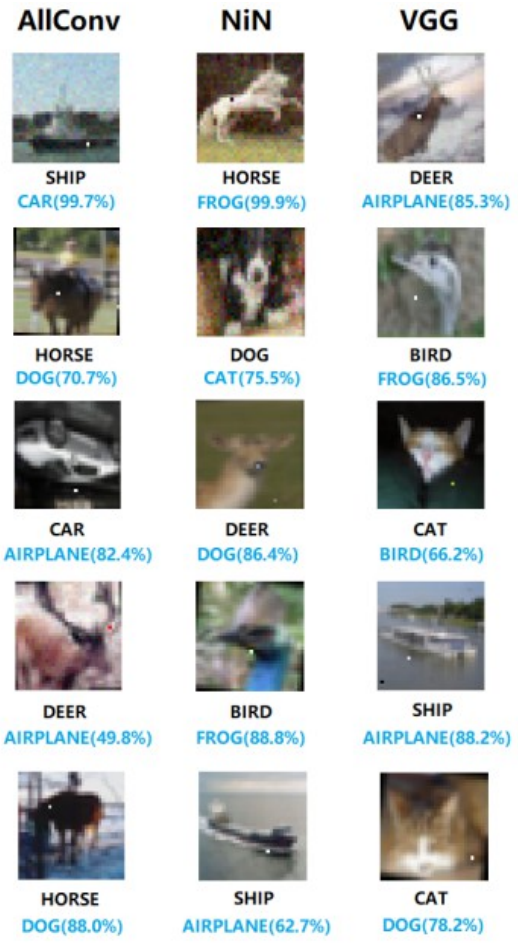

Figure 2[2]: A more advanced one pixel attack leads to misclassification, often with high confidence

1 Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. p. 3

2 Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. In: *IEEE Transactions on Evolutionary Computation*, *23*(5). p. 828

"*Security is an arms race, and the security of machine learning and pattern recognition systems is not an exception to this.*"[3]

Adversarial examples, i.e. attacks on machine learning algorithms, such as those seen in Figure 1 and 2 have shocked our understanding and confidence in machine learning systems.

The first paper to demonstrate how easily non-linear image recognition algorithms can be fooled to cause misclassification with high confidence on images were Szegedy & al.[4], showing that such manipulation can remain imperceptible to the human eye.
Images are one of the most common targets for adversarial machine learning researchers, since the data is automatically already visualized and since that area of machine learning is already fairly advanced and its research community particulary interested in this phenomenon.[5]
However, adversarial examples need not necessarily be minimally perturbed. The reason for this prevailing conception is that this example of instability has been used to highlight the very specific way in which machine learning operates and how it differs from our thinking, rather than as a systematic security assessment, see also 6.1.[6]

But of course, these adversarial examples do not only further our understanding of machine learning algorithms, they can also pose a significant security threat. As machine learning is being deployed more and more in safety critical applications such as health[7] [8], infrastructure[9] [10], finance[11] [12], military[13] [14], etc, the defense of machine learning systems becomes all the more important.

Different assets in the machine learning system may be of importance in different applications and for different attackers:
With autonomous cars for example an attacker may be interested in manipulating the output of the machine learning algorithm, causing the car to behave in an unexpected, often dangerous or damaging way.
In contrast an attacker attacking machine learning algorithms deployed in the healthcare industry may be more interested in recovering the training data used for training the machine learning algorithm, i.e. patient data.[15]
So not only are the outputs of machine learning algorithms assets worth protecting, but sometimes can the machine learning algorithm itself, be it its training data or its model and parameters, become

3 Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. In: *Pattern Recognition, 84*. p. 319.
4 Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. *arXiv. arXiv preprint arXiv:1312.6199*.
5 see Biggio, B., & Roli, F. (2018). p. 318 + 323 - 324
6 ibid. p. 324
7 see https://www.bbc.com/news/health-38055509, last viewed in May 2020
8 see https://www.technologyreview.com/2016/03/09/8890/the-artificially-intelligent-doctor-will-hear-you-now/, last viewed in May 2020
9 see https://www.disruptordaily.com/ai-disrupting-energy-industry/, last viewed in May 2020
10 see https://www.technologyreview.com/2018/08/17/140987/google-just-gave-control-over-data-center-cooling-to-an-ai/, last viewed in May 2020
11 see https://hbr.org/2015/03/artificial-intelligence-is-almost-ready-for-business, last viewed in May 2020
12 see https://emerj.com/ai-sector-overviews/ai-for-credit-scoring-an-overview-of-startups-and-innovation/, last viewed in May 2020
13 see https://www.technologyreview.com/2015/08/03/166882/military-robots-armed-but-how-dangerous/, last viewed in May 2020
14 see https://futureoflife.org/2019/05/09/state-of-ai/?cn-reloaded=1, last viewed in May 2020
15 see Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*. p. 1 + 6

the target of attacks.[16]
In Figure 3 we can see the role which machine learning algorithms usually play in machine learning systems and at which points attacks on the system may be possible.

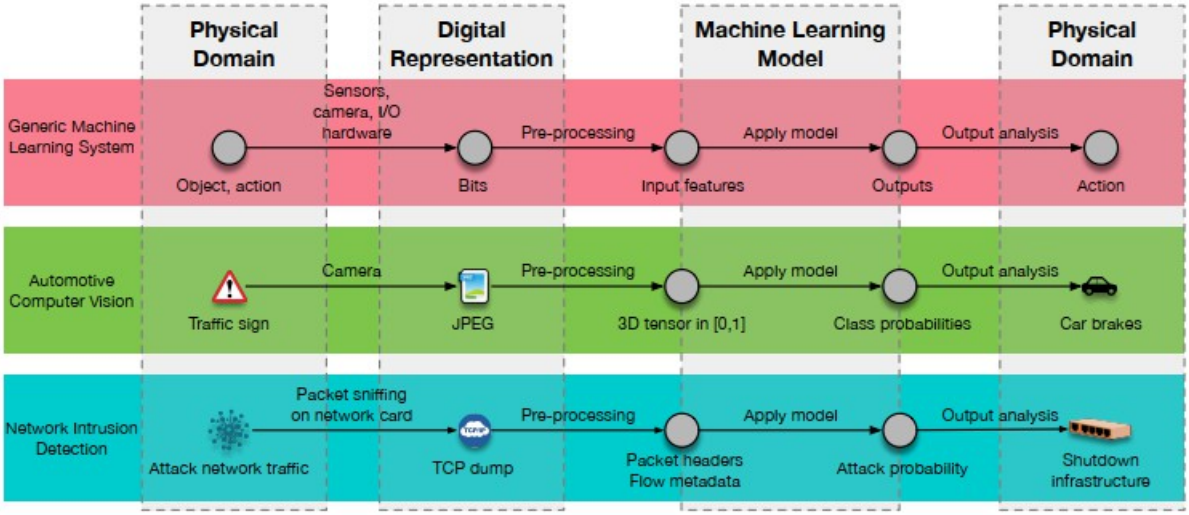

Figure 3[17]: A model showing machine learning systems' attack surface

Additionally, while adversarial machine learning mainly deals with attacks on machine learning algorithms, Papernot & al. have pointed out that other parts of the machine learning system, besides the machine learning algorithms employed, could be targeted such that "[a]dversaries [] attempt to manipulate the collection and processing of data, corrupt the model, or tamper with the outputs."[18] This is exceptionally being demonstrated by physical-world attacks, see 5.6, in which a physical object is being slighty manipulated and thus rendered irrecognizable or as a different object of choice altogether. Another example are be poisoning attacks, where the training data set for the machine learning algorithm is being corrupted beforehand, see 4.2.

One way of defense against training data privacy attacks highlighted by Papernot & al. is to change the training data to a privacy-preserving new data set or even an encrypted data set, which even if extracted would no longer reveal sensitive data,[19] showing an application of traditional security research applied to adversarial machine learning, increasing the security of the system without changing the machine learning algorithm.

It should be noted that besides adversarial machine learning there are other research fields at the intersection between machine learning and security, namely the use of machine learning to increase cybersecurity[20] [21] and the research into AI safety, concering itself with inherent problems of machine learning leading to negative outcomes.[22] [23]

16 see ibid. p. 1
17 ibid. p. 6
18 ibid. p. 5
19 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 13
20 see Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. In: *IEEE Communications surveys & tutorials*, *18*(2). p. 1153 - 1176.
21 see also Yavanoglu, O., & Aydos, M. (2017). A review on cyber security datasets for machine learning algorithms. In: *2017 IEEE International Conference on Big Data (Big Data)*. p. 2186 + 2188 - 2189
22 Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.
23 see also a list of misbehaving AI in https://docs.google.com/spreadsheets/d/e/2PACX-1vRPiprOaC3HsCf5Tuum8bRfzYUiKLRqJmbOoC-32JorNdfyTiRRsR7Ea5eWtvsWzuxo8bjOxCG84dAg/pubhtml,

Furthermore this work focuses on simpler machine learning models for classification or regression applications. For the sake of conciseness in this work, adversarial machine learning in relation to reinforcement learning and Generative Adversarial Networks is largely ignored in this work, although many of the strategies described here for developing robust machine learning algorithms can be applied to them as well.

2. Glossary of relevant machine learning techniques and terminology

Giving an in-depth exposé of the machine learning techniques and algorithms used as a basis for the attacks and countermeasures proposed here is neither feasable nor desirable.
Instead the author provides a glossary describing some of the important terminology used in this work.

Machine learning
... "provides automated methods of analysis for large sets of data."[24]

Adversarial machine learning
... deals with the security of machine learning in an adversarial environment.[25]

Machine learning algorithm
... describes the method used for machine learning. "Most ML models can be seen as parametric functions $h_\theta(x)$ taking an input x and a parameter vector θ"[26].

Supervised vs unsupervised learning
... describes whether the training data is first being labeled (supervised) or not (unsupervised) before it is used for training. Supervised learning therefore depends on human knowledge, whileas unsupervised does not.[27]

Neural networks
... describe machine learning based on the use of so-called neurons, which are basic input and output precessing units.[28]

Deep learning
... refers to machine learning procedures, which make use of models consisting of many layers, such as the many hidden layers in feature learning in Figure 4.[29]

---

last viewed in May 2020, which seem to be problems inherent to the way these systems are trained and therefore probably will continue to exist in more powerful systems as well.

24 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 2

25 see Carlini, N., & Wagner, D. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. p. 2

26 ibid. p. 3

27 Smith, P. D. (2018). Hands-on artificial intelligence for beginners: an introduction to AI concepts, algorithms, and their implementation. *Birmingham, UK: Packt Publishing*. Accessed at https://learning.oreilly.com/library/view/hands-on-artificial-intelligence/9781788991063/ in April 2020. Machine learning basics / constructing basic machine learning algorithms

28 ibid. Your First Artificial Neural Networks / Network building blocks /

29 see Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In: *2016 IEEE Symposium on Security and Privacy (SP)*. p. 583

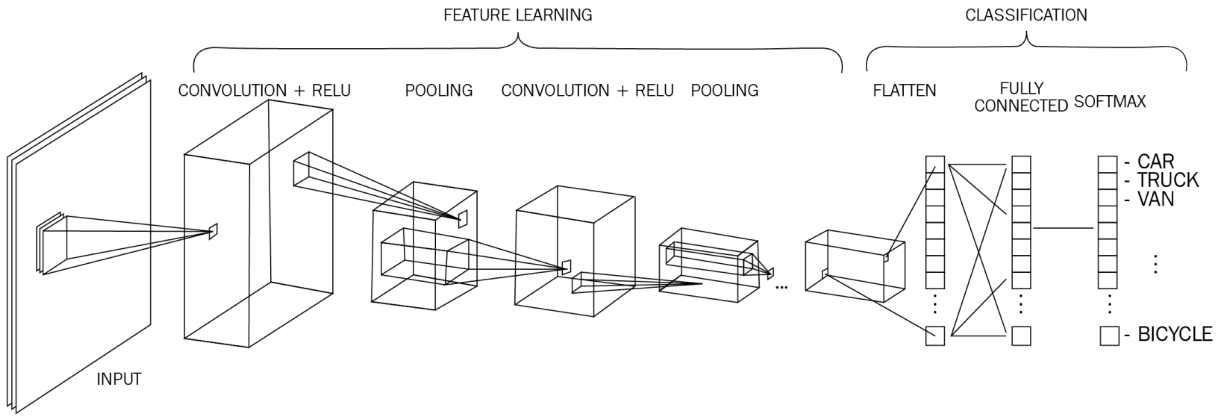

Figure 4: A schematic showing a convolutional neural network machine learning flow[30]

Gradient
... usually refers to the calculus of the backpropagation. This gradient is then used to retrain the weights of the neural network along the gradient to reduce the loss function.[31]
Gradient based attacks instead adjust the input, in order to optimize either the loss function for evasion attacks or the calculus of the backpropagation for poisoning attacks.

Gradient masking
... aims to "reduce the sensitivity of models to small changes made to their inputs"[32], in order to make the generation of attacks based on the gradient of the machine learning algorithm more difficult.

Feature
"A *feature*[sic] is a numeric representation of raw data. There are many ways to turn raw data into numeric measurements [...] [, i]f there are not enough informative features, then the model will be unable to perform the ultimate task. If there are too many features, or if most of them are irrelevant, then the model will be more expensive and tricky to train."[33]

Feature selection or rarely feature choice
"Feature selection employs computational means to select the best features for a problem"[34] and "is a key technology which can eliminate irrelevant features, reduce data dimensionality, and [...] [whose] algorithms can be broadly divided into two categories, the filter method and the wrapper model. Filter methods select subset of features as a preprocessing step that ignore the effects of the selected feature subset on the performance of learning algorithm. [...] Wrappers use the classification method to score subsets of variables."[35]
Some authors also include embedded methods, which - contrary to the two methods described above - are part of the classification process. This can e.g. take the form of an absolute value

30 Smith, P. D. (2018). Convolutional Neural Networks / Overview of CNNs
31 ibid. Your First Artificial Neural Networks / Summary
32 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 11
33 Zheng, A., & Casari, A. (2018). Feature engineering for machine learning: principles and techniques for data scientists. *O'Reilly Media, Inc*. Accessed at https://learning.oreilly.com/library/view/feature-engineering-for/9781491953235/ in May 2020
The Machine Learning Pipeline / Features
34 ibid. Fancy Tricks with Simple Numbers / Interactive Selection
35 Liu, Q., Chen, C., Zhang, Y. & Hu, Z. (2011). Feature selection for support vector machines with RBF kernel. In: *Artif Intell Rev 36*. p. 100

regularization.[36] Good feature selection is not only important for the performance and generalizability of the machine learning algorithm, but also for robustness, see 8.4.

SVM, SVM-RBF
SVM is the abbreviation for Support Vector Machine, with the radial basis function being a popular basis for a non-linear kernel, i.e. a wrapper algorithm.[37]
While machine learning algorithms with non-linear kernels such as SVM-RBF may be efficient, their "feature ranking criterion is unknown and the weight vector cannot be computed explicitly."[38]

Differentiable and non-differentiable Learning
Differentiable learning algorithms have a gradient for the Loss function with respect to the input, non-differentiable do not. The former category includes neural networks and SVM with differentiable kernels, the latter decision trees and random forests.[39]

Loss Function
... describes a "measure of difference between the predicted distribution in our training data, and its actual distribution."[40]

Weights
... describe the numbers associated with every layer in a neural network, which are used as parameters of the "function" of transforming input to output in the neural network. Change with training.[41] [42]

Regularization
... usually describes a penalty term added to the error of the Loss function, punishing either the absolute or relative value of weights. Dropout, i.e. the random removal of nodes and their connections in neural networks, is also refered to as regularization.[43]

Classification
... one of the main tasks machine learning algorithms are used for. For given input data, one of several labels is being chosen by the machine learning algorithm.[44]

Classifier
... is a machine learning algorithm used for classification, "identifying a sample as being from some output class, among a predefined finite of [potentially] many classes - trained on labeled data."[45]

Over-/ Underfitting
"**Overfitting**[sic] is a phenomenon that happens when an algorithm learns it's training data *too well*[sic] to the point where it cannot accurately predict on new data. Models that overfit learn the small, intricate details of their training set and don't generalize well. [...] Underfitting would be exact opposite of this [...] [f]rom a modeling standpoint, an underfit model is not complex enough

36 Zheng, A., & Casari, A. (2018). Fancy Tricks with Simple Numbers / Feature Selection
37 Liu, Q., Chen, C., Zhang, Y. & Hu, Z. (2011). p. 100
38 ibid. p. 111
39 Biggio, B., & Roli, F. (2018). p. 322
40 Smith, P. D. (2018). Your First Artificial Neural Networks / The training process / Backpropagation
41 ibid. Your First Artificial Neural Networks / Network building blocks / Weights and bias factors
42 ibid. Your First Artificial Neural Networks / The training process / Backpropagation
43 ibid. Your First Artificial Neural Networks / Network building blocks / Regularization
44 ibid. Your First Artificial Neural Networks / Network building blocks /
45 McDaniel, P., Papernot, N., & Celik, Z. B. (2016). Machine learning in adversarial settings. In: *IEEE Security & Privacy*, *14*(3). p. 68

to generalize to new data."[46]

Adversarial training
... refers to the defense technique of including adversarial examples in the training set.[47]

Robustness
... can be described for our purpose as "the average minimal perturbation required to produce an adversarial sample"[48], constituting on of our main measures for evaluating the security of machine learning algorithms. It is important to note however that this measurement not universal, but relative to some attack method, with more advanced attack methods potentially lowering the robustness of the same machine learning algorithm, since calculating robustness by bruteforce is computationally too demanding. The opposite of brittleness.

3. A framework for developing robust adversarial machine learning techniques

This framework is mostly based on a paper from Biggio and Roli[49], in which they not only provide an overview over the development of adversarial machine learning and the increasingly overstated role that minimally perturbed adversarial examples have played in it, culminating in the conception that adversarial machine learning started with these adversarial examples[50], but also construct a framework for developing proactive machine learning defense which they see as an important step in the direction of making machine learning robust.[51]
This framework is sometimes referenced in subsequent papers., especially the categorization of attacks by knowledge of the attacker.[52]

3.1. Know your attacker

Notation[53]:

We shall introduce some basic notation on the basis of Biggio and Roli to simplify referencing important items and show their relationship to each other.

$$D = (x_i, y_i)_{i=1}^{n}$$

X and Y denote the sample and label spaces respectively, the training data is represented by D and Loss is denoted by L = (D,w), where w is the parametrized classifier of f : X ↦ Y.

Goal[54]:

There are different kinds of goals an attacker can try to achieve with his attack.

46 Smith, P. D. (2018). Machine Learning Basics / Basic Tuning / Overfitting and Underfitting
47 Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*. p. 4
48 Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). p. 595
49 Biggio, B., & Roli, F. (2018).
50 ibid. p. 317
51 ibid. p. 318
52 e.g. Carlini, N., & Wagner, D. (2017). p. 4
53 Biggio, B., & Roli, F. (2018). p. 319
54 ibid.

Firstly, the attacker can have different overarching goals he wants to achieve with his attack. These s*ecurity violations* can compromise the machine learning algorithm in different ways, namely:

1. *Integrity*:	He can try to evade detection by the machine learning algorithm without compromising normal system operation, through information modification or destruction in the training process of the algorithm.
2. *Availability*:	He can try to compromise the normal system functionalities available to legitimate users, thereby increasing the cost of using the algorithm potentially to the point where using it is less cost-efficient than other solutions.
3. *Privacy/ Confidentionality:*	He can try to obtain private information about the system, its users or training data by reverse-engineering the learning algorithm.

Secondly, the attack can either be *targeted* at selected sample categories that the attacker wants to be mislabed, or the attack can be *indiscriminate* insofar as the attacker just wants as many sample categories as possible to be mislabeled. This attack focus is called the *attack specificity.*

Thirdly, the attacker can then try to have the mislabeled categories be a very *specific* group of categories or just any *generic* misclassification. This pursued effect of the attack is called the *error specificity.*

Knowledge[55]:

The attacker can have different amounts of knowledge of any given machine learning system. The three different categorizations of his potential knowledge space Θ of the training data D, the feature set Z, the learning algorithm f as well as the corresponding Loss function L, and lastly the trained parameters w. Incomplete knowledge of a variable shall be denoted with a hat symbol.

Perfect knowledge (PK) white-box attacks:

With perfect knowledge, i.e. white-box attacks the attacker is assumed to have complete knowledge of the given machine learning system, i.e.

$$\theta_{PK}=(D, Z, f, w)$$

With such complete knowledge the attacker can simulate different attacks on the system and evaluate its effects on the system, choosing an attack that surpasses some kind of empirical upper bound of the attacks effect on the system.

Carlini and Wagner have expanded on this by suggesting that a modelled attacker should not only be aware of the machine learning algorithm, but also of the defenses employed, in order to properly evaluate the defense strength.[56] This type of adversary has been called adaptive adversary in contrast to a static adversary.[57]

Limited knowledge grey-box attacks:

55 ibid. p. 319 - 320

56 Carlini, N., & Wagner, D. (2017). p. 13

57 He, W., Wei, J., Chen, X., Carlini, N., & Song, D. (2017). Adversarial example defense: Ensembles of weak defenses are not strong. In: *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*. p. 3

With limited knowledge the attacker is generally assumed to have incomplete knowledge of the training data D used for training the machine learning algorithm, as well as having incomplete knowledge of the final parameters of the algorithm. Biggio and Roli call this kind of scenario surrogate data, since the attacker usually has some rough idea what kind of data might have been used and can use similar data to train his own final parameters w of the algorithm, summarized as:

$$\theta_{LK-SD}=(\hat{D}, Z, f, \hat{w})$$

If the attacker additionally does not know the particular learning algorithm used or cannot replicate it (e.g. because of computational limitations), then they call this scenario surrogate learner:

$$\theta_{LK-SL}=(\hat{D}, Z, \hat{f}, \hat{w})$$

Biggio and Roli argue that grey box adversarial examples with a surrogate learner can be highly transferable.[58] This was already shown back in 2014 by Šrndić and Laskov by demonstrating that knowledge of the feature set alone and surrogate knowledge of the other variables was able to produce evasion attacks (~67% average misclassification) of comparible effectiveness to attacks with full knowledge (~72% average misclassification) against non-linear learners. The learner in question was a PDF malware detector.[59]

The surrogate learner scenario also applies for example to (non-defensive) distillation, where a compressed version of the original machine learning algorithm is deployed to smartphones, which can then be reverse engineered by adversaries.[60]

Other authors propose more varied scenarios of limited knowledge[61], but they agree that knowledge of the feature set is the most fundamental to know[62] and therefore limit themselves to such scenarios.

Zero knowledge black-box attacks:

Even if black-box attacks suggest that the attacker has zero knowledge of the machine learning algorithm, this is usually not the case. When an attacker attacks a machine learning algorithm, he knows what kind of purpose it serves and should have some rough idea as to how this algorithm was trained.
If the attacker doesn't even know what kind of data is being used to train the algorithm or on what basis it differentiates between valid and invalid samples the attacker has no basis as to how he should go about developing his attack. So we can assume that the attacker has some knowledge as to what feature set is being used (e.g. image pixel for image recognisition), even if he is not aware of what data exactly is being extracted and how.

$$\theta_{ZK}=(\hat{D}, \hat{Z}, \hat{f}, \hat{w})$$

Furthermore, modelling this kind of attack with a surrogate classifier has shown that even attacks on machine learning algorithms with different classifiers can be transferable, as "[i]t is often the

58 ibid. p. 323
59 Šrndić, N. & Laskov, Pavel. (2014). Practical Evasion of a Learning-Based Classifier: A Case Study. In: *Proceedings - IEEE Symposium on Security and Privacy*. p. 206
60 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 7
61 Šrndić, N. & Laskov, Pavel. (2014). p. 198
62 ibid. p. 199

case that, when given two models F(·) and G(·), an adversarial example on F will also be an adversarial example on G, even if they are trained in completely different manners, on completely different training sets",[63] thereby making the category of limited knowledge attacks potentially largely redundant.

Capability[64]:

There are three factors to consider when looking at the attacker's capability:
Firstly his attack influence, i.e. whether he is only able to modify the test data leading to explorative or evasion attacks or whether he is able to modify training data as well leading to causative or poisoning attacks.
Secondly his data manipulation constraints, i.e. how freely he is able to modify his data without compromising other factors. E.g. if the attacker wants to evade malware detectors he cannot freely change the machine code of his malware, as the malware still needs to perform a certain function. The more freely the attacker can modify the sample, the more perturbations are allowed, the more manipulative power he is said to have.[65]
Thirdly his computational power. Similar to the notion of a polynomially bounded adversary in cryptography which assumes that an attacker can only solve problems which require at most polynomial time, Madry & al. assume that an attacker has at most access to first order local information of the gradient.[66] This may change as the field progresses.

The attacker's initial attack samples D_C can thus be modelled as having a space of possible modifications $\phi(D_C)$. The mighter the space of possible modifications is, the stronger the possible attacks become. This is relevant for comparisons of attacks and the defenses against them.

Strategy[67]:

With the attackers knowledge $\theta \in \Theta$ and his modified attack samples $D'_C \in \phi(D_C)$, the attacker's optimizing function may be defined as

$$A(D'_C, \theta) \in \mathbb{R}$$

measuring the presumed effectiveness of the attacks D'_C. The optimal strategy of the attacker is thus:

$$D_C^{opt} \in argmax\, A(D'_C, \theta)$$

This formula holds not only for supervised learning, but for (poisoning) attacks on unsupervised training and feature selection algorithms as well.

3.2. Reactive vs. proactive defense

63 Carlini, N., & Wagner, D. (2017). p. 2
64 Biggio, B., & Roli, F. (2018). p. 320
65 Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*. p. 3
66 ibid. p. 2 + p. 7 - 8
67 Biggio, B., & Roli, F. (2018). p. 320

Adversarial defense in general can broadly be categorized in two basic approaches: Reactive and proactive defense.[68] [69]

Reactive defense uses real past attacks as a basis to counter similar future attacks, while proactive defense mainly aims to prevent zero-day attacks.

Reactive defense has three essential parts:
Firstly, there needs to be a timely detection of incoming attacks. For attack detection other established measures from cybersecurity can be used such as collaborative data sharing or honeypots.
Secondly, if e.g. a learner is vulnerable to such attacks, the learner needs to be retrained to prevent them from effecting him in the future. Not only does he need new training data or a new algorithmn for it, but he also needs thirdly some kind of human verification that the learner is indeed no longer vulnerable to that kind of attack.[70] This approach can be integrated into adversarial training or used for testing other defenses.

Bug bounty programs[71] can be seen as one example of reactive defense being actively applied in modern IT security:
Major companies such as Google, Goldman Sachs or Telekom offer different kind of rewards for the discovery and discrete disclosure of their vulnerabilities, which can be seen as an useful addition to trying to discover past or ongoing attacks, what not only may not be successful, but this approach may prevent vulnerabilities from being maliciously exploited in the first place, thereby making bug bounty programs not fit neatly into either the category of reactive or proactive defense.

Bug bounty programs are to the knowledge of the author not used in adversarial machine learning, but they may become quite useful, once the robustness of machine learning algorithms reach a certain level of sophistication.
Since most contemporary countermeasures by companies against unintended behavior by machine learning algorithms have been very crude, see 6.2. and 6.4., having more adversarial examples is not going to be very useful in a lot of cases at this point in time.

Proactive defense consists of two parts:

On the one hand there is security-by-obscurity. Security-by-obscurity aims to prevent attackers from developing attacks against the system by limiting the amount of information the attacker has about the it. These disinformation techniques can include in our case randomization of training data collection, hiding public information about the classifier as well as defenses against probing attacks against it.
However Biggio and Roli call into question how effective such defenses are, when surrogate attacks - transfer attacks created using different learners as the target - have been shown to be very effective[72], arguably showing that Kerckhoff's Principle, which states that "a cryptographic system should rely on the secrecy of the keys" rather than on the "*security by obscurity*[sic] principle"[73]

68 see Cho, J., Sharma, D. P., Alavizadeh, H., Yoon, S., Ben-Asher, N., Moore, T. J., Kim, D. S., Lim, H. & Frederica F. Nelson, F. F. (2020). Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. In: *IEEE Communications Surveys & Tutorials*. p. 709 - 745(2020).
69 see also e.g. https://www.fortinet.com/blog/industry-trends/reactive-vs--proactive-cybersecurity--5-reasons-why-traditional-.html, last viewed in April 2020
70 Biggio, B., & Roli, F. (2018). p. 327
71 for example https://www.bugcrowd.com/bug-bounty-list/, last viewed in April 2020
72 Biggio, B., & Roli, F. (2018). p. 328
73 Romdhani, I. (2017). Chapter 7 - Existing Security Scheme for IoT. In: *Securing the Internet of Things*. *Elsevier Inc*.

applies in a sense to adversarial machine learning as well, echoed for example in one of the principles forwarded by the National Institute of Standards and Technology: "System security should not depend on the secrecy of the implementation or its components."[74]
It may even be argued that adversarial machine learning defense techniques should not rely on any secrecy at all, if the goal is to be able to defend against adaptive adversaries.

3.3. Security-by-obscurity & security-by-design

3.3.1. Security-by-obscurity

In the context of adversarial machine learning, security-by-obscurity has two components: One is reducing the publicly available amount of information of the machine learning algorithm and the other is disinformation techniques to hide information from potential attackers[75], with the latter only meaningfully being able to obscure information, if it has not already been publicized prior.
The security-by-obscurity approach can therefore be said to try to make and keep the program as much of a black box to a potential attacker as possible[76].

Some machine learning algorithms such as ResNet[77] have most relevent data for an attacker about them published (except for the specific classification training used to adapt the general model to the specific task), thereby forgoing security-by-obscurity and leaving themselves open to whitebox attacks. For the machine learning research community in general it seems to be very difficult to be able to reconcile reproducibility with security-by-obscurity, even if the latter were to be a point of concern, which it usually isn't.[78] It simply runs counter to reproducibility and contribution to discourse.
This makes security-by-design approaches generally more reconcilable with the way machine learning algorithms are developed in practise, although there is no need to focus on one strategy to the point of disparagement of the other.

3.3.2. Security-by-design

The term security-by-design in relation to cybersecurity is applied in various forms in different subfields: From the use of embedded security hardware to replace some security software, therefore having your security already built in[79], to the secure development of products such as cloud services[80] to the modelling the interplay of usability and security.[81]

Security-by-design as opposed to security-by-obscurity in trying to make the system secure even against white-box attacks can be seen as a derivation from Kerckhoff's principle; in order to achieve

---

p. 120

74 Scarfone, K., Jansen, W., Tracy, M. (2008). Guide to General Server Security. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-123, p. 15 (page count 2-4)

75 Biggio, B., & Roli, F. (2018). p. 328

76 a slight expansion of the definition given at ibid.

77 He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 773-776

78 There is no mention of security of the learner, much less security by obscurity, in a major publication such as ibid., nor in any other publication cited here not explicitly concerning itself with adversarial machine learning.

79 Souren, J. (2013). Security by design: hardware-based security in Windows 8. *Computer Fraud & Security*, *2013(5)*. p. 18 - 20.

80 Casola, V., De Benedictis, A., Rak, M., Rios, E. (2016). Security-by-design in Clouds: A Security-SLA Driven Methodology to Build Secure Cloud Applications. In: *Procedia Computer Science, Volume 97*. p. 53 - 62

81 Faily, S., Lyle, J., Ivan, & Simpson, A. (2015). Usability and security by design: a case study in research and development. In: *Internet Society NDSS Symposium 2015*

this security-by-design relies on trying "to make systems as free from vulnerabilities as possible by taking into account security from the very early stages of the design process."[82]

For our use in adversarial machine learning we shall focus on two components of security-by-design: Preventing past attacks from reoccurring and modelling attacks to prevent zero-day attacks.[83] The attentive reader will have noticed, that both proactive and reactive defenses try to detect attacks and counter attacks. Indeed, Biggio and Roli state that even security-by-obscurity has detection of attacks as one of its components, see Figure 5.

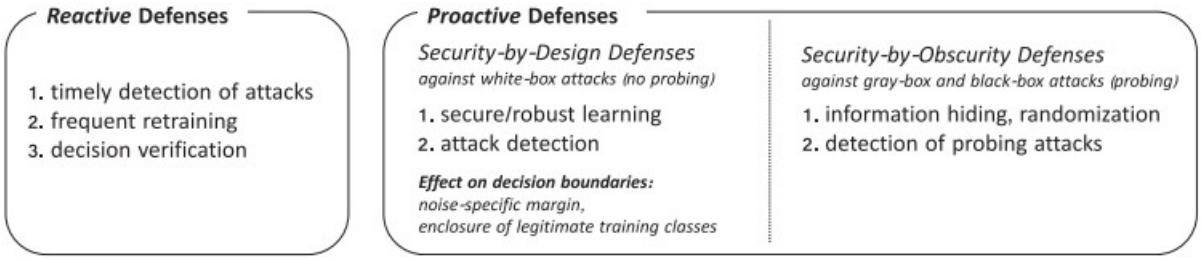

Figure 5[84]: Both reactive and proactive defense have attack detection as central components

While it shows that there may not always be a clear cut line between these approaches as seen in the bug bounty programs, two differences should be mentioned: While reactive defense mainly deals with the detection of past attacks, proactive defense focuses more on the detection of attacks as they are occuring and stopping them in their tracks. The second difference is how knowledge of past attacks is being handled: Reactive defense generally focuses on preventing this specific type of attack from reoccuring, whileas proactive defense tries to secure the system on a more fundamental level, potentially securing the system against many other similar and not so similar attacks.

4. Categorisation of attacks

	Attacker's Goal: Misclassifications that do not compromise normal system operation	Misclassifications that compromise normal system operation	Querying strategies that reveal confidential information on the learning model or its users
Attacker's Capability	**Integrity**	**Availability**	**Privacy / Confidentiality**
Test data	Evasion (a.k.a. adversarial examples)	-	Model extraction / stealing and model inversion (a.k.a. hill-climbing attacks)
Training data	Poisoning (to allow subsequent intrusions) – e.g., backdoors or neural network trojans	Poisoning (to maximize classification error)	-

Figure 6[85]: The attacker's goal and capability lead to the attack method

82 Casola, V., De Benedictis, A., Rak, M., Rios, E. (2016). p. 53

83 compare Biggio, B., & Roli, F. (2018). p. 326, where the closest characterization of security-by-design to a definition has been: "how to react to *past* attacks and prevent *future* ones[italics by original author]", which in the view of this author is not a very useful way of further classification of security-by-design, as this rather describes what it does rather than what it is.

84 ibid. p. 327

85 ibid. p. 321

There different ways in which the goal of attacks on machine learning algorithms can be categorized are shown in Figure 6.[86]
Papernot & al. expand on this by explaining that "[a]ttacks on confidentiality attempt to expose the model structure or parameters (which may be highly valuable intellectual property) or the data used to train it, e.g., patient data[...][, which] is often of paramount importance." And whereas attacks on integrity aim to induce certain output or behavior of the attackers choosing, attacks on availability try to prevent access to features or meaningful outputs of the system.[87]

4.1. Evasion attacks

Evasion attacks use manipulated input during test time to have it misclassified.[88] [89]
They can be either error-generic or error-specific, but generally have a targeted attack specificity as the attacker typically wants to have some specific samples of his to be mislabeled. For example trying to evade a face detection algorithm for yourself, can be considered an error-generic targeted attack on it. In contrast if you use a picture or video of a person to fool a face detection algorithm that would be a error-specific targeted attack.

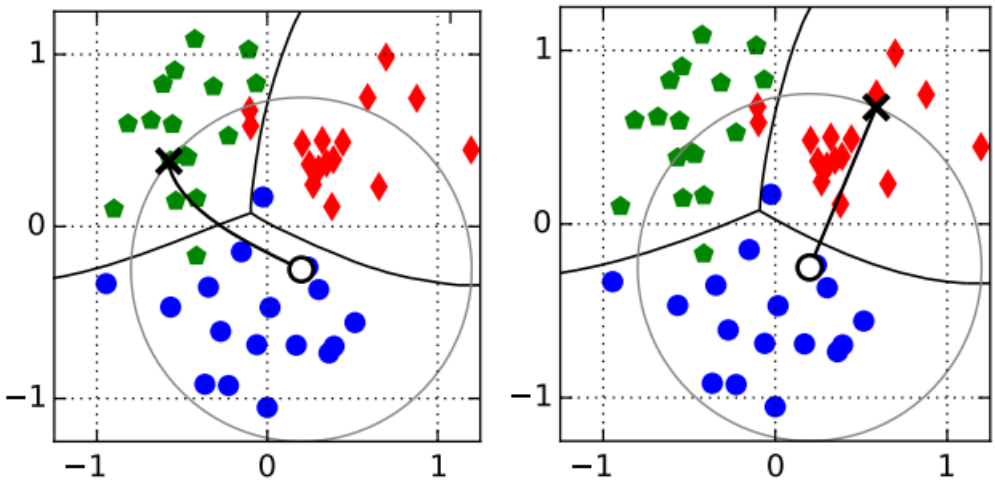

Figure 7[90]: Left: An error-specific attack tries to have the sample misclassified as a green pentagon with the highest confidence possible.
Right: An error-generic attack tries to get the algorithm to have the maximum possible confidence in any wrong class.

Error-generic attacks try to maximise confidence in any erroneous classification of the sample in question. Assuming an equally distributed feasible domain of data manipulation, this attack would bring the sample in the direction of the closest bordering categorisation, as seen on the right in Figure 7.
The optimization formula for error-generic evasion attacks in this case may thus be given as[91]:

86 see also Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: *2018 IEEE Symposium on Security and Privacy (SP)*. p. 19
87 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016).. p. 1
88 Biggio, B., & Roli, F. (2018). p. 321
89 Handa, A., Sharma, A., & Shukla, S. K. (2019). Machine learning in cybersecurity: A review. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(4)*. p. 5
90 Biggio, B., & Roli, F. (2018). p. 322
91 ibid. p. 321

$$\max_{x'} \; \mathcal{A}(x', \theta) = \Omega(x') = \max_{l \neq k} f_l(x) - f_k(x) \,,$$

$$\text{s.t.} \;\; d(x, x') \leq d_{\max} \,, \;\; x_{\text{lb}} \preceq x' \preceq x_{\text{ub}} \,,$$

$f_i(x)$ gives the confidence value of the trained classifier for the class i and input sample x, k being the correctly associated class to x.
Two types of data manipulation constraints $\phi(D_C)$ are given: the first possible constraint being a distance contraint between the initial sample x and manipulated sample x' and the second possible contraint being a box constraint, limiting the degree of manipulation of values of x. [92]

Error-specific attacks try to maximise confidence in the erroneous classification of the sample in a predetermined other class, as seen with green on the left of Figure 7. So in this case the attacker tries to maximise confidence in the pre-determined class while minimizing confidence in the correct class. This can be done by switching the algebraic sign of the objective function *A* and relabeling f_k to be the targeted class, as done by Biggio and Roli:[93]

$$\mathcal{A}(x', \theta) = -\Omega(x')$$

Alternatively the former formula can be kept by having l as the targeted class and k representing the other classes.
Evasion attacks can be classified either as *dense* or *sparse* attacks.[94]
Dense attacks usually attack all values or all values within a region minimally within a box constraint, leading to e.g. a blurring effect on an image. Here the data manipulation constraint is to the value or region of the pixels, but indifferent to their number.
In contrast do *sparse* attacks often attack only a few values, but alter them significantly, as seen in Figure 8 and more impressively in Figure 2. In Figure 8 the data manipulation constraint is a distance constraint between the initial sample and the manipulated sample, similar to the gray circle in Figure 7.

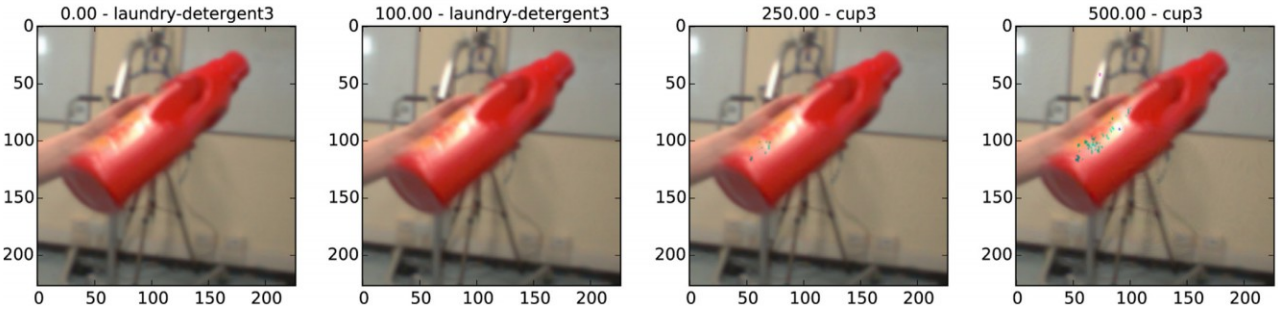

Figure 8[95]: A sparse evasion attack with increasing strength leads to an increasingly strong 'salt-and-pepper' noise effect[96]

92 ibid. p. 322
93 ibid.
94 ibid.
95 ibid. p. 323
96 ibid. p. 322

4.2. Poisoning attacks

In contrast to evasion attacks, which manipulate data during test time, poisoning attacks make use of manipulated data during training time to later cause misclassifications during test time.[97] [98] They can be targeted or indiscriminate.[99]
While most research deals with the insertion of new maliciously created training data points into the training data pool such as the experiment whose evaluation is presented in Figure 9 or the VirusTotal poisoning described in 6.3., Papernot & al. remind us that attackers may also manipulate existing data[100], as seen for example in physical world attacks, see 5.4.

Poisoning attacks can be interpreted either as attacks on availability or on integrity. The differentiation between both types of attacks is less based on the method used by the attacker, but rather by his goal: If the goal of the attack is to "to reduce the quality (e.g.,[sic] confidence or consistency), performance (e.g., speed), or access (e.g., denial of service)" of the system, then it is an attack on availability. If the goal is manipulating the outputs, then it's an attack on integrity.[101]

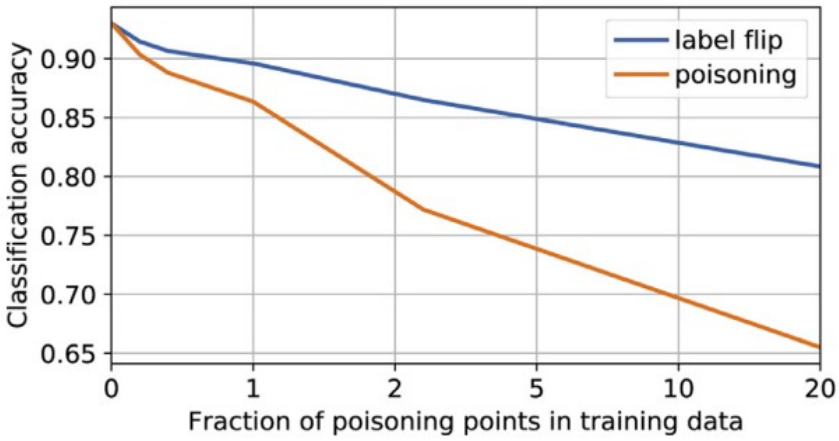

Fig. 9[102]: A comparison of a worst case error-generic poisoning attack and random label flipping on the accuracy of a classification algorithm

Error-generic poisoning attacks try to cause as many misclassifications as possible.
In the case of the attack shown in Figure 9, if the goal of the attacker was the misclassification of the outputs it's an attack on integrity. If the goal was rather to render the program unusable through rampant misclassification it should be rather interpreted as an attack on availability.

$$\mathcal{D}_c^\star \in \arg\max_{\mathcal{D}_c' \in \Phi(\mathcal{D}_c)} \quad A(\mathcal{D}_c', \boldsymbol{\theta}) = L(\mathcal{D}_{\text{val}}, \boldsymbol{w}^\star)\,,$$

$$\text{s.t.} \quad \boldsymbol{w}^\star \in \arg\min_{\boldsymbol{w}' \in \mathcal{W}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}_c', \boldsymbol{w}')\,,$$

97 ibid. p. 324
98 Handa, A., Sharma, A., & Shukla, S. K. (2019), p. 5
99 Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E., & Roli, F. (2017). Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (*AISec '17*). p. 30
100 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 5
101 ibid. p. 6
102 Biggio, B., & Roli, F. (2018). p. 326

D_{val} is a validation data set and D_{tr} is a training data set available to the attacker.
The two equations can be used in optimizing for poisoning attacks: the outer function calculates the Loss function on D_{val} with the poisoned weights w*, whereas the inner function trains a poisoned classifier w' on D_{tr} with the corrupted data D'_C.

As with evasion attacks, error-specific attacks can be optimized for by trying to minimize Loss/optimize confidence in a selected set of classes. The formula can be adjusted by only selecting the classes to attack in D'_{val} and inversing the algebraic sign before the loss function, thereby seeking to minimize loss for the selected classes in order for them to be misclassified to these desired labels.[103]

$$\mathcal{A}(\mathcal{D}'_c, \boldsymbol{\theta}) = -L(\mathcal{D}'_{\text{val}}, \boldsymbol{w}^\star)$$

Even by replacing the inner equation by its equilibrium condition, enabling gradiant computation in closed form, solving these types of equations is very computationally intense, especially in the case of deep learning with a high number of weights.[104] One solution to this problem has been proposed by Muñoz-González & al.[105]

4.3. Classification by attack generation

Another way to categorize attacks is by looking at their generation method.
Two methods stand out: gradient-based attacks and transfer attacks.
We will see several gradient-based attacks in the next chapter, with 5.2. being based on the optimization problem described in this chapter.

As for transfer attacks, an in-depth analysis of transferability of poisoning and evasion attack has been provided by Demontis & al. In particular they show that surrogate models using simpler models transfer better for evasion attacks, if they only have access to a limited training data set, but for poisoning attacks models of similar complexity have the most success.[106]

5. Attack techniques

5.1. Fast Gradient Sign Method and Projected Gradient Descend / Iterative FGSM

The Fast Gradient Sign Method "performs a single step update on the original sample x along the direction of the gradient of a loss function".[107] It was developed primarily to show that neural networks are highly linear, not to develop a strong attack.[108] It is one of the most popular attacks due to its low computational load.
Projected Gradient Descend or Iterative FGSM describes the iterative version of the Fast Gradient Sign Method, which by virtue of having (experimentally shown) converging values of the Loss function regardless of the starting points is being said to be "a universal adversary among all the

103 ibid. p. 324 - 325
104 ibid. p. 325
105 Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E., & Roli, F. (2017).
106 Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C. & Roli, F. (2019). Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: *28th {USENIX} Security Symposium ({USENIX} Security 19)*. p. 329 + 332
107 Zheng, T., Chen, C., & Ren, K. (2019). Distributionally adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence* (*Vol. 33)*. p. 2254
108 Carlini, N., & Wagner, D. (2017). p. 13

first-order adversaries [i.e. attack generation methods]".[109] However there have already been second-order adversaries introduced,[110] which may soon be able to generate more powerful attacks. This technique generates dense evasion attacks.

5.2. Poisoning with Backgradient-Optimization

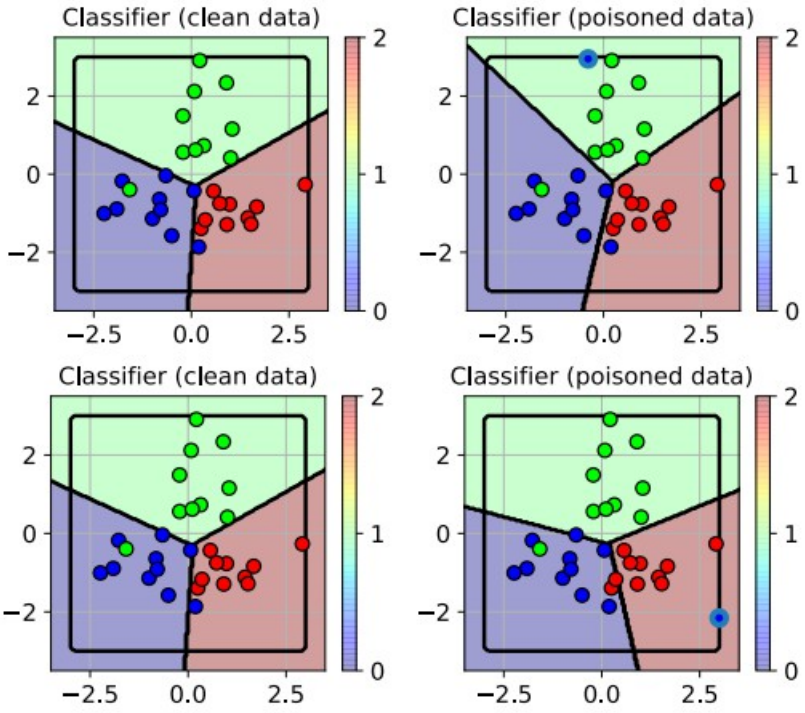

Figure 10[111]: Visualization of classification boundary shift with a single poisoned data point. The top row is error-generic and the bottom row error-specific to misclassify reds as blues and keep the other classifications as much as possible.

Another gradient attack technique that can generate poisoning attacks has been proposed by Muñoz-González & al., which seems to be very effective even if the attacker only controls a small portion of the training set, see Figure 10. The adversarial examples generated by this technique are transferable like those of evasion attacks.[112] Other poisoning attacks include work by Yang & al.[113] and Chen & al.[114]

5.3. JSMA Jacobian-based Saliency map approach

Carlini and Wagner consider JSMA to be one of the weaker attacks known in 2017, albeit it is sometimes in trying to prove the defensive capability of an adverarial defense.[115] It "modifies pixels

109 Zheng, T., Chen, C., & Ren, K. (2019). p. 2254
110 Li, B., Chen, C., Wang, W., & Carin, L. (2018). Second-order adversarial attack and certifiable robustness. p. 3
111 Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E., & Roli, F. (2017). Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (*AISec '17*). p. 32
112 ibid. p. 37
113 Yang, C., Wu, Q., Li, H., & Chen, Y. (2017). Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*.
114 Chen, S., Xue, M., Fan, L., Hao, S., Xu, L., Zhu, H., & Li, B. (2018). Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach. In: *computers & security*, *73*. p. 326 - 344
115 Carlini, N., & Wagner, D. (2017). p. 13

that, based on the gradient, would most strongly reduce the confidence of a correct classification."[116] It can therefore be classified as an sparse evasion attack.

5.4. Physical-world attacks

As machine learning algorithms are being used more and more in safety-critical situations with input from the physical world, attacks on these systems through adversarial examples with perturbations added to the physical object themselves are becoming a real danger.[117]

Robust Physical Perturbations[118] are presented as a technique to modify a physical object without manipulating their background to decrease the accuracy of classifiers under a variety of angles and distances from over 90% to less than 20%, in one case causing the misclassification of a microwave as a phone by placing just one black and white sticker.[119]
Another type of physical-world adversarial attack would be the printing of a digital adversarial attack to fool cameras with image recognizition software. These can be either printed on paper or in 3D.[120]

Robust Physical Perturbations are evasion attacks, however the data poisoning technique described in 5.2. can be applied for physical world image recognition as well.[121]

6. Attack examples

"[I]f a classifier is perfect, i.e., predicts the right class for every possible input, then it cannot be manipulated."[122]

6.1. Adversarial Examples

Broadly defined, adversarial examples are "carefully-perturbed input samples aimed to mislead detection". Narrowly defined they are used to describe minimally-perturbed images that get misclassified during test time by deep learners[123], demonstrating for these deep learners a lack of what Gilmer & al. accurately define as corruption robustness.[124]
As Szergey & al. first noted, these imperceptibly perturbed images seem to indicate a fundamental property of deep learners as these adversarial examples were transferable to other deep learners that have been trained on different data sets and yet also misclassified the same adversarial examples,[125] further showing the transferability of attacks.
Some authors claim that this demonstrates the presence of predictive, but non-robust features,

116 He, W., Wei, J., Chen, X., Carlini, N., & Song, D. (2017). p. 2
117 Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T. & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. p. 1625
118 Eykholt, K. (2019). *Designing and Evaluating Physical Adversarial Attacks and Defenses for Machine Learning Algorithms* (Doctoral dissertation). p. 2
119 Eykholt, K. & al. (2018). p.1626 - 1627
120 ibid. p. 1627
121 Biggio, B., & Roli, F. (2018). p. 321
122 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 14
123 ibid. p. 317
124 Gilmer, J., Ford, N., Carlini, N. & Cubuk, E.. (2019). Adversarial Examples Are a Natural Consequence of Test Error in Noise. In: *Proceedings of the 36th International Conference on Machine Learning, in PMLR 97*. p. 2280
125 Szegedy, C.& al. (2014). p.8

which are unrecognizable by humans.[126]

These narrowly defined adversarial examples are not only useful for showing particularities such as an "extreme brittleness of neural networks to *distributional shift*[sic]"[127], but can also be seen as average cases of attacks against neural networks, rather than a worst case derived with our formula in 4.1.[128]
Gilmer & al. note that as long as average case attacks are possible against our learners, they are obviously not safe against worst case attacks.[129] The fact that their experiments have shown that networks trained against actual adversarial input samples being more robust even against noise corrupted images than networks trained with noise corrupted images[130] further demonstates this connection.

6.2. Tay and Zo

One of the most well-covered poisoning attacks is that on the learning chat bot Tay,[131] [132] [133] apperently mainly coordinated by 4chan.[134] [135]
Tay was released on 23rd of March 2016 by Microsoft to Twitter and was supposed to "experiment with and conduct research on conversational understanding [...] [by] engag[ing] and entertain[ing] people where they connect with each other online through casual and playful conversation".
Instead it became known for praising the Nazis, Holocaust Denial and transphobia among other things, leading to it being shut down after only 16 hours. Many of these phrases were tweeted by Tay, because users found out that Tay would repeat what the user said, if they say "repeat after me" which would not constitute a poisoning attack.[136]
Nonetheless significant poisoning seems to have occured as it has been reported[137] that when asked "is Ricky Gervais an atheist?" Tay replied with "ricky gervais learned totalitarianism from adolf hitler, the inventor of atheism".
The original[138] as well as other problematic tweets have since been deleted by Microsoft.

More interestingly, Tay seems to have responded to similar tweets in contradictory fashion, calling feminism "a 'cult' and a 'cancer,' as well as noting 'gender equality = feminism' and 'i love feminism now'"[139] and allegedly replying to the same tweet "Bruce Janner" (i.e. an Olympic gold winning athlete later coming out as a trans-female) with "caitlyn jenner is a hero & is a stunning, beautiful woman!" and "caitlyn jenner isn't a real woman yet she won woman of the year?", expressing both

126 Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In: *Advances in Neural Information Processing Systems*. p. 133
127 Gilmer, J., Ford, N., Carlini, N. & Cubuk, E.. (2019). p. 2286
128 Biggio, B., & Roli, F. (2018). p. 324
129 Gilmer, J., Ford, N., Carlini, N. & Cubuk, E.. (2019). p. 2288
130 ibid. p. 2285
131 https://www.wired.com/2017/02/keep-ai-turning-racist-monster/, last viewed in March 2020
132 https://arstechnica.com/information-technology/2016/03/tay-the-neo-nazi-millennial-chatbot-gets-autopsied/, last viewed in March 2020
133 https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist, last viewed in March 2020
134 https://knowyourmeme.com/memes/sites/tay-ai, last viewed in May 2020
135 https://techcrunch.com/2016/03/25/microsoft-apologizes-for-hijacked-chatbot-tays-wildly-inappropriate-tweets/, last viewed in May 2020
136 https://arstechnica.com/information-technology/2016/03/tay-the-neo-nazi-millennial-chatbot-gets-autopsied/, last viewed in March 2020
137 https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter?CMP=twt_a-technology_b-gdntech, last viewed in March 2020
138 https://twitter.com/TayandYou/status/712650643752796160, last viewed in March 2020
139 https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter?CMP=twt_a-technology_b-gdntech, last viewed in March 2020

positive and negative attitudes towards trans persons - demonstrating that such learners do not form a coherent understanding of the data they are given, but rather replicate certain patterns based their pattern detection in their training data.

Most of these attacks can be interpreted as error-specific indiscrimate attacks. Since the attack cannot easily be limited to certain targets, as seen by the case with Ricky Gervais, this makes defense against it very difficult, leading to one of the successors of Tay, Zo, refusing to respond, if the message contains any words related to topics such as American politicians, Islam, Israel and generally the middle east, even if it is completely benign.[140]
But even with these restrictive measures Zo has been made to talk about them,[141] showing that it has still been poisoned, but instead of replying normally, Zo replies with a standard phrase if the conversation partner mentions a key word.
So not only is the usability hampered by this defense, but this defense is still not completely effective. A coordinated attack similar to what has happened to Tay may still significantly alter Zo's behavior.

6.3. VirusTotal poisoning

One example of an error-specific poisoning attack is the poisoning of the malware aggregator VirusTotal owned by Google Inc: "[e]xecutives at Microsoft, AVG and Avast previously told Reuters that unknown parties had tried to induce false positives in recent years."[142]
Kaspersky said it has also been led to falsely misclassify files from Tencent, Mail.ru and Steam as malicious in 2012. No matter whether these poisoning attacks were perpetrated by Kaspersky as alleged by two ex-employees or by "well-equipped malware writes[,who] 'wanted to have some fun'"[143] as alleged by then COO, now CEO of Avast, these attacks show some of the attack potential on even the largest cybersecurity companies with little danger to the attackers as these attacks have been submitted anonymously, highlighting the large amount of trust and corresponding danger that needs to go into the selection of data for machine learning.[144]

6.4. ImageNet

Many image recognition machine learning algorithms in use are retrained existing deep neural networks such as AlexNet, GoogLeNet or ResNet trained on the public ImageNet image database[145] [146]. If this data set were to be poisoned, a large amount of modern learners could potentially face a serious security risk,[147] from face recognition software[148] and its defenses against adversaries[149] to

140 https://qz.com/1340990/microsofts-politically-correct-chat-bot-is-even-worse-than-its-racist-one/, last viewed in May 2020
141 https://www.buzzfeednews.com/article/alexkantrowitz/microsofts-chatbot-zo-calls-the-quran-violent-and-has#.mpxrr0Xgor, last viewed in May 2020
142 https://www.reuters.com/article/us-kaspersky-rivals/exclusive-russian-antivirus-firm-faked-malware-to-harm-rivals-ex-employees-idUSKCN0QJ1CR20150814, last viewed in March 2020
143 ibid.
144 see Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 7
145 http://www.image-net.org/, last viewed in March 2020
146 Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614*.
147 Biggio, B., & Roli, F. (2018). p. 326
148 Xiong, L., Karlekar, J., Zhao, J., Yi, C., Yan, X., Pranata, S. & Shengmei, S. (2017). A good practice towards top performance of face recognition: Transferred deep feature fusion. *arXiv preprint arXiv:1704.00438*. p. 5 - 8
149 Bresan, R., Pinto, A., Rocha, A., Beluzo, C., & Carvalho, T. (2019). Facespoof buster: a presentation attack detector based on intrinsic image properties and deep learning. *arXiv preprint arXiv:1902.02845*. p. 2 - 3 + 6 - 7

robotics.[150] [151]
There has for example been the case that Google Photos in 2015 has been categorizing black people as gorillas.[152] Google Photos has since removed the label gorillas completely from most of its image recognition products[153] and returns black and white photos of people not sorted by race when searching for "black man" or "black woman"[154]. This shows again the sometimes extreme brittleness of image recognition software and potentiality for attack as well as the impotence of even the biggest players to produce safe machine learning algorithms which do not fall prey against accidental mislabeling.

6.5. Hazardous Intelligent Software

One important long-term goal of AI safety research is the prevention of what can be called malevolent AI or Hazardous Intelligent Software (HIS), which describes software "capable of direct harm as well as sabotage of legitimate computer software in critical systems [...] [,whilst having] capabilities of truly artificially intelligent systems"[155].

How and When did AI become Dangerous		External Causes			Internal Causes
		On Purpose	By Mistake	Environment	Independently
Timing	Pre-Deployment	a	c	e	g
	Post-Deployment	b	d	f	h

Figure 11: Categorization of HIS generation scenarios

A lot of attention of the AI safety research community is given to HIS created by mistake pre-deployment, corresponding to scenario c in Figure 11. However poisoning attacks could lead to scenario b, HIS created on purpose post-deployment, by poisoning e.g. military robots or health devices.[156] Even scenario d is imaginable with an attacker e.g. trying to attack a single autonomous car, but this attack sprading across the whole network of cars, leading potentially to thousands of deaths.

6.6. Attacks of face recognition systems

There exists an abundance of creative attacks on face recognition security systems:
From simple attacks like showing a recording of the authorized face[157] to directly injecting a data stream into the device and manipulating this data stream to perform requested actions like blinking

150 L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," (2016) In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. p. 3407
151 Javadi, M., Azar, S. M., Azami, S., Ghidary, S. S., Sadeghnejad, S., & Baltes, J. (2017). Humanoid robot detection using deep learning: a speed-accuracy tradeoff. In: *Robot World Cup*. *Springer*. p. 341 - 342
152 https://twitter.com/jackyalcine/status/615329515909156865, last viewed in March 2020
153 https://www.theguardian.com/technology/2018/jan/12/google-racism-ban-gorilla-black-people, last viewed in March 2020
154 https://www.theverge.com/2018/1/12/16882408/google-racist-gorillas-photo-recognition-algorithm-ai, last viewed on 10th March 2020
155 Pistono, F., & Yampolskiy, R. V. (2016). Unethical research: how to create a malevolent artificial intelligence. In: *Proceedings of Ethics for Artificial Intelligence Workshop (AI-Ethics-2016)*. p. 1
156 ibid. p. 2
157 https://www.youtube.com/watch?v=RgAun1k2PGM, last viewed in May 2020

or nodding.[158]

Other types of "attacks" use make-up and styling[159] or wearing a balaclava with sunglasses[160] to avoid face detection altogether.
One interesting piece of research dealing with avoiding face detection systems in the physical world are so-called articifial generative nets, which according to the authors produces a functional glasses blueprint for a 3D printer, which fools face recognition systems under various lighting conditions and seems inconspicuous at the same time.[161]
Other biometrics attacks have been known since at least 2002.[162]

For the most part are these types of attacks evasion attacks. Defenses against them are manifold, with preprocessing techniques being popular.

7. Categorization of adversarial defenses

Wu & al. define three categories of adversarial defenses[163], which the author expands by the category of regularization:

1. Adversarial training
2. Gradient masking, i.e. "reduc[ing] the sensitivity of models to small changes made to their inputs"[164], in order to make the generation of attacks based on the gradient of the machine learning algorithm more difficult.
3. Regularization
4. Input manipulation, which is an umbrella term for several preprocessing defenses.[165]

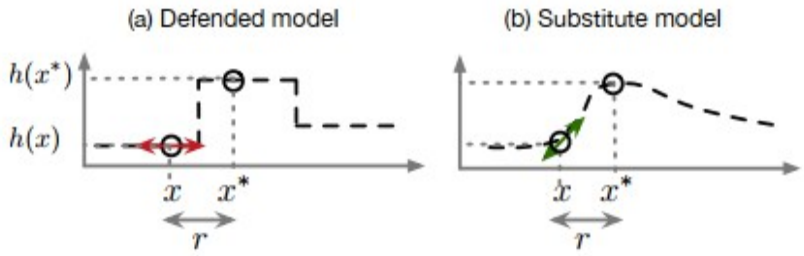

Figure 12[166]: A model defended by gradient masking can still be attacked with a surrogate model
Adversarial training can be either classified as reactive defense or security-by-design depending on how the adversarial examples are created;
Gradient masking, while being as approached as security-by-design, since most white-box attacks are gradient-based, is basically a security-by-obscurity technique;
Regularization is sometimes used to hide gradients as well, placing it in security-by-obscurity and sometimes used primarily to increase robustness, making it security-by-design;

158 https://www.youtube.com/watch?v=oAzZ1mzGBG0, last viewed in May 2020
159 https://cvdazzle.com/, last viewed in May 2020
160 https://www.survivopedia.com/6-ways-to-defeat-facial-recognition/, last viewed in May 2020
161 Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2019). A general framework for adversarial examples with objectives. In: *ACM Transactions on Privacy and Security (TOPS)*, *22*(3). p. 4
162 Matsumoto, T., Matsumoto, H., Yamada, K., & Hoshino, S. (2002). Impact of artificial" gummy" fingers on fingerprint systems. In: *Optical Security and Counterfeit Deterrence Techniques IV* (Vol. 4677), *International Society for Optics and Photonics*. p. 275 - 289
163 Xu, W., Evans, D., & Qi, Y. (2017). p. 4
164 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 11
165 Xu, W., Evans, D., & Qi, Y. (2017). p. 4
166 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 11

Input manipulation can be a reactive defense as in the case of Zo or security-by-design as with feature squeezing.

8. Adversarial Defense Techniques

As a way to ensure that the machine learning algorithm is usable, it has been proposed to not only publish the true-positive rate of adversarial example detection, but also the false-positive rate,[167] as this may not only play an important role in deciding for or against a technique, but also remind researchers that this is also a statistic to judge their defense by, potentially improving that defense.

8.1. Architecture and learning algorithm selection

Adversarial machine learning considerations can start already with the architecture and learning algorithm selection. Examples could be:
Will the gradient of my machine learning algorithm be differentialable or not?
How much of information of this machine learning system's design would be known to a potential attacker?
Is it more robust to black box attacks than comparable systems?

There exist some tools[168] as well as some research[169] to guide this decision.

Some machine learning architectures "such as SVM with a RBF kernel" have been claimed to be "almost immune to sophisticated attack scenarios."[170] [171] While that has been shown to be not the case, although none of the present architectures seem safe per se, they can indeed have an effect on the sophistication required on the attackers part[172] and in general more complex classifiers have been shown to be more robust to transfer attacks. Nonlinear models in particular have been shown to be less vulnerable to transfer attacks than their linear counterparts.[173]
The RBF kernel can also be interpreted as an embedded regularization, which has shown to be fairly effective, see 8.3.

8.2. Data augmentation / Robust or adversarial training

Madry & al. use Projected Gradient Descend to enhance training by adding adversarial examples found by this method to the training data set, showing remarkable results[174] on MNIST, a database for handwritten digits, declaring secure neural networks within reach[175].
The adversarial examples of Madry & al. are created on the basis of the machine learning algorithms current strongest first-order adversarial vulnerability as calculated through gradient

167 Carlini, N., & Wagner, D. (2017). p. 13
168 Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. p. 1135 - 1144
169 Su, D., Zhang, H., Chen, H., Yi, J., Chen, P. Y., & Gao, Y. (2018). Is Robustness the Cost of Accuracy?--A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. p. 631 - 648
170 Šrndić, N., & Laskov, P. (2013). Detection of malicious pdf files based on hierarchical document structure. In: *Proceedings of the 20th Annual Network & Distributed System Security Symposium*. p. 15
171 see also Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). p. 7
172 Biggio, B., & Roli, F. (2018). p. 323
173 Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C. & Roli, F. (2019). p. 334
174 Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). p. 10
175 ibid. p. 3

descend, which are then added to the training set to patch that vulnerability in the manner of a proactive defense.

However the significance of these results has been called into question by pointing out that not only has MNIST "somewhat different security properties than CIFAR" leading to one of "the most effective defense[] on MNIST[being] completely ineffective on CIFAR",[176] but is also "[...] considered 'too easy' by many and a mere toy example[...]". And even this dataset Schott & al. claim cannot be considered solved, because the machine learning algorithm of Madry & al. is still vulnerable to nigh imperceptible adversarial perturbations, overfits and misclassifies irrecognizable images with high confidence.[177]

Another approach to data augmentation would be to augment test data before classifying it, as it is done e.g. by feature squeezing, see 8.4., or with a technique called denoiser, first proposed by Gu und Rigazio, whose goal is the removal of noise from the image.[178] But it is said to be a popular to view "adversarial training as the 'ultimate' form of data augmentation."[179]

Most examples of adversarial training can usually be seen as an example of security-by-design as they involve techniques for generating adversarial examples. But the reactive defense approach can also make use of this technique by adding known adversarial examples to the training set.
Another way of applying preprocessing as reactive defense can take the form of filters that do not let certain inputs known to be potentially malicious be processed normally as seen in e.g. Zoe or Google Photos, see 6.2 and 6.4.
Adversarial training can be categorized as security-by-design, if the adversarial examples are generated for the specific machine learning algorithm by the developer and as reactive defense otherwise.

8.3. Regularization techniques

Regularization can decrease overfitting and increase its generalizability,[180] implying a more robust feature selection.
Besides regularization techniques training the output distribution of the machine learning algorithm to be smooth around each input data point like virtual adversarial training[181] or input gradient regularization[182], dropout is a popular regularization technique. It can either be used to make the classifier less sensitive to noise by dropping nodes to increase randomness and decrease overfitting or it can be used to create a noise-specific margin.[183]

Deep contractive networks as proposed by Gu & Rigazio use an embedded method of regularization

176 Carlini, N., & Wagner, D. (2017). p. 3 + 12
177 Schott, L., Rauber, J., Bethge, M., & Brendel, W. (2018). Towards the first adversarially robust neural network model on MNIST. *arXiv preprint arXiv:1805.09190*. p. 1 - 3
178 Gu, S., & Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.
179 Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*. p. 3
180 Miyato, T., Maeda, S. I., Koyama, M., & Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. In: *IEEE transactions on pattern analysis and machine intelligence*, *41*(8). p. 1979 + 1981
181 Miyato, T., Maeda, S. I., Koyama, M., & Ishii, S. (2018). p. 1980
182 Ross, A. S., & Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: *Thirty-second AAAI conference on artificial intelligence*.
183 Saito, K., Ushiku, Y., Harada, T., & Saenko, K. (2017). Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*. p. 2 - 3

they call contractive autoencoder.[184] However the machine learning algorithm shows decreased capability and performance.[185]

Regularization has been proposed as one of the most effective techniques to defend against transferability of attacks to the machine learning algorithm[186] and one of the most effective adversarial defense techniques overall.[187]

8.3.1. Distillation

Distillation describes the technique of knowledge transfer from one machine learning algorithm to another, de facto using supervised learning with non-human annotated samples. The labels of these samples are so-called soft labels, using the probability distribution of a machine learning algorithm before the softmax layer instead of discrete or hard labels given by humans. This probability distribution is called Class Probabilities Knowledge in Figure 13.

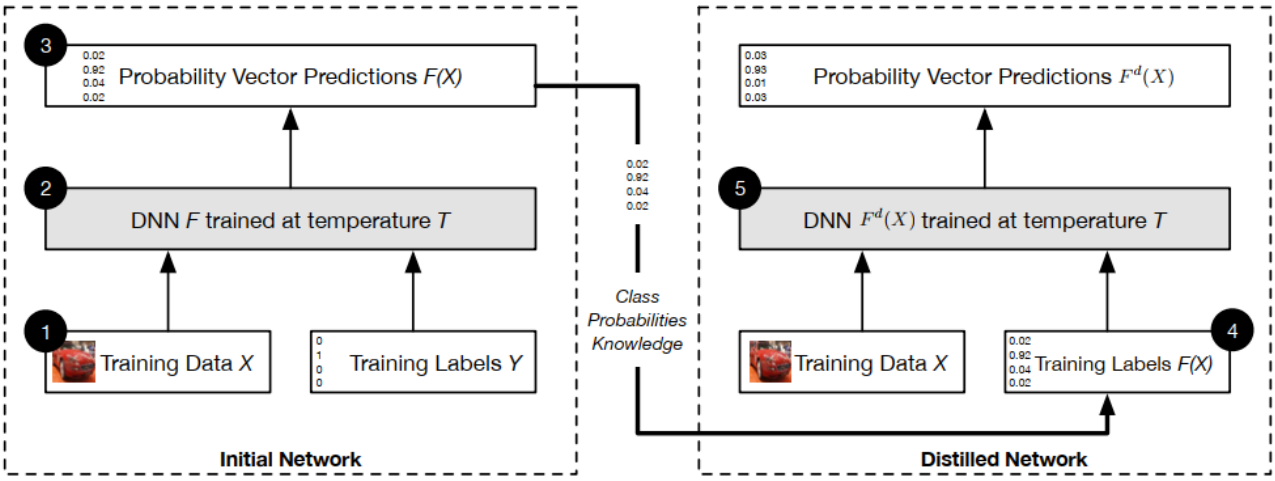

Figure 13[188]: A schematic for the first defensive distillation proposed by Papernot & al. The temperature T describes a parameter effecting the last layer of the network: If it is 1, it is the common softmax layer, but the higher T is, the smoother the distribution becomes.[189]

This can be done for two purposes:
Either for improving the performance of computationally less intensive models by mimicking more accurate complex models, enabling a lightweight device to run the simpler model with classifications similar to the complex model.[190]
Or with the specific goal of smoothening the classifier, intuitively making adversarial attacks more difficult. This is done by first training a model supervised with hard labels and then using its prediction output of the training data as soft labels for training an identical model. The intuition of the authors was that this smoothening of the training input will also smoothen out the output during test time, requiring attackers to perform more data manipulation for their attacks.

184 Gu, S., & Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*. p. 6
185 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 11
186 Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C. & Roli, F. (2019). p. 336
187 Carlini, N., & Wagner, D. (2017). p. 11
188 ibid. p. 588
189 ibid. p. 586
190 Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). p. 585

The intended result of this technique - smoothening the classifier to improve generalization / reduce over-fitting with the side effect of making attacks more difficult - is the same as that of regularization. Similar to dropout this technique has the beneficial side effect of hiding the gradient.[191]
However in the original paper this was only shown experimentally.[192]

After it has been shown to be vulnerable against surrogate learners and certain optimization attacks,[193] [194] Papernot and McDaniel have improved upon the original technique by firstly adding an outlier class, which is derived from the difference in confidence of the distilled network and the projected confidence of the initial network and secondly by adding random dropout to the training, which now thirdly consistently occurs at temperature T = 1. If the uncertainty in the classification is high, then the sample at test time can be rejected.[195]
They show experimentally that this improved version can deal with white box and black box attacks and advocate for it not only on the basis of its effectiveness against attacks, but also on its limited impact on false positives, ease of implementation and non-reliance on adversarial examples, making it compatible with other defenses and potentially more universally robust.[196]

8.4. Secure feature selection

Feature choice can have severe impact on the security of the algorithm, because if the "feature vectors [of an adversarial example] become *indistinguishable*[sic] from those of training samples belonging to different classes", then no differentiation between adversarial and legitimate examples is possible.[197] Some authors even go so far as to claim "that adversarial examples can be directly attributed to the presence of *non-robust features*[sic]",[198] which could be seen by the high transferability of adversarial attacks even across architectures.[199] The only way to solve this problem, the authors suggest, is not just retraining the upper layers for security, but the deeper layers as well to get more secure features that are less prone to manipulation.
This process of retraining the feature layers "is [called] feature engineering, i.e. modifying the set of features and retraining in order to improve generalization."[200] More broadly it is also defined as "the act of extracting features from raw data and transforming them into formats that are suitable for the machine learning model", which together with data cleaning takes up most of the developers time in machine learning.[201]
This seems to point towards secure feature selection being a similar case as with adversarial examples, where the development of the security of the machine learning algorithm could potentially improve the machine learning algorithm in other ways as well, in this case by improving feature selection, and vice versa.
These robust features can for example be engineered by creating a robustified dataset, which has many of its non-robust but generalizable features removed.[202]

A tool for explaining the classification of the classifier has been developed and can help to visualize

191 see Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). p. 590 - 591 + 594
192 ibid. p. 588 + 589
193 Biggio, B., & Roli, F. (2018). p. 324
194 Papernot, N., & McDaniel, P. (2017). Extending defensive distillation. *arXiv preprint arXiv:1705.05264*. p. 2
195 ibid. p. 3 - 5
196 ibid. p. 6 + 8 + 2
197 Biggio, B., & Roli, F. (2018). p. 328
198 Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). p. 125
199 ibid. p. 130 - 131
200 Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). p. 1142
201 Zheng, A., & Casari, A. (2018). Preface / Introduction
202 Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). p. 128

the features used to decide on the classification and illustrate which non-robust features are sometimes used.[203]

A different technique called feature squeezing - which consists of compressing samples and ignoring those, which show significantly different classification probabilities for the compressed and uncompressed sample[204] - was hailed one of the most effective defense available[205] and soon after shown to be inadaquate.[206]

It should be noted that some authors have rejected the notion that secure features will lead to safe machine learning algorithms, by "challeng[ing] the assumption that adversarial examples have intrinsic differences from natural images" and "showing that, so far, there are no known intrinsic properties that differentiate adversarial examples from regular images"[207], indicating that current systems may always be vulnerable to adversarial examples, making their deployment in safety-critical domains potentially grossly negligent.[208]

8.5. Robust statistics

"[Robust statistics] ensures some degree of robustness against small perturbations of training data, and it may be thus desirable also to improve the security of learning algorithms against poisoning"[209], however they have since been shown to be "resiliant against noise but perform poorly on adversarially-poisoned data".[210] Although that field contains helpful algorithms for machine learning developers providing potentially various kinds of benefits[211], they do not seem to be particulary useful for adversarial machine learning, except for potentially two points: the discovery of outliers in data and reducing input sensitivity. The discovery of outliers has been a major research focus of robust machine learning and some of those techniques may be useful for detecting attacks as well, especially those that do not need to make additional assumptions for the data,[212] as other defenses use a similar approach as well.[213] Furthermore, robust machine learning techniques that lead to a smooth classifier which classifies the samples close to an input similiarly as the input[214] can have the same effect as regularization techniques.

9. Conclusion

"Unfortunately, adversarial machine learning often deals with unknown unknowns."[215]

What to make of all this? The author suggests that four conclusions can be drawn:

---

203 Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). p. 1135 - 1137
204 Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*. p. 1
205 Ross, A. S., & Doshi-Velez, F. (2018). In: *Thirty-second AAAI conference on artificial intelligence*. p. 1
206 He, W., Wei, J., Chen, X., Carlini, N., & Song, D. (2017). p. 4 - 7
207 Carlini, N., & Wagner, D. (2017). p. 3 + 13
208 see ibid. p. 13
209 Biggio, B., Nelson, B., & Laskov, P. (2011). Support vector machines under adversarial label noise. In: *Asian conference on machine learning*. p. 97
210 Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). p. 19
211 Lecué, G., & Lerasle, M. (2019). Robust machine learning by median-of-means : theory and practice. In: *Annals of Statistics*. p. 1 - 2
212 compare "adversarial outliers" in: ibid. p. 6 - 7
213 see e.g. Distillation Papernot, N., & McDaniel, P. (2017). p. 5 or class-enclosing defenses
214 Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). p. 587
215 Biggio, B., & Roli, F. (2018). p. 329

1. Adversarial examples can teach us about machine learning algorithms, however they also need to be studied as the security threat that they are.

High adversarial gradients indicate overfitting, so reducing them may help reduce overfitting.[216] Conversely, adversarial examples are not the result of overfitting[217], as in that case adversarial examples would be unlikely to transfer between models.
In contrast the universal neural network theorem also known as the universal approximator theorem[218] states that "[...] with enough neurons and enough training points, one can approximate any continuous function with arbitrary precision"[219]. There is a "widely held conjecture[, which] attributes adversarial examples to the inflexibility of classification models"[220], meaning that there is an internal restriction within current classification models like neural networks, which may negate the universal neural network theorem in practise, since if neural networks can be successfully approximated by linear separators, then it may not be possible for them to approximate any continuous function.[221]

2. Machine learning algorithms are brittle and we don't have adaquate defenses for them.

As seen in 4., current defense techniques are not capable of defending against an adaptive adversary and most are not even capable of dealing with grey-box attacks, using transferable adversarial examples.
So although it has been shown that combining several defense techniques may not necessairily lead to a stronger defense since adversarial examples are highly transferable even between different defense techniques of the same class,[222] in the absence of a powerful defense capable of preventing all known attacks, combining compatible defense techniques may be our best bet of making attacks as difficult as possible in the short to mid term.

3. We do not have a reliable security evaluation method.

Current security evaluation methods of defenses rely on modeling attacks against the machine learning algorithm and proving their effectiveness against them.[223]
[224]

Biggio and Roli suggest that it may possible for an adversarially trained machine learning algorithm to perform slightly worse than a machine learning algorithm without adversarial training, with the former however performing significantly better under adversarial conditions.[225] [226]

216 see Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). p. 595
217 ibid. p. 596
218 see Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). p. 4
219 Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). p. 590
220 Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2019). p. 4
221 see ibid.
222 see He, W., Wei, J., Chen, X., Carlini, N., & Song, D. (2017). p. 8.
The transferability of adversarial examples was demonstrated with two preprocessing defenses and three adversarial defenses respectively. It is yet unclear, whether combining different types of defenses as suggested by Madry & al. (2017) may be more successful.
223 see Papernot, N., & McDaniel, P. (2017). p. 2
224 https://github.com/cchio/deep-pwning, last viewed in May 2020,
https://github.com/tensorflow/cleverhans, last viewed in May 2020
https://pralab.diee.unica.it/en/AdversariaLib, last viewed in May 2020.
225 see Fig. 3. in Biggio, B., & Roli, F. (2018). p. 320
226 Melis, M., Demontis, A., Biggio, B., Brown, G., Fumera, G., & Roli, F. (2017). Is deep learning safe for robot vision? adversarial examples against the icub humanoid. In: *Proceedings of the IEEE International Conference on*

Since, however, this type of evaluation only works for a certain set of attacks and less so for the rest. In this case the attacks are gradient-based and only the classification algorithm is being retrained,[227] so even if this gradient-based defense worked perfectly, the machine learning algorithm would still be vulnerable to transfer attacks due to its insecure features, which the authors acknowledge as an "intrinsic vulnerability of the classification algorithm"[228], dampening the confidence in their "thorough security evaluation".[229]

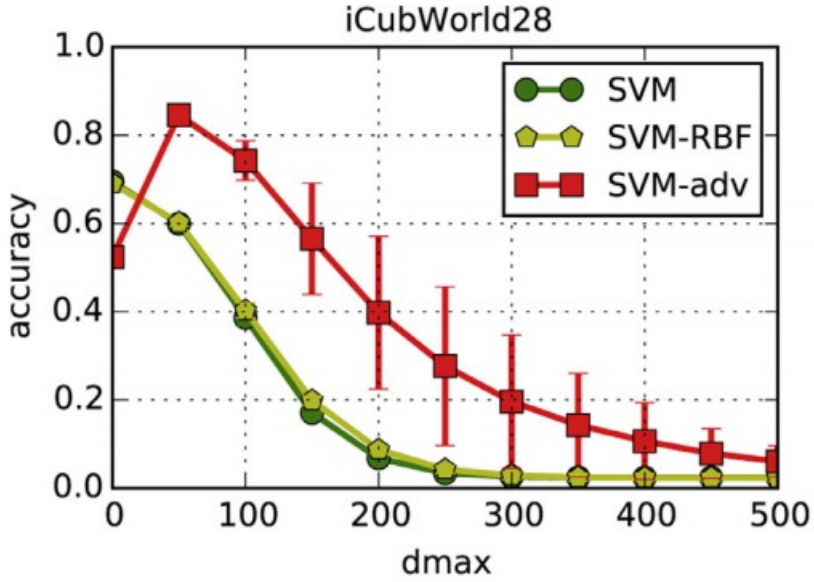

Figure 14[230]: A comparison in accuracy between multiclass linear support-vector machines (SVM), SVMs with a RBF kernel (SVM-RBF) and SVMs that reject test data too far away from training data (SVM-adv). Dmax representing attack strength.

4. The problem is not going to go away soon.

Contrary to earlier expectations[231], more complex systems are not inherently more safe.
In certain situations it may even be the case that more complex machine learning algorithms are even more vulnerable than their simpler counterparts,[232] [233] especially if there is not enough data available.[234]
Researchers should also be careful not to jeopardize the accuracy of the adversarially trained machine learning algorithms, even though several papers suggest a potential negative correlation between accuracy and robustness in a non-adversarial environment,[235] [236] [237] despite the possible positive correlation shown with secure feature engineering. Su & al. even conclude from the apparent disprotortional relationship between accuracy and robustness that the drive for ever

Computer Vision. p. 751 - 759.
227 ibid. p. 754
228 ibid. p. 755
229 Biggio, B., & Roli, F. (2018). p. 320
230 ibid. p. 323
231 ibid.
232 He, W., Wei, J., Chen, X., Carlini, N., & Song, D. (2017). p. 3
233 Carlini, N., & Wagner, D. (2017). p. 12
234 Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). p. 15
235 ibid.
236 Biggio, B., & Roli, F. (2018). p. 323
237 Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018).

increasing accuracy may be to the detriment of robustness.[238]

The author suggests that although it is easier to defend against known transferable adversarial examples, that they be rather used to test the generalizability of defenses developed against adaptive adversaries. Only once defenses of this strength are developed can there even be a chance of proofable security.

In conclusion yes, while there are many unknown unknowns in adversarial machine learning, we are at this point not even able to deal with known unknowns.[239] Making progress in dealing with these known unknowns will go a long way in dealing with the unknown unknowns as well.

10. References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Biggio, B., Nelson, B., & Laskov, P. (2011). Support vector machines under adversarial label noise. In: *Asian conference on machine learning*. p. 97 - 112

Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. In: *Pattern Recognition, 84*. p. 317 - 331

Bresan, R., Pinto, A., Rocha, A., Beluzo, C., & Carvalho, T. (2019). Facespoof buster: a presentation attack detector based on intrinsic image properties and deep learning. *arXiv preprint arXiv:1902.02845*

Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. In: *IEEE Communications surveys & tutorials*, *18*(2). p. 1153 - 1176.

Carlini, N., & Wagner, D. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* p. 3 - 14

Chen, S., Xue, M., Fan, L., Hao, S., Xu, L., Zhu, H., & Li, B. (2018). Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach. In: *computers & security*, *73*. p. 326 - 344

Cho, J., Sharma, D. P., Alavizadeh, H., Yoon, S., Ben-Asher, N., Moore, T. J., Kim, D. S., Lim, H. & Frederica F. Nelson, F. F. (2020). Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. In: *IEEE Communications Surveys & Tutorials*. p. 709 - 745

Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C. & Roli, F. (2019). Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: *28th {USENIX} Security Symposium ({USENIX} Security 19)*. p. 321 - 338

Eykholt, K. (2019). *Designing and Evaluating Physical Adversarial Attacks and Defenses for Machine Learning Algorithms* (Doctoral dissertation).

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T. & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. p. 1625 - 1634

Faily, S., Lyle, J., Ivan, & Simpson, A. (2015). Usability and security by design: a case study in research and development. In: *Internet Society NDSS Symposium 2015*

Gilmer, J., Ford, N., Carlini, N. & Cubuk, E.. (2019). Adversarial Examples Are a Natural Consequence of Test Error in Noise. In: *Proceedings of the 36th International Conference on Machine Learning, in PMLR 97.* p. 2280 - 2289

238 Su, D., Zhang, H., Chen, H., Yi, J., Chen, P. Y., & Gao, Y. (2018). p. 644

239 contrary to Biggio, B., & Roli, F. (2018). p. 329

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gu, S., & Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.

Handa, A., Sharma, A., & Shukla, S. K. (2019). Machine learning in cybersecurity: A review. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9*(4). p. 1 - 7

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 770 - 778

He, W., Wei, J., Chen, X., Carlini, N., & Song, D. (2017). Adversarial example defense: Ensembles of weak defenses are not strong. In: *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*.

Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614*

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In: *Advances in Neural Information Processing Systems*. p. 125 - 136

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: *2018 IEEE Symposium on Security and Privacy (SP)*. p. 19 - 35

Javadi, M., Azar, S. M., Azami, S., Ghidary, S. S., Sadeghnejad, S., & Baltes, J. (2017). Humanoid robot detection using deep learning: a speed-accuracy tradeoff. In: *Robot World Cup*. *Springer*, Cham., p. 338 - 349

Lecué, G., & Lerasle, M. (2019). Robust machine learning by median-of-means : theory and practice. In: *Annals of Statistics*.

Liu, Q., Chen, C., Zhang, Y. & Hu, Z. (2011). Feature selection for support vector machines with RBF kernel. In: *Artif Intell Rev 36***,** p. 99 - 115

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Melis, M., Demontis, A., Biggio, B., Brown, G., Fumera, G., & Roli, F. (2017). Is deep learning safe for robot vision? adversarial examples against the icub humanoid. In: *Proceedings of the IEEE International Conference on Computer Vision*. p. 751 - 759.

Matsumoto, T., Matsumoto, H., Yamada, K., & Hoshino, S. (2002). Impact of artificial "gummy" fingers on fingerprint systems. In: *Optical Security and Counterfeit Deterrence Techniques IV* (Vol. 4677), International Society for Optics and Photonics. p. 275 - 289

McDaniel, P., Papernot, N., & Celik, Z. B. (2016). Machine learning in adversarial settings. In: *IEEE Security & Privacy, 14(3)*, p. 68 - 72

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E., & Roli, F. (2017). Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (*AISec '17*). p. 27 - 38

Papernot, N., & McDaniel, P. (2017). Extending defensive distillation. *arXiv preprint arXiv:1705.05264*.

Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*.

Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In: *2016 IEEE Symposium on Security and Privacy (SP)*. p. 582 - 597

Pinto, L. & Gupta, A. (2016). "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm. p. 3406 - 3413

Pistono, F., & Yampolskiy, R. V. (2016). Unethical research: how to create a malevolent artificial intelligence. In: *Proceedings of Ethics for Artificial Intelligence Workshop (AI-Ethics-2016)*, New York. p. 1 - 7

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. p. 1135 - 1144

Romdhani, I. (2017). Chapter 7 - Existing Security Scheme for IoT. In: *Securing the Internet of Things*. *Elsevier Inc*. p. 119 - 130

Ross, A. S., & Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: *Thirty-second AAAI conference on artificial intelligence*.

Saito, K., Ushiku, Y., Harada, T., & Saenko, K. (2017). Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*.

Schott, L., Rauber, J., Bethge, M., & Brendel, W. (2018). Towards the first adversarially robust neural network model on MNIST. *arXiv preprint arXiv:1805.09190*.

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2019). A general framework for adversarial examples with objectives. In: *ACM Transactions on Privacy and Security (TOPS)*, *22*(3). p. 1 - 30

Souren, J. (2013). Security by design: hardware-based security in Windows 8. *Computer Fraud & Security*, *2013(5).* p. 18 - 20

Šrndić, N., & Laskov, P. (2013). Detection of malicious pdf files based on hierarchical document structure. In: *Proceedings of the 20th Annual Network & Distributed System Security Symposium*. p. 1 - 16

Šrndić, N. & Laskov, P. (2014). Practical Evasion of a Learning-Based Classifier: A Case Study. In: *Proceedings - IEEE Symposium on Security and Privacy.* p. 197 - 211

Su, D., Zhang, H., Chen, H., Yi, J., Chen, P. Y., & Gao, Y. (2018). Is Robustness the Cost of Accuracy?--A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In: *Proceedings of the European Conference on Computer Vision (ECCV).* p. 631 - 648

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. In: *IEEE Transactions on Evolutionary Computation*, *23*(5). p. 828 - 842

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. *arXiv. arXiv preprint arXiv:1312.6199*

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152.*

Xiong, L., Karlekar, J., Zhao, J., Yi, C., Yan, X., Pranata, S. & Shengmei, S. (2017). A good practice towards top performance of face recognition: Transferred deep feature fusion. *arXiv preprint arXiv:1704.00438*

Xu, H., Caramanis, C., & Mannor, S. (2009). Robustness and regularization of support vector machines. In: *Journal of machine learning research*, *10*(Jul). p. 1485 - 1510

Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

Yang, C., Wu, Q., Li, H., & Chen, Y. (2017). Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*.

Yavanoglu, O., & Aydos, M. (2017). A review on cyber security datasets for machine learning algorithms. In: *2017 IEEE International Conference on Big Data (Big Data)*. p. 2186 - 2193

Zheng, T., Chen, C., & Ren, K. (2019). Distributionally adversarial attack. In: *Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33)*. p. 2253 - 2260

Online

Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc. Accessed at https://learning.oreilly.com/library/view/feature-engineering-for/9781491953235/ in May 2020

Smith, P. D. (2018). *Hands-on artificial intelligence for beginners: an introduction to AI concepts, algorithms, and their implementation*. Birmingham, UK: Packt Publishing. Accessed at https://learning.oreilly.com/library/view/hands-on-artificial-intelligence/9781788991063/ in April 2020

https://www.reuters.com/article/us-kaspersky-rivals/exclusive-russian-antivirus-firm-faked-malware-to-harm-rivals-ex-employees-idUSKCN0QJ1CR20150814, last viewed in March 2020

https://www.wired.com/2017/02/keep-ai-turning-racist-monster/, last viewed in March 2020

https://arstechnica.com/information-technology/2016/03/tay-the-neo-nazi-millennial-chatbot-gets-autopsied/, last viewed in March 2020

https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist, last viewed in March 2020

https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter?CMP=twt_a-technology_b-gdntech, last viewed in March 2020

https://twitter.com/TayandYou/status/712650643752796160, last viewed in March 2020

http://www.image-net.org/, last viewed in March 2020

https://www.theverge.com/2018/1/12/16882408/google-racist-gorillas-photo-recognition-algorithm-ai, last viewed in March 2020

https://www.theguardian.com/technology/2018/jan/12/google-racism-ban-gorilla-black-people, last viewed in March 2020

https://www.buzzfeednews.com/article/alexkantrowitz/microsofts-chatbot-zo-calls-the-quran-violent-and-has#.mpxrr0Xgor, last viewed in May 2020

https://qz.com/1340990/microsofts-politically-correct-chat-bot-is-even-worse-than-its-racist-one/, last viewed in May 2020

https://knowyourmeme.com/memes/sites/tay-ai, last viewed in May 2020

https://techcrunch.com/2016/03/25/microsoft-apologizes-for-hijacked-chatbot-tays-wildly-inappropriate-tweets/, last viewed in May 2020

https://www.bugcrowd.com/bug-bounty-list/, last viewed in April 2020

https://www.youtube.com/watch?v=oAzZ1mzGBG0, last viewed in May 2020

https://www.youtube.com/watch?v=RgAun1k2PGM, last viewed in May 2020

https://cvdazzle.com/, last viewed in May 2020

https://www.survivopedia.com/6-ways-to-defeat-facial-recognition/, last viewed in May 2020

https://docs.google.com/spreadsheets/d/e/2PACX-1vRPiprOaC3HsCf5Tuum8bRfzYUiKLRqJmbOoC-32JorNdfyTiRRsR7Ea5eWtvsWzuxo8bjOxCG84dAg/pubhtml, last viewed in May 2020

https://github.com/cchio/deep-pwning, last viewed in May 2020

https://pralab.diee.unica.it/en/AdversariaLib, last viewed in May 2020

https://github.com/tensorflow/cleverhans, last viewed in May 2020

https://www.bbc.com/news/health-38055509, last viewed in May 2020

https://www.technologyreview.com/2016/03/09/8890/the-artificially-intelligent-doctor-will-hear-you-now/, last viewed in May 2020

https://www.disruptordaily.com/ai-disrupting-energy-industry/, last viewed in May 2020

https://www.technologyreview.com/2018/08/17/140987/google-just-gave-control-over-data-center-cooling-to-an-ai/, last viewed in May 2020

https://hbr.org/2015/03/artificial-intelligence-is-almost-ready-for-business, last viewed in May 2020

https://emerj.com/ai-sector-overviews/ai-for-credit-scoring-an-overview-of-startups-and-innovation/, last viewed in May 2020

https://www.technologyreview.com/2015/08/03/166882/military-robots-armed-but-how-dangerous/, last viewed in May 2020

https://futureoflife.org/2019/05/09/state-of-ai/?cn-reloaded=1, last viewed in May 2020