Article

Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey

Andrew McCarthy*, Essam Ghadafi, Panagiotis Andriotis and Phil Legg*

Computer Science Research Centre, University of the West of England, Bristol, BS16 1QY, UK
* Correspondence: andrew6.mccarthy@uwe.ac.uk

Abstract: Machine learning has become widely adopted as a strategy for dealing with a variety of cybersecurity issues, ranging from insider threat detection to intrusion and malware detection. However, by their very nature, machine learning systems can introduce vulnerabilities to a security defence whereby a learnt model is unaware of so-called adversarial examples that may intentionally result in mis-classification and therefore bypass a system. Adversarial machine learning has been a research topic for over a decade and is now an accepted but open problem. Much of the early research on adversarial examples has addressed issues related to computer vision, yet as machine learning continues to be adopted in other domains, then likewise it is important to assess the potential vulnerabilities that may occur. A key part of transferring to new domains relates to functionality-preservation, such that any crafted attack can still execute the original intended functionality when inspected by a human and/or a machine. In this literature survey, our main objective is to address the domain of adversarial machine learning attacks and examine the robustness of machine learning models in the cybersecurity and intrusion detection domains. We identify the key trends in current work observed in the literature, and explore how these relate to the research challenges that remain open for future works. Inclusion criteria were: articles related to functionality-preservation in adversarial machine learning for cybersecurity or intrusion detection with insight into robust classification. Generally, we excluded works that are not yet peer-reviewed; however, the authors include some significant papers that make a clear contribution to the domain. There is a risk of subjective bias in the selection of non-peer reviewed articles; however, this is mitigated by co-author review. We selected the following databases with a sizeable computer science element to search and retrieve literature: IEEE Xplore, ACM Digital Library, ScienceDirect, Scopus, SpringerLink,Google Scholar. The literature search was conducted upto January 2022. We have striven to ensure a comprehensive coverage of the domain to the best of our knowledge. We have performed systematic searches of the literature, noting our search terms and results, and following up on all materials that appear relevant and fit within the topic domains of this review. This research was funded by the Partnership PhD scheme at the University of the West of England in collaboration with Techmodal Ltd.

Keywords: cybersecurity; adversarial machine learning; machine learning; intrusion detection; functionality-preservation

Citation: McCarthy, A.; Ghadafi, A.; Andriotis, P.; Legg, P. Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey. *Preprints* **2022**, *1*, 0. https://doi.org/

1. Introduction

Machine Learning (ML) has become widely adopted as a strategy for dealing with a variety of cybersecurity issues. Cybersecurity domains particularly suited to ML include: intrusion detection and prevention [1], network traffic analysis [2], malware analysis [3] [4], user behaviour analytics [5], insider threat detection [6], social engineering detection [7], spam detection [8], detection of malicious social media usage [9], health misinformation [10], climate misinformation [11], and more generally "Fake News" [12]. Critically ML systems are increasingly trusted within cyber physical systems [13] such as power stations, factories, and oil and gas industries. In such complex physical environments, the potential damage that could be caused by a vulnerable system might even be life threatening [14]. Despite our reliance and trust in ML systems, the inherent nature of machine learning -

 doi:10.20944/preprints202202.0099.v1

learning to identify patterns - is in itself a potential attack vector for adversaries wishing to circumvent ML-based system detection processes. Adversarial examples are problematic for many ML algorithms and models including Random Forests (RF) and Naive Bayes (NB) Classifiers; however we focus on Artificial Neural Networks and particularly Deep Neural Networks. Artificial Neural Networks (ANN) are inspired by the network of neurons in the human brain. ANNs are useful because they can generalize from a finite set of examples, essentially mapping a large input space (infinite for continuous inputs) to a range of discrete outputs. Unfortunately, in common with other ML algorithms, neural networks are vulnerable to attacks using carefully crafted perturbations to inputs, including evasion and poisoning attacks. In recent work, carefully crafted inputs described as "adversarial examples" are considered possible in ANN because of these inherent properties that exist within neural networks [15], such as:

1. The semantic information of the model is held across the model and not localised to specific neurons.
2. Neural Networks learn input-output mappings that are discontinuous (and discontiguous).

These properties mean that even extremely small perturbations of an input could cause a neural network to provide a misclassified output. Given that neural networks have these properties, we reasonably expect our biological neural networks to suffer misclassifications, and/or to have evolved mitigations. Human brains are more complex than current artificial neural networks yet suffer a type of misclassification (illusory perception), in the form of face pareidolia [16] [17]. This strengthens the case that the properties of neural networks are the source of adversarial examples (AE). In cybersecurity related domains it has been seen how adversaries exploit adversarial examples, using carefully-crafted noise to evade detection through misclassification [18] [19].

In this way, an adversarial arms race exists between adversaries and defenders. The recent SolarWinds supply chain attack [20] [21] identified in December 2020 indicates the reliance that organisations have on intrusion detection software, and the presence of Advanced Persistent Threats (APTs) with the expertise and resources to attack organisations' network defenses. Adversarial machine learning is a critical area of research. If not addressed, there is increasing potential for novel attack strategies that seek to exploit the inherent weaknesses that exist within machine learning models. For this reason, this survey addresses the issues related to the robustness of machine learning models against adversarial attacks across the cybersecurity domain, where problems of functionality-preservation are recognized. While we use a case study of a network-based intrusion detection system (NIDS), these issues might be applicable in other areas where ML systems are used. We focus on papers detailing adversarial attacks and defenses. Attacks are further classified by attack type, attack objective, domain, model, knowledge required, and constraints. Defenses are further categorised by defense type, domain, and model. In the domain of network traffic analysis, adversaries need to evade detection methods. A suitable network firewall will reject adversarial traffic and malformed packets while accepting legitimate traffic. Therefore, successful adversarial examples must be crafted to comply with domain constraints such as those related to the transmission control protocol/internet protocol (TCP-IP) stack. Moreover, adversaries wish to preserve the functionality of their attacks. A successful attack must not lose functionality at the expense of evading a classifier. The essence of a simple adversarial attack is that a malicious payload evades detection by masquerading as benign. We refer to this characteristic as *functionality-preserving*. Compared to domains such as computer vision whereby the image modification is only to fool human vision sensors, adversarial attacks in other domains are significantly more challenging to fool both a human and/or system-based sensor.
The major contributions of this paper are:

 doi:10.20944/preprints202202.0099.v1

- We conduct a survey of the literature to identify the trends and characteristics of published works on adversarial learning in relation to cybersecurity, addressing both attack vectors and defensive strategies.
- We address the issue of functionality-preservation in adversarial learning in contrast to domains such as computer vision, whereby a malformed input must suitably fool a system process as well as a human user such that the original functionality is maintained despite some modification.
- We summarise this relatively-new research domain to address the future research challenges associated with adversarial machine learning across the cybersecurity domain.

The remainder of this paper is structured as follows: Section 2 provides an overview of other important surveys; Section 3 discusses background material; Section 4 details the literature survey; Section 5 details our results; Section 6 provides our discussion, summarises our findings, and identifies research priorities.

2. Related Works

Corona *et al.* [22] provide a useful overview of intrusion detection systems. They predict greater use of machine learning for intrusion detection and call for further investigation into adversarial machine learning. We now consider a number of related academic surveys that have been presented in the last 5 years with a focus on adversarial examples, security, and intrusion detection.

2.1. Secure and Trustworthy Systems

Machine learning systems are used in increasingly diverse areas including those of cyber-security. Trust in these systems is essential. Hankin and Barrèere [23] note that there are many aspects to trustworthiness: reliability, trust, dependability, privacy, resilience, and safety. Adversaries ranging from solo hackers to state-sponsored APTs have an interest in attacking these systems. Successful attacks against machine learning models mean that systems are vulnerable and therefore potentially dangerously deployed in cyber-security domains. Cho *et al.* [24] propose a framework considering the security, trust, reliability and agility metrics of computer systems; however, they do not specifically consider adversarial machine learning, or robustness to adversarial examples.

2.2. Adversarial ML in General

Papernot *et al.* [25] note that the security and privacy of ML is an active but nascent area of research. In this early work, they systematize their findings on security and privacy in machine learning. They note that a science for understanding many of the vulnerabilities of ML and countermeasures is slowly emerging. They analyse ML systems using the classical confidentiality, integrity and availability (CIA) model. They analyse: training in adversarial settings; inferring adversarial settings; robust, fair, accountable, and private ML models. Through their analysis, they identify a total of 8 key takeaways that point towards two related notions of sensitivity. The sensitivity of learning models to their training data is essential to privacy preserving ML, and similarly the sensitivity to inference data is essential to secure ML. Central to both notions of sensitivity is the generalization error (i.e. the gap between performance on training and test data). They focus on attacks and defenses for machine learning systems and hope that understanding the sensitivity of modern ML algorithms to the data they analyse will foster a science of security and privacy in machine learning. They argue that the generalization error of models is key to secure and privacy-preserving ML.

Zhang and Li [26] discuss opportunities and challenges arising from adversarial examples. They introduce adversarial examples and survey state-of-the-art adversarial example generation methods, and defenses before raising future research opportunities and challenges. They note three challenges for the construction of adversarial examples:

1. The difficulty of building a generalizable method.

 doi:10.20944/preprints202202.0099.v1

2. The difficulty in controlling the size of perturbation (too small will not result in adversarial examples, and too large can easily be perceived).
3. Difficulty in maintaining adversarial stability in real-world applications (some adversarial examples do not hold for transformations such as blurring).

They identify two challenges for defense against adversarial examples. Firstly, black-box attacks do not require knowledge of the model architecture and therefore cannot be easily resisted by modifying the model architecture or parameters. Secondly, defenses are often specific to an attack method and are less suitable as a general defense. Defenses against one attack method do not easily defend against adversarial examples based on other methods for generating adversarial examples. They subsequently identify three opportunities:

1. Construction of adversarial examples with high transferability (high confidence).
2. Construction of adversarial examples without perturbing the target image, they suggest that perturbation size will affect the success rate and transferability of adversarial examples.
3. Considering and modeling physical transformations (translation, rotation, brightness, and contrast).

Their focus is on the visual domain and they do not specifically discuss IDS or functionality-preserving adversarial attacks.

Apruzzese *et al.* [27] examine adversarial examples and consider realistic attacks, highlighting that most literature considers adversaries with complete knowledge about the classifier and are free to interact with the target systems. They Further emphasize that few works consider 'relizable' perturbations that take account of domain and/or real-world constraints. There is perhaps a perception that the threat from adversarial attacks is low based on the assumption that much prior knowledge of the system is required. This approach has some merit; however, this could be an over-confident position to take. Their idea that realistically the adversary has less knowledge of the system conflicts with Shannon's maxim [28] and Kerckhoff's second cryptographic principle [29] which states that the fewer secrets the system contains, the higher its safety. The pessimistic 'complete knowledge' position is often used in cryptographic studies, in cryptographic applications it is considered safe because it is a bleak expectation. This expectation is also realistic since we must expect well-resourced adversaries to eventually discover or acquire all details of the system. Many adversarial example papers assume complete knowledge, this is however unlikely to always be the case. Perhaps leading some to believe models are more secure against adversarial examples. However, the transferability property of adversarial examples means that complete knowledge is not required for successful attacks, and black-box attacks are possible with no prior knowledge of machine learning models. An adversary may only learn through interacting with the model. We must therefore account for the level of knowledge required by an adversary, including white-box, black-box, and gray-box knowledge paradigms.

2.3. Intrusion Detection

Wu *et al.* [30] consider several types of deep learning systems for network attack detection, including supervised and unsupervised models, and they compare the efficiency and effectiveness of different attack detection methods using two intrusion detection datasets: "KDD Cup 99" dataset and an improved version known as NSL-KDD [31] [32]. These two datasets have been used widely in the past by academic researchers; however, they do not fairly represent modern network traffic analysis problems due to concept-drift. Networks have increasing numbers of connected devices, increasing communications per second, and new applications using the network. The use of computer networks and the internet has changed substantially in twenty years. The continued introduction of IPv6, Network address Translation, Wi-Fi, mobile 5G networks, and cloud providers has changed network infrastructure [33]. Furthermore, the internet is now increasingly used for financial services. Akamai [34] report financial services now see millions or tens of millions of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

attacks each day. These attacks were less common twenty years ago. Furthermore, social media now constitutes much internet traffic and most social media platforms were founded after the KDD Cup 99 and NSL-KDD datasets were introduced. For example, Facebook, YouTube, and Twitter were founded in 2004, 2005, and 2006 respectively. This limits the validity of some research using outdated datasets. Therefore, we suggest research should use modern datasets that represent modern network traffic.

Kok *et al.* [35] analyse intrusion detection systems (IDS) that use a machine learning approach. They specifically consider the datasets used, the ML algorithms, and the evaluation metrics. They warn that some researchers are still using datasets first introduced decades ago (e.g., KDD Cup 99, NSL-KDD). They warn that this trend could result in no or insufficient progress on IDS. This would ultimately lead to the untenable position of obsolete IDS while intrusion attacks continue to evolve along with user behaviour and the introduction of new technologies. Their paper does not consider adversarial examples or robustness of ML models. Alatwi and Morisset [36] tabulate a list of Network Intrusion datasets in the literature that we extend in Table 1 .

Table 1. Datasets used in the literature.

Work	Dataset	Network	Year	Attack Categories
[31]	KDD Cup 99	Traditional	1999	DoS, Probe, User 2 Root and Remote to User
[37]	NSL-KDD	Traditional	2009	DoS, Probe, User 2 Root and Remote to User
[38]	DARPA	Traditional	2009	DDoS, Malware, Spambots, Scans, Phishing
[39]	CTU-13	Traditional	2011	Botnet
[40]	Kyoto	Traditional	2015	Botnet
[41]	UNSW-NB15	Traditional	2016	Backdoors, Fuzzers, DoS, Generic, Shell code, Reconnaissance, Worms, Exploits, Analysis
[42]	WSN-D5	Wireless	2016	Greyhole, Blackhole, Scheduling, Flooding.
[43]	SDN Traffic	SDN	2016	DDoS
[44]	CICIDS2017	Traditional	2017	DoS, DDoS, SSH-Patator, Web, PortScan, FTP-Patator, Bot.
[45]	Mirai	IoT	2017	Botnet
[44]	CICIDS2018	Traditional	2018	Bruteforce Web, DoS, DDoS, Botnet, Infiltration.
[44]	CICDDoS2019	Traditional	2019	DDoS
[46]	Bot-IOT	IoT	2018	DDoS, DoS, OS Service Scan, Keylogging, Data exfiltration
[47]	Kitsune	IoT	2018	Recon, Man in the Middle, DoS, Botnet Malware
[48]	IEEE BigData Cup	Traditional	2019	N/A
[49]	HIKARI-2021	IOT	2021	Brute force attack, Brute force attack (XML-RPC), Vulnerability, probing, Synthetic Traffic

Martins *et al.* [50] consider adversarial machine learning for intrusion detection and malware scenarios, noting that IDS are typically signature-based, and that machine learning approaches are being widely employed for intrusion detection. They describe five 'tribes' of ML algorithms before detailing some fundamentals of adversarial machine learning, including commonly used distance metrics: L_∞, L_0, and L_2. They subsequently describe common white-box methods to generate adversarial examples including: Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS), Fast Gradient Sign Method (FGSM), Jacobian-based Saliency Map Attack (JSMA), Deepfool, and Carlini & Wagner attacks (C&W). They also consider black-box methods using Generative Adversarial Networks (GANS). Traditional GANS sometimes suffer problems of mode collapse. Wasserstein Generative Adversarial Networks (WGANS) solve some of these problems. They introduce Zeroth-order optimization attack (ZOO) as a black-box method. ZOO *estimates* the gradient and optimizes an attack by iteratively adding perturbations to features. They note that most

 doi:10.20944/preprints202202.0099.v1

attacks have been initially tested in the image domain, but can be applied to other types of data which poses a security threat. Furthermore, they consider there is a trade-off when choosing an adversarial attack. For example, JSMA is more computationally intensive than FGSM but modifies fewer features. They consider JSMA to be the most realistic attack because it perturbs fewer features. When considering defenses, they tabulate advantages and disadvantages of common defenses. For example, feature squeezing is effective in image scenarios, but unsuitable for other applications because compression methods would result in data loss for tabular data. They note that GANS are a very powerful technique that can result in effective adversarial attacks where the samples follow a similar distribution to the original data but cause misclassification.

2.4. Cyber-Physical Systems

Cyber-Physical Systems (CPS) rely on computational systems to create actuation of physical devices. The range of devices is increasing from factory operations to power stations, autonomous vehicles to healthcare operations. Shafique *et al.* [51] consider such smart cyber-physical systems. They discuss reliability and security vulnerabilities of machine learning systems, including: hardware trojans, side channel attacks, and adversarial machine learning. This is important, because system aging and harsh operating environments mean CPS are vulnerable to numerous security and reliability concerns. Advanced persistent threats could compromise the training or deployment of CPSs through stealthy supply-chain attacks. A single vulnerability is sufficient for an adversary to cause a misclassification which could lead to drastic effects in a CPS (e.g. an incorrect steering decision of an autonomous vehicle could cause a collision). We consider that vulnerabilities in ML could lead to a range of unwanted effects in CPSs including those that could lead to life-threatening consequences[14]. The Stuxnet worm is an example of malware with dire consequences.

2.5. Contributions of this survey

Our main objectives are:

- Collect and collate current knowledge regarding robustness and functionality-preserving attacks in cybersecurity domains.
- Formulate key takeaways based on our presentation of the information, aiming to assist understanding of the field.

This survey aims to complement existing work while addressing clear differences, by also studying the robustness of adversarial examples, specifically functionality-preserving use cases. Most previous work aims to improve the accuracy of models or examine the effect of adversarial examples. Instead, we consider the robustness of models to adversarial examples.

Machine Learning systems are already widely adopted in cybersecurity. Indeed, with increasing network traffic, automated network monitoring using ML is becoming essential. Modern computer networks carry private personal and corporate data including financial transactions. These data are an attractive lure to cyber-criminals. Adversaries may wish to steal or disturb data. Malware, spyware, and ransomware threats are endemic on many computer networks. IDS help keep networks safe; however, an adversarial arms race exists, and it is likely that adversaries, including advanced persistent threats are developing new ways to evade network defenses. Some research has evaded intrusion detection classifiers using adversarial examples.

We identify that while adversarial examples in the visual domain are well understood, less work has focused on how adversarial examples can be applied to network traffic analysis and other non-visual domains. Similarly with machine learning models used for image and object recognition. For example, Convolutional Neural Networks (CNNs) are well researched, whereas other model types used for intrusion detection, e.g. Recurrent Neural Networks (RNNs) receive less attention. The generation of adversarial examples to fool IDS is more complicated than visual domains because the features include discrete

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

and non-continuous values [52]. Compounding the defense against adversarial examples is the overconfident assumption that successful adversarial examples require 'complete knowledge' of the model and parameters. On the contrary black-box attacks are possible with no or limited knowledge of the model. Most defenses so far proposed consider the visual domain and most are ineffective against strong and black-box attacks. This survey addresses the problem of adversarial machine learning across cyber-security domains. Further research is required to head off future mature attack methods that could facilitate more complex and destructive attacks.

3. Background

Here we provide further background on some key concepts that are related to adversarial learning, to support the reader of this survey. We cover the topics of robustness, common adversarial example algorithms, adversary capabilities, goals, and attack methods.

3.1. Robustness

Robustness can be defined as the performance of well-trained models facing adversarial examples [53]. Essentially, robustness considers how sensitive a model's output is to a change in the input. The robustness of a model is related to the generalization-error of the model. There is a recognised trade-off between accuracy and robustness in machine learning. That is, highly accurate models are less robust to adversarial examples. Machine learning models in adversarial domains must be both highly accurate and robust. Therefore, improving the robustness of machine learning models enables safer deployment of ML systems across a wider range of domains.

To critically evaluate and make fair comparisons, robustness metrics are necessary. Common metrics include: Precision, Recall, and F1 Score.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{1}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2}$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Other Possible useful metrics to evaluate robustness include: The Lipschitzian property which monitors the changes in the output with respect to small changes to inputs. CLEVER (Cross-Lipschitz Extreme Value for nEtwork Robustness) is an Extreme Value Theory (EVT) based robustness score for large-scale deep neural networks (DNNs). The proposed CLEVER score is attack-agnostic and computationally feasible for large neural networks improving on the Lipschitzian property metric [54].

Table 2. Robustness Metrics.

Work	Metric	Advantages	Disadvantages
N/A	F1-Score	Commonly used by researchers.	Biased by the majority class
[54]	CLEVER	attack-agnostic and computationally feasible	CLEVER is less suited to Black-box attacks and where gradient masking occurs [55]; however extensions to CLEVER help mitigate these scenarios[56]
[57]	Empirical Robustness	Suitable for very deep neural networks and large datasets.	N/A

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

3.2. Common Adversarial Example Algorithms

There are numerous algorithms to produce adversarial examples Szegedy *et al.* [15] used a box-constrained limited memory L-BFGS. Other methods include FGSM [58] and iterative derivatives: Basic Iterative Method (BIM), Projected Gradient Descent (PGD). JSMA optimises for the minimal number of altered features (L_0) .The Deepfool algorithm [57] optimizes for the root-mean-square (Euclidean distance , L_2). Carlini and Wagner [59] propose powerful C&W attacks optimizing for the L_0, L_2, L_∞ distance metrics. There are many algorithms to choose from. Furthermore, Papernot *et al.* [60] developed a software library for the easy generation of adversarial examples. There are now a number of similar libraries that can be used to generate adversarial examples as shown in table 3.

Table 3. Libraries for Generating Adversarial Examples

Work	Library Name	Year	Advantages	Disadvantages
[60]	CleverHans	2016	Recently updated to v4.0.0, well used by the community. MIT License	Can be complicated to configure.
[61]	Foolbox	2017	Fast Generation of Adversarial examples. MIT License	Large number of open issues
[62]	Adversarial Robustness Toolbox	2018	Well Maintained and supported. Supports most known machine learning frameworks Extensive attacks and Model robustness tools are supported	N/A
[63]	Advertorch	2019	GNU Lesser Public License	Few Active contributors

Moreover, algorithms like FGSM that modify all features are unlikely to preserve functionality. Algorithms like JSMA that modify a small subset of features are not guaranteed to preserve functionality; although, with fewer modified features, the likelihood improves. Checking for and keeping only examples that preserve functionality is possible; although it is a time-consuming and inelegant solution. A potentially better solution could ensure only functionality-preserving adversarial examples are generated.

When considering the robustness of machine learning models, we first must consider the threat model. We must consider how much the adversary knows about the classifier, ranging from *no knowledge* to *perfect knowledge*. Adversaries may have a number of different goals:

1. Accuracy degradation (where the adversary wants to sabotage the effectiveness of the overall classifier accuracy)
2. Target misclassification (where the adversary wants to misclassify a particular instance as another given class),
3. Untargeted classification (where the adversary wants to misclassify a particular instance to any random class).

We now consider the attack surface. In IDS, the attack surface can be considered as an end-to-end pipeline, with varying vulnerabilities and potential for compromise at each stage of the pipeline.

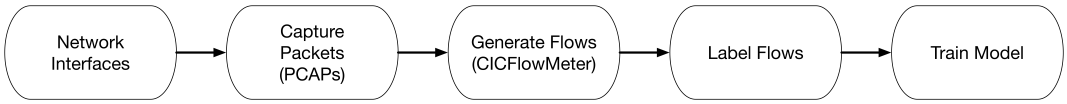

Figure 1. End to End Pipeline for Network Intrusion Detection System

In one basic pipeline as shown in Figure 1 the raw network traffic on network interfaces is collected as packet capture files (PCAPs), which are then processed into network flows. There are different applications that could be used to process PCAPs into network flows.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

CICFlowMeter [64] is a network traffic flow generator and analyser that has been used in cyber-security datasets [65] [66] and produces bidirectional flows with over 80 statistical network traffic features. The generated flows are unlabelled and so must be labelled manually with the traffic type, typically benign/malicious, although multiclasses could be labelled given sufficient information including attack type, IP source and destination dyad, duration, and start time. Finally, the labelled flows are used to train the model. Repetitive training cycles could enable detection of new attacks; however, the cyclic nature of the training means that an adversary could attack any iteration of training. Furthermore, an adversary could choose to attack any point in the pipeline. The training data used to train the model generally consists of feature-vectors and expected outputs; although some researchers are considering unsupervised learning models. The collection and validation of these data offer an attack surface. Separately, the inference phase also offers an attack surface. It is interesting to note that the size of the feature-set a machine learning model uses can be exploited as an attack surface. A fundamental issue is that each feature processed by a model may be modified by an adversary. Large feature-sets include more features and hence provide more opportunities to an adversary for manipulation. Almomani *et al.* [67] indicate accuracy can be maintained with fewer features, and McCarthy *et al.* [68] indicate that more features tend to reduce the necessary size of perturbations. Therefore, larger feature-sets are more readily perturbed than smaller feature-sets which have fewer modifiable features and hence require larger perturbations.

3.3. Threat Model - Adversary Capabilities

Adversaries are constrained by their skills, knowledge, tools, and access to the system under attack. An insider threat might have access to the classification model and other associated knowledge, whereas an external threat might only be able to examine data packets. While the attack surface may be the same for both adversaries, the insider threat is potentially a much stronger adversary because they have greater knowledge and access. Adversary capabilities mean that attacks can be split into three scenarios: White-box, Black-box, and Gray-Box.

In white-box attacks, an adversary has access to all machine learning model parameters. In black-box attacks, the adversary has no access to the machine learning model's parameters. Adversaries in black-box scenarios may therefore use a different model, or no model at all to generate adversarial examples. The strategy depends on successfully transferring adversarial examples to the target model. Gray-box attacks consider scenarios where an adversary has some, but incomplete knowledge of the system. White-box and black-box are most commonly considered.

3.4. Threat Model - Adversary Goals

Adversaries aim to subvert a model through attacking its confidentiality, integrity, or availability. Confidentiality attacks attempt to expose the model or the data encapsulated within. Integrity attacks occur when an adversary attempts to control the output of the model. For example, to misclassify some adversarial traffic and therefore allow it to pass a detection process. Availability attacks could misclassify all traffic types, or deteriorate a model's confidence, consistency, performance, and access. In this way, an integrity attack resembles a subset of availability attack, since an incorrect response is similar in nature to a correct response being unavailable; however, the complete unavailability of a response would likely be more easily noticed than decreases in confidence, consistency, or performance. The goals of an adversary may be different but are often achieved with similar methods.

3.5. Threat Model - Common Attack Methods

3.5.1. Poisoning

An adversary with access to the training data or procedure, manipulates it, implanting an attack during the training phase, when the model is trained on adversarial training

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

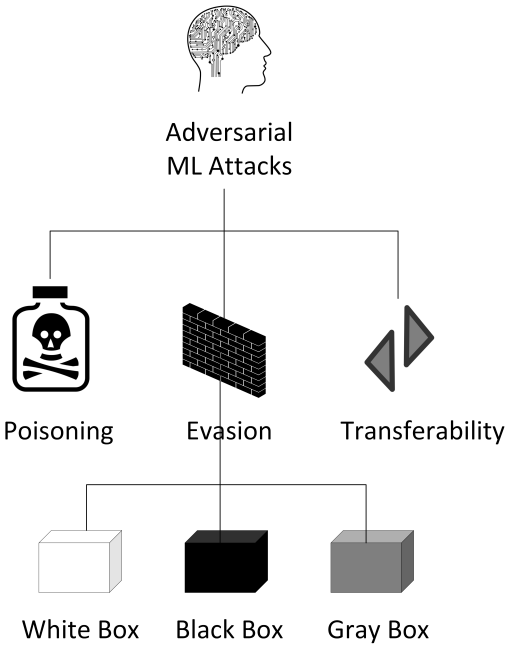

Figure 2. Common Adversarial Machine Learning Attacks

data. This is achieved with carefully crafted noise or sometimes random noise. Unused or dormant neurons in a trained Deep Neural Network (DNN) signify that a model can learn more, essentially an increased number of neurons allows for a greater set of distinct decision boundaries forming distinct classifications of data. The under-utilised degrees of freedom in the learned model could potentially be used for unexpected classification of inputs. That is, the model could learn to provide selected outputs based on adversarial inputs. These neurons have very small weights and biases. However, the existence of such neurons allows successful poisoning attacks through training the model to behave differently for poisoned data. This suggests that distillation [69] could be effective at preventing poisoning attacks, because smaller models have lower knowledge capacity and likely fewer unused neurons. Distillation reduces the number of neurons that contribute to a model by transferring knowledge from a large model to a smaller model. Despite initial analysis indicating reduction in the success of adversarial attacks, Carlini [59] experimented with three powerful adversarial attacks and a high confidence adversarial example in a transferability attack, and found that distillation does not eliminate adversarial examples and provides little security benefit over undistilled networks in relation to powerful attacks. Unfortunately, they did not specifically consider poisoning attacks. Additional experiments could determine whether distillation is an effective defense against poisoning attacks.

3.5.2. Evasion

In evasion attacks, the adversary is often assumed to have no access to the training data. Instead, adversaries exploit their knowledge of the model and its parameters, aiming to minimise the cost function of adversarial noise, which when combined with the input causes changes in the model output. Untargeted attacks lead to a random incorrect output, targeted attacks lead to a specific incorrect output, and an attack may disrupt the model by changing the confidence of the output class. In the visual domain, the added noise is often imperceptible to humans. In non-visual domains such as intrusion detection, this problem may be much more challenging since even small modifications may corrupt network packets and may cause firewalls to drop these malformed packets. This highlights the need for functionality preservation in adversarial learning, as a clear distinction from vision-based

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

attacks that exploit the human visual system.

4. Methodology

In this section, we describe our approach for surveying the literature to conduct an effective and meaningful survey of the literature.

Eligibility Criteria We determined our search terms leading to the most relevant articles. We chose the following search terms in table 4.

Table 4. Topics and associated search terms used in this survey.

Topic	Search Query
Cyber Security / Intrusion Detection	("Cyber Security" OR "Intrusion detection" OR IDS)
Adversarial Machine Learning Attacks and Defences	("adversarial machine learning" or "machine learning" or "adversarial example") and (attack or defence)
Robustness / Functionality Preservation	((robustness or generalization error or accuracy or f1score or f-score or TPR or FPR) or ((functionality or payload) and preservation)))

We expect these to result in good coverage of the relevant literature. We searched each database using the identified search terms. The literature search was conducted up to September 2021. Generally, we have chosen to exclude works that have not yet been peer-reviewed, such as those appearing on arXiv, unless deemed by the authors as a significant paper that makes a clear contribution to the subject domain. We collated the searches and any subsequent duplicates were removed. Each paper was screened by reading the title and abstract to determine the relevance. Inclusion criteria were: the article is related to functionality preservation in adversarial machine learning for cybersecurity or intrusion detection with insight into robust classification.

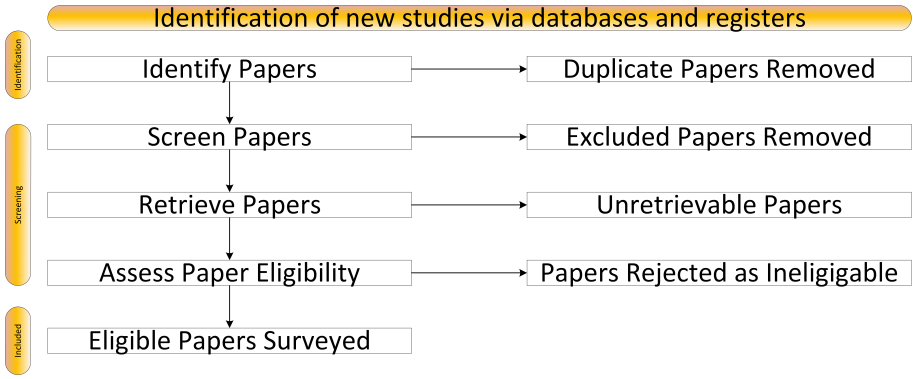

Figure 3. Preferred Reporting Items for Systematic Meta-Analysis

From this large list, we specifically focused on adversarial machine learning attacks and defenses, narrowing the literature down to relevant papers. Our selection process is roughly based on the Preferred Reporting Items for Systematic Meta-Analysis (PRISMA) framework [70]. Figure 3 details our selection process.

Information Sources We selected the following databases with a sizeable computer science element to search and retrieve literature: **IEEE Xplore**, **ACM Digital Library**, **ScienceDirect**, **Scopus**, **SpringerLink**, **Google Scholar**.

5. Results

In this section, we describe the results of our search and selection process. We further describe our classification scheme, tabulate and discuss our findings including: Adversarial Attacks in traditional and cybersecurity domains of malware, IDS, and CPS. We include 146 relevant papers in this survey.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

Table 5. Chronologically ordered summary of adversarial example attacks.

Work	Year	Attack	Type			Obj		Domain			Model			Knowledge			Constraint		
			AE	Sequence of AEs	Transferability	Targeted	Untargeted	Visual	Cybersecurity	Text	MLP	CNN	RNN	White-Box	Black-box	Gray-Box	Box	Sparse	Func-Preserving
[15]	2014	L-BFGS	✓			✓		✓				✓		✓			✓		
[71]	2013	GradientDescent	✓			✓		✓				✓		✓		✓			
[72]	2016	Adversarial Sequences		✓			✓			✓			✓	✓					✓
[73]	2016	JSMA	✓			✓	✓	✓				✓		✓				✓	
[57]	2016	Deepfool	✓				✓	✓				✓		✓					
[74]	2017	AddSent,AddOneSent	✓	✓			✓			✓			✓		✓				
[75]	2018	GAN	✓				✓	✓				✓			✓				
[76]	2017	EnchantingAttack		✓		✓		✓			✓			✓					
[76]	2017	StrategicAttack	✓			✓		✓			✓			✓					
[59]	2017	C&W, L_0, L_2,L_∞	✓				✓	✓				✓		✓					
[77]	2017	FGSM,JSMA	✓				✓		✓		✓			✓					
[78]	2018	Generative RNN	✓				✓		✓				✓		✓				✓
[79]	2018	NPBO	✓				✓		✓		✓			✓					✓
[80]	2018	GADGET	✓				✓		✓				✓		✓				✓
[81]	2018	JSMA,FGSM,DeepFool,CW	✓			✓			✓		✓			✓					
[82]	2018	FGSM	✓				✓		✓		✓			✓					
[83]	2018	IDS-GAN	✓				✓		✓		✓				✓				
[84]	2018	ZOO, GAN	✓				✓		✓		✓				✓				
[85]	2019	One Pixel Attack	✓			✓		✓				✓		✓				✓	
[86]	2019	ManifoldApproximation	✓				✓		✓		✓				✓				✓
[87]	2019	FGSM,BIM,PGD	✓				✓		✓		✓			✓		✓			
[88]	2019	GAN Attack	✓				✓		✓		✓				✓				✓
[89]	2020	PWPSA	✓	✓	✓		✓		✓				✓	✓					✓
[89]	2020	GA	✓	✓	✓		✓		✓				✓	✓					✓
[90]	2020	One Pixel Attack	✓				✓		✓			✓		✓				✓	
[91]	2020	Opt Attack,GAN Attack	✓				✓		✓		✓				✓				✓
[92]	2021	GAMMA	✓				✓		✓						✓				✓
[93]	2021	UAP	✓				✓		✓		✓			✓		✓			✓
[94]	2020	Variational Auto Encoder	✓				✓		✓		✓			✓					✓
[95]	2021	Best-Effort Search	✓				✓		✓		✓	✓		✓	✓	✓	✓		✓

5.1. Classification Scheme

We classify attacks by attack type, attack objective (targeted/untargeted), domain, model, knowledge required, and whether any constraints are placed on the adversarial examples. Defenses are classified by type, domain, and model. To summarise the attacks and defenses, we produced three tables. Attacks are detailed in Table 5. Defenses are detailed in Table 7. Functionality-preserving attacks are detailed in 6.

5.2. Adversarial Example Attacks

The attacks we focus on exploit adversarial examples that cause differences in the output of neural networks. Adversarial examples were discovered by Szegedy *et al.* [15]. Adversarial examples are possible in ANN as a consequence of the properties of neural networks; however, they are possible for other ML models. This complicates mitigation efforts, and adversarial examples can be found for networks explicitly trained on adversarial examples [96]. Furthermore, adversarial examples can be algorithmically generated, e.g. using gradient descent. Moreover, adversarial examples are often transferable, that is, an adversarial example presented to a second machine learning model trained on a subset of the original dataset may also cause the second network to misclassify the adversarial example.

 doi:10.20944/preprints202202.0099.v1

5.2.1. Adversarial Examples - Similarity Metrics

In the visual domain, distance metrics are well used to judge how similar two inputs are, and therefore how easy the differences might be perceived. The following metrics are commonly used to describe the difference between normal and adversarial inputs:

- Number of altered pixels (L_0)
- Euclidean distance (L_2, root-mean-square)
- Maximum change to any of the co-ordinates. (L_∞)

Human perception may not be the best criterion to judge a successful adversarial input. A successful attack in a vision ML task may be to fool a human. Success in an ML-based system is to fool some other detection routine, while conforming to the expected inputs of the system. For example, a malicious packet must remain malicious after any perturbation has been applied. If a perturbed packet is very close to the original packet, this would only be considered successful if it also retained its malicious properties, and hence its intended function.

5.2.2. Adversarial Examples - Types of Attack

White-box Attacks: Most white-box attacks are commonly achieved through gradient descent to increase the loss function of the target model. The algorithmic generation of adversarial examples is possible. Moreover, Papernot *et al.* [60] developed a software library for the easy generation of adversarial examples and other libraries are now available. An early gradient descent approach was proposed by Szegedy *et al.* [15] using a box-constrained limited memory L-BFGS. Given an original image, this method finds a different image that is classified differently, whilst remaining similar to the original image. Gradient descent is used by many different algorithms; however, algorithms have been designed to be optimized for different distance metrics. There are numerous gradient descent algorithms that produce adversarial examples; they can differ in their optimization. FGSM [58] was improved by Kurakin *et al.* [97] who refined the fast gradient sign by taking multiple smaller steps. This iterative granular approach improves on FGSM by limiting the difference between the original and adversarial inputs. This often results in adversarial inputs with a predictably smaller L_∞ metric. However, FGSM modifies all parameters. This is problematic for features that must remain unchanged or for discrete features such as Application Programming Interface (API) calls. JSMA differs from FGSM in that it optimizes to minimize the total number of modified features (L_0 metric). In this greedy algorithm, individual features are chosen with the aim of step-wise increasing the target classification in each iteration. The gradient is used to generate a saliency map, modelling each feature's impact towards the resulting classification. Large values significantly increase the likelihood of classification as the target class. Thus, the most important feature is modified at each stage. This process continues until the input is successfully classified as the target class, or a threshold number of pixels is reached. This algorithm results in adversarial inputs with fewer modified features. The Deepfool algorithm [57] similarly uses gradient descent but optimizes for the root-mean-square also known as Euclidean distance (L_2). This technique simplifies the task of shifting an input over a decision boundary by assuming a linear hyper-plane separates each class. The optimal solution is derived through analysis and subsequently an adversarial example is constructed; however, neural network decision boundaries are not truly linear. Therefore, subsequent repetitions may be required until a true adversarial image is found.

The optimizations for different distance metrics are types of constraint: Maximum change to any feature (L_∞); minimal root-mean-square (L_2); minimal number of altered features (L_0). Constrained adversarial examples are important for functionality-preserving attacks. Additional constraints for specific domains are likely required, and this remains an open avenue for further research.

Most gradient descent algorithms were originally presented in the visual domain and used on images and pixel values. The pixel values of images are often presented as

 doi:10.20944/preprints202202.0099.v1

continuous values ($0 - 255$). The use of adversarial examples with discrete data values is less well explored and remains an interesting avenue for further research.

Black-box Attacks: Researchers have also considered black-box attacks that do rely on gradient descent. Some black-box techniques commonly rely on the transferability of adversarial examples. Table 5 shows that few researchers employ the transferability of adversarial examples. Other common black-box techniques include GANS and Genetic Algorithms (GAs). Sharif *et al.* [98] propose a way of attacking DNN with a general framework to train an attack generator or generative adversarial network (GAN). GANs can be trained to produce new, robust, and inconspicuous adversarial examples. Attacks like Biggio *et al.* [71] are more suitable for the security domain where assessing the security of algorithms and systems under worst-case attacks [99] [100].

An important consideration in attacks against intrusion detection systems is that attackers cannot perform simple oracle queries against an intrusion detection system and must minimize the number of queries to decrease the likelihood of detection. Apruzzese *et al.* [101] further note that the output of the target model is not directly observable by the attacker; however, exceptions occur where detected malicious traffic is automatically stopped or dropped, or where the attacker gains access to/or knowledge of the system.

Gray-box attacks consider scenarios where an adversary has only partial knowledge of the system.

Building on Simple Adversarial Examples: Table 5 shows much research considers simple adversarial examples; although less research considers sequences of adversarial examples or transferability. We chose to classify attacks as either a simple adversarial example, a sequence of adversarial examples, or a transferable adversarial example. A simple adversarial example is sufficient to alter the output of a simple classifier. Lin *et al.* [76] suggest using adversarial examples strategically could affect the specific critical outputs of a machine learning system. Sequences of adversarial examples consist of two or more adversarial examples. Sequences of adversarial examples are more challenging than simple adversarial examples. Lin *et al.* [76] further suggest an *enchanting* attack to lure a machine learning system to a target state through crafting a series of adversarial examples. Table 5 shows that most research considers simple adversarial examples. Researchers are starting to consider sequences of adversarial examples and consider the transferability of adversarial examples. We chose to classify attacks in this way to clarify the complexity level of attack types. Furthermore, the table shows that sequences of adversarial examples and the transferability of adversarial examples is under-represented, providing opportunities for further research.

5.2.3. Adversarial Examples - Attack Objectives

There is a distinction between the objectives of attacks: targeted or untargeted. An attack objective might be to cause a classifier to misclassify an input as *any* other class (untargeted) or to misclassify an input as a *specific* class (targeted). In the cyber-security domain, IDS often focus on binary classification: malicious or benign. For binary classification the effect of targeted and untargeted attacks is the same. More complex multi-class IDS can help network analysts triage or prioritise different types of intrusions. Network analysts would certainly treat a Distributed Denial of Service (DDoS) attack differently than a BotNet or infiltration attempt. Adversaries could gain significant advantage through *targeted* attacks. For example, by camouflaging an infiltration attack as a comparatively less serious network intrusion.

Recent research goes beyond adversarial examples causing misclassification of a single input. Moosavi-Dezfooli *et al.* [102] further show the existence of untargeted *universal* adversarial perturbation (UAP) vectors for images, and venture this is problematic for

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

classifiers deployed in real-world and hostile environments. In the cyber-security domain Labaca *et al.* [93] demonstrate UAPs in the feature space of malware detection. They show that UAPs have similar effectiveness as adversarial examples generated for specific inputs. Sheatsley *et al*. [103] look at UAP in the constrained domain of intrusion detection. Adversaries need only calculate one UAP that could be applied to multiple inputs. Pre-calculation of a UAP could enable faster network attacks (DDoS) that would otherwise require too much calculation time. Table 5 shows most research considers untargeted attacks. Targeted attacks are less represented in the literature. Furthermore, UAPs are a more recent avenue for research.

5.2.4. Adversarial Examples in Traditional Domains

Table 5 shows attacks in the visual domain were the subject of much early research, and the visual domain continues to attract researchers; although, researchers are beginning to consider attacks against other DNN systems such as machine learning models for natural language processing, with some considering semantic preserving attacks.

In visual domains, features are generally continuous. For example, pixel values range from $0 - 255$. A consensus exists in the visual domain that adversarial examples are undetectable to humans. Moreover, the application domain is clearly interrelated with the choice of machine learning model. Models such as CNNs are appropriate for visual-based tasks, whereas RNNs are appropriate for sequence-based tasks. We discuss model types in section 5.2.6.

Some models such as recurrent neural networks cannot be attacked using traditional attack algorithms; however, some research aims to discover new methods to attack these systems. Papernot *et al.* [72] note that because RNNs handle time sequences by introducing cycles to their computational graphs. The presence of these computation cycles means that applying traditional adversarial example algorithms is challenging because cycles prevent direct computation of the gradients. They adapt adversarial example algorithms for RNNs and evaluate the performance of their adversarial samples. If the model is differential, FGSM can be applied even to RNN models. They use a case study of a binary classifier (positive or negative) for movie reviews. They define an algorithm that iteratively modifies words in the input sentence to produce an adversarial sequence that is misclassified by a well-trained model. They note that their attacks are white-box attacks, requiring access to, or knowledge of, the model parameters. Szegedy [15] discovered the transferability of adversarial examples, noting the same perturbation can cause a different network that was trained on a different subset of the dataset, to misclassify the same input. This property of adversarial examples has serious implications because it means gaining access to a model is unnecessary to attack it. An adversary can employ the transferability of adversarial examples, where adversarial examples generated against a model under the adversary's control can be successfully used to attack the target model. The transferability of adversarial examples implies that an adversary does not need full access to a model to attack it (Black-box).

5.2.5. Adversarial Examples in Cyber-Security Domains

Adversarial examples (AE) have been shown to exist in many domains. Indeed, no domain identified (so far) is immune to adversarial examples [103]. Researchers are beginning to consider cyber-security domains where features are often a mixture of categorical, continuous and discrete. Some research focuses on adversarial example attacks against IDS; although few specifically consider functionality-preserving attacks.

In the visual domain, we briefly discussed the consensus that adversarial examples are undetectable to humans. However, it is unclear how this idea should be translated to other domains. Carlini [59] holds that, strictly speaking, adversarial examples must be similar to the original input. However, Sheatsley *et al.* [103] note that research in non-visual domains provide domain specific definitions: perturbed malware must preserve its malware functionality [103], perturbations in audio must be *nearly* inaudible [103],

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

Table 6. Functionality-Preservation in Cybersecurity and Intrusion Detection.

Work	Year	Domain	Generation Method	Realistic Constraints	Findings
[52]	2019	Malware	Gradient-based	Minimal Content additions/modification	Experiments showed that we are able to use that information to find optimal sequences of transformations without rendering the malware sample corrupt.
[88]	2019	IDS	GAN	Preserve functionality	The proposed adversarial attack successfully evades the IDS while ensuring preservation of functional behavior and network traffic features.
[104]	2019	IDS	Gradient-based	Respect mathematical dependencies and domain constraints.	Evasion attacks achieved by inserting a dozen network connections.
[105]	2019	IDS	Random Modification $\leq$ 4 features: flow duration, sent bytes, received bytes, exchanged packets.	Retain internal logic	Feature removal is insufficient defense against functionality-preserving attacks which may are possible by modifying very few features.
[106]	2019	IDS	Legitimate transformations: Split, Delay, Inject	Packets must maintain malicious intent, transformations hold to underlying protocols.	Detection rate of packet-level features dropped by up to 70% and flow-level features dropped by up to 68%.
[103]	2020	IDS - Flows	Jacobian Method (JSMA)	Obey TCP/IP constraints	Biased distributions with low dimensionality enable constrained adversarial examples. Constrained to five random features, -50% adversarial examples succeed.
[89]	2020	IDS - packet	Valid packet	Minimal modification/insertion of packets	Experimental results show powerful and effective functionality preserving attacks. More accurate models are more susceptible to adversarial examples.
[92]	2021	Malware	Injected unexecuted benign content	Minimal Injected Content	Section-injection attack can decrease the detection rate. Their analysis highlights that commercial products can be evaded via transfer attacks.
[95]	2021	CPS	Best-effort search	Real-world linear inequality	Best-effort search algorithms effectively generate adversarial examples meeting linear constraints. Their evaluation shows constrained adversarial examples significantly decrease detection accuracy.
[107]	2021	IDS	Minimal perturbation of each feature	FGSM	Functionality is not reported, but is less likely to preserve functionality because all features are perturbed.
[108]	2021	CPS/ICS	JSMA	Minimal number of perturbed features	Functionality is not reported, but is more likely to preserve functionality because relatively few features are perturbed.
[109]	2021	IDS	PSO-based mutation	original traffic retained and packet order is unchanged	Measured attack effect, malicious behavior and attack efficiency
[110]	2021	IDS	GAN	preserving functional features of attack traffic	F1 score drop to zero from around 99% DIGFuPAS adversarial examples.
[111]	2021	IDS	PSO/GA/GAN	Only modify features where network functionality is retained	in the network traffic data, it is unrealistic to assume an adversary can alter all traffic features - ConstraintS on features that do not break functionality
[112]	2020	IDS	GAN/PSO	Original traffic and packet order is retained.	Detection performance and robustness should both be considered in feature extraction systems.

 doi:10.20944/preprints202202.0099.v1

Figure 4. Common Machine Learning Tasks in Cyber Security

perturbed text must preserve its meaning. Sheatsley *et al.* further offer a definition for adversarial examples in intrusion detection: perturbed network flows must maintain their attack behaviour. We consider human perception may not be the best criterion for defining adversarial examples in cyber-security domains. Indeed, human perception in some domains might be immaterial. For example, only very skilled engineers could *perceive* network packets in any meaningful way even with the use of network analysis tools. Furthermore, users likely cannot perceive a difference between the execution of benign or malicious software. After malware is executed, the effects are clear; however, during malware execution users often suspect nothing wrong. We therefore consider that while fooling human perception remains a valid ambition. It is critical that adversarial perturbations in cyber-security domains preserve functionality and behaviour.

In the cyber-security domain, traditional gradient descent algorithms may be insufficient. Algorithms that preserve functionality are required. Moreover, some models used in the cyber-security domain are distinct from those used for purely visual problems. For example, RNNs are useful for time sequences of network traffic analysis. We now consider recent functionality-preserving attacks in the cybersecurity domains of Malware, Intrusion Detection, and CPS.

Malware: Hu and Tan [78] propose a novel algorithm to generate adversarial sequences to attack a RNN based malware detection system. They claim that algorithms adapted for RNNs are limited because they are not truly sequential. They consider a system to detect malicious API sequences. Generating adversarial examples effective against such systems is non-trivial because API sequences are discrete values. There is a discrete set of API calls; changing any single letter in an API call will create an invalid API call and cause that API call to fail. This will result in a program crash. Therefore, any perturbation of an API call must result in a set of valid API calls. They propose an algorithm based around a generative RNN and a substitute RNN. The generative RNN takes an API sequence as input and generates an adversarial API sequence. The substitute RNN is trained on benign sequences and the outputs of the generative RNN. The generative model aims to minimize the predicted malicious probability. Subsequently, adversarial sequences are presented to six different models. Following adversarial perturbation, the majority of the malware was not detected by any victim RNNs. The authors note that even when the adversarial generation algorithm and the victim RNN are implemented with different models and trained on different training sets, the majority of the adversarial examples successfully attack the victim RNN through the *transferability* property of adversarial examples. In MLP, they report a TPR of 94.89% which falls to 0.00% under adversarial perturbations.

Demetrio *et al.* [92] preserve the functionality of malware while evading static windows malware detectors. Their attacks exploit the structure of portable executable (PE) file format. Their framework has three categories of functionality-preserving manipulations: Structural, Behavioural, and Padding. Some of their attacks work by injecting unexecuted (benign) content in new sections in the PE file, or at the end of the malware file. The attacks are a constrained minimization problem optimizing the trade-off between the probability of evading detection and the size of injected content. Their experiments successfully evade two Windows malware detectors with few queries and small payload size. Furthermore, they discover their attacks transfer to other windows malware products. We note that the creation of new sections provides a larger attack surface that may be populated with

 doi:10.20944/preprints202202.0099.v1

adversarial content. They report that their section-injection attack is able to drastically decrease the detection rate (e.g., from an original detection rate of 93.5% to 30.5% also significantly outperforming their random attack at 85.5%).

Labaca-Castro *et al.* [52] present a gradient-based method to generate valid executable files that preserve their intended malicious functionality. They note that malware evasion is a current area of adversarial learning research. Evading the classifier is often the foremost objective; however, the perturbations must also be carefully crafted to preserve the functionality of malware. They note that removing objects from a PE file often leads to corrupt files. Therefore, they only implement additive or modifying perturbations. Their gradient-based attack relies on *complete-knowledge* of the system with the advantage that the likelihood of evasion can be calculated and maximised. Furthermore, they state that their system only generates valid executable malware files.

Intrusion Detection: Usama *et al.* [88] use a Generative Adversarial Network (GAN) to generate functionality-preserving adversarial examples. They note that adversarial examples aiming to evade IDS should not invalidate network traffic features. A typical GAN composed of two neural networks: a generator G and discriminator D is used to construct adversarial examples that masquerade as benign but functionally probe the network. Their attack is able to evade an IDS while preserving the intended behaviour. They suggest that adversarial training using GAN generated adversarial examples improves the robustness of their model. They report F1-Scores of 89.03 (original), 40.86 (After attack), 78.49 (after adversarial training), and an improved 83.56 after GAN-based adversarial training.

Wang *et al.* [113] note that relatively few researchers are addressing adversarial examples against IDS. They propose an ensemble defense for network intrusion detection that integrates GANS and adversarial retraining. Their training framework improves robustness while maintaining accuracy of unperturbed samples. Unfortunately, they evaluate their defences against traditional attack algorithms: FGSM, Basic Iterative Method (BIM), Deepfool, JSMA. However, they do not specifically consider functionality-preserving adversarial examples. They further recognise the importance of using recent datasets for intrusion detection. They report F1-Scores for three classifiers and a range of adversarial example algorithms. For example, F1-Score for an ensemble classifier tested on clean data is 0.998 compared to 0.746 for JSMA. Among all classifiers, the ensemble classifier achieved superior F1-Scores under all conditions.

Huang *et al.* [89] note that it is more challenging to generate DDoS adversarial examples because of their discrete properties. They note that work in the visual domain cannot be directly applied to adversarial examples for intrusion detection of DDoS. The input to their algorithm is a series of packets. This makes it difficult to optimize the distance between the original and adversarial sample while guaranteeing the validity of each packet. They propose two black-box methods to generate DDoS adversarial examples against LSTM-based intrusion detection system: Genetic Algorithm (GA) and Probability Weighted Packet Saliency Attack (PWPSA). Each method modifies the original input, either inserting or modifying packets. The GA method evolves a population of DDoS samples and selects adversarial examples from the population. In PWPSA find the most important packet in the sequence and replaces it with a different 'best packet' for this position. Both methods produce adversarial examples that can successfully evade their DDoS intrusion detection model. They report success rates for their different attacks against different detectors. For example, success results for detector D: GA-Replace 91.37 % GA-Insert 74.5%, PWPSA-Replace 88.9%, PWPSA-Insert 67.17 %.

Cyber-Physical Systems: Cai *et al.* [94] warn that adversarial examples have consequences for system safety because they can cause systems to provide incorrect outputs. They present a detection method for adversarial examples in CPS. They use a case study of an Advanced Emergency Braking System where a DNN estimates the distance to an obstacle. Their adversarial example detection method uses a variational auto-encoder to predict a target variable (distance) and compare it with a new input. Any anomalies are

 doi:10.20944/preprints202202.0099.v1

considered adversarial. Furthermore, adversarial example detectors for CPS must function efficiently in a real-time monitoring environment and maintain low false alarm rates. They report since the p-values for the adversarial examples are almost 0, the number of false alarms is very small and the detection delay is smaller than 10 frames or 0.5 s.

CPS include critical national infrastructure such as power grids, water treatment plants, and transportation. Li *et al.* [95] assert that adversarial examples could exploit vulnerabilities in CPS with terrible consequences; however, such adversarial examples must satisfy real-world constraints (commonly linear inequality constraints). For example, meter readings downstream may never be larger than meter readings upstream. Adversarial examples breaking constraints are noticeably anomalous. Risks to CPS arising from adversarial examples are not yet fully understood. Furthermore, algorithms and models from other domains may not readily apply because of distributed sensors and inherent real-world constraints. However, generating adversarial examples that meet such linear constraints were successfully applied to power grids and water treatment system case studies. The evaluation results show that even with constraints imposed by the physical systems, their approach still effectively generates adversarial examples, significantly decreasing the detection accuracy. For example, they report the accuracy under adversarial conditions to be as low as 0%.

5.2.6. Adversarial Examples and Model Type

We classify models based on their architecture in four broad types: Multi-Layer Perceptron (MLP), CNN, RNN, and RF. Ali *et al.* [114] observed that different deep learning architectures are more robust than others. They note that CNN and RNN detectors are more robust than MLP and hybrid detectors, based on low attack success rate and high query counts. Architecture plays a role in the accuracy of these models because CNNs can learn contextual features due to their structure, and RNNs are temporally deeper, and thus demonstrate greater robustness.

Unsurprisingly, research on CNNs coincides with research in the visual domain as shown in Table 5. The majority of adversarial example research on RNNs has until recently focused on the text or natural language domain; however, RNNs are also useful in the cybersecurity domain and researchers have recently considered adversarial example attacks against RNN-based IDS.

Other promising research shows that radial basis function neural networks (RBFNN) are more robust to adversarial examples [115]. RBFNNs fit a non-linear curve during training, as opposed to fitting linear decision boundaries. Commonly RBFNNs transform the input such that when it is fed into the network it gives a linear separation. The non-linear nature of RBFNNs could be one potential direction for adversarial example research. Powerful attacks able to subvert RBFNNs would improve our understanding of decision boundaries. Goodfellow *et al.* [58] argue the primary cause of neural networks' vulnerability to adversarial perturbation is their linear nature. However, RBFNNs are less commonly deployed and are therefore not further discussed.

5.2.7. Adversarial Examples and Knowledge Requirement

The majority of the research focus is on white-box attacks as shown in Table 5, perhaps because such attacks are known to be efficient and effective. Less research focuses on black-box attacks and few recognise gray-box attacks that need only partial model knowledge. Gray-box attacks will likely have advantages over black-box attacks. Adversaries will undoubtedly use any and all information available to them.

We classify the attacks on the knowledge required by the adversary. White-box attacks are likely the most effective and efficient method of attack, because the adversary has *complete-knowledge* of the model architecture, and information on how the model was trained. However, access to this knowledge is harder to attain; although it might also be gained through insider threats [116] or model extraction attacks [117]. Extracted models might be a feasible proxy on which to develop and test adversarial examples.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

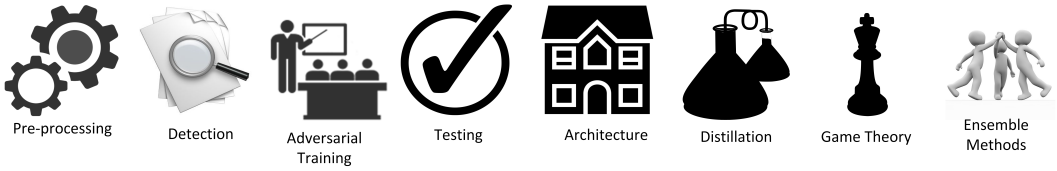

Figure 5. Common Defence Types against Adversarial Machine Learning

Notwithstanding the efficiency of white-box attacks, effective black-box attacks are possible. Black-box (or oracle) attacks require no knowledge of the model. Adversaries only need the ability to query the model and receive its output. Adversaries generate inputs and receive the output of the model. Typical black-box attacks include GA [89] , and GANs [83]. [91].

Gray-box attacks require only limited model knowledge, perhaps including knowledge of the features used by the model. This is a realistic prospect as adversaries will likely have or gain at least partial knowledge of the model.

5.2.8. Adversarial Example Constraints

Table 5 shows little research considering constraints of any sort. Much research on IDS ignores constraints; however, network traffic is highly constrained by protocols, and some network firewalls may drop malformed packets. Furthermore, it is insufficient that well-formed adversarial examples progress past firewalls. They must also retain their intended functionality.

Stringent constraints exist in the cyber-security domain. Extreme care must be taken to create valid adversarial examples. For example, in IDS, adversaries must conform the protocol specification of the TCP/IP stack.

We classify adversarial example constraints into three groups: 1) box constraints, simple constraints where values must remain within certain values. 2) Sparse constraints, where a maximum number of features can be modified, the most extreme version being where only one feature can be modified. . 3) functionality-preserving constraints, where adversarial examples must retain their original functionality. For example, malware must function as malware when perturbed to evade a malware detector, and DDoS attacks must function as DDoS attacks when perturbed to evade detection. Functionality-preserving adversarial examples are an interesting avenue for further research.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

5.3. Defenses Against Adversarial Examples

Table 7. Chronologically ordered summary of defenses against adversarial examples.

Work	Year	Defense	Type							Domain				Model		
			Pre-Process	Detection	Adv-Training	Testing	Architectural	Distillation	Ensemble	Game Theory	Visual	Cybersecurity	Text	MLP	CNN	RNN	RF
[58]	2014	Adversarial Training			✓						✓				✓		
[69]	2016	Distillation as defense						✓			✓				✓		
[118]	2016	Feedback Alignment					✓				✓				✓		
[119]	2016	Assessing Threat	✓								✓				✓		
[120]	2017	Statistical Test		✓							✓	✓		✓	✓		
[121]	2017	Detector SubNetwork		✓							✓				✓		
[122]	2017	Artifacts		✓							✓				✓		
[123]	2017	MagNet		✓							✓				✓		
[124]	2017	Feature Squeezing		✓							✓				✓		
[125]	2017	GAT			✓						✓				✓		
[96]	2018	EAT			✓						✓				✓		
[126]	2018	Defense-GAN			✓						✓				✓		
[97]	2018	Assessing Threat	✓								✓				✓		
[127]	2018	Stochastic Activation Pruning					✓				✓			✓	✓		
[128]	2018	DeepTest				✓					✓				✓		
[129]	2018	DeepRoad				✓					✓				✓		
[130]	2018	Defensive Dropout					✓				✓				✓		
[130]	2018	Def-IDS			✓							✓		✓			
[99]	2018	Multi-Classifier System							✓		✓				✓		
[131]	2019	Weight Map Layers					✓				✓				✓		
[132]	2019	Sequence Squeezing		✓								✓				✓	
[105]	2019	Feature Removal	✓									✓		✓			
[133]	2020	Adversarial Training			✓							✓					✓
[134]	2020	Adversarial Training			✓							✓					✓
[135]	2019	Game Theory								✓		✓		✓			
[136]	2020	Hardening			✓					✓		✓					✓
[137]	2021	Variational Auto-encoder		✓							✓				✓		
[138]	2021	MANDA		✓								✓		✓			

It is hard to defend against adversarial examples. People expect ML models to give good outputs for all possible inputs. Because the range of possible inputs is so large, it is difficult to guarantee correct model behaviour for every input. Some researchers explore the possibility of exercising all neurons during training [128]. Furthermore, consideration must be given to how adversaries might react when faced with a defense. Researchers in secure machine learning must evaluate whether defenses remain secure against adversaries with knowledge of model defenses.

We classify the suggested defenses against adversarial examples into the following groups: Pre-processing, Adversarial Training, Architectural, Detection, Distillation, Testing, Ensembles, and Game Theory.

5.3.1. Pre-Processing as a Defense against Adversarial Examples

Some promising research considers transformations such as: translation, additive noise, blurring, cropping, resizing. These often occur with cameras and scanners in the visual domain. Translations have shown initial success in the visual domain. Initial successes have prompted some researchers to discount security concerns. For example, Graese [119] overreaches by declaring adversarial examples an 'academic curiosity', not a

 doi:10.20944/preprints202202.0099.v1

security threat. This position misunderstands the threat from adversarial examples which remain a concern for cyber-security researchers.

Eykolt *et al.* [139] note the creation of perturbations in physical space that survive more challenging physical conditions (distance, pose, and lighting). Transformations are appropriate for images; however, such translations may make little sense in cyber-security domains. For example, what would it mean to rotate or blur a network packet? Nevertheless, inspiration could be taken from pre-processing methods in the visual domain. Adapting pre-processing methods to cyber-security and other nonvisual domains is an interesting avenue for research.

5.3.2. Adversarial Training as a Defense against Adversarial Examples

Szegedy *et al.* [15] found robustness to adversarial examples can be improved by training a model on a mixture of adversarial examples and unperturbed samples. Specific vulnerabilities in the training data can be identified through exploring UAPs. Identified vulnerabilities could potentially be addressed with adversarial training. We recognise adversarial training is a simple method aiming to improve robustness; however, it is potentially a cosmetic solution: the problem of adversarial examples cannot be solved only through ever greater amounts of adversarial examples in the training data. Tramér *et al.* [96] found adversarial training is imperfect and can be bypassed. Moreover, black-box attacks have been shown to evade models subject to adversarial training. Adversarial training has some merit because it is a simple method to improve robustness. It is unfortunately not a panacea and should be bolstered by other defenses. Research avenues could combine adversarial training with other techniques. We warn that models used in cyber-security or other critical domains should not rely solely on adversarial training.

5.3.3. Architectural Defenses against Adversarial Examples

Some research, rather than modifying a model's training data investigate defenses through hardening the architecture of the model. This could involve changing model parameters or adding new layers. In table 7 we classify such defenses as architectural.

Many white-box attacks rely on the quality of the gradient. Some research considers how the model's weights can be used to disrupt adversarial examples. Amer and Maul [131] modify Convolutional Neural Networks (CNN) adding a weight map layer. Their proposed layer easily integrates into existing CNNs. A Weight Mapping layer may be inserted between other CNN layers; thus increasing the network's robustness to both noise and gradient-based adversarial attacks.

Other research aims to block algorithms from using weight transport and back-propagation to generate adversarial examples. Lillicrap *et al.* [118] propose a mechanism called 'feedback alignment' which introduces a separate feedback path via random fixed synaptic weights. Feedback alignment blocks the generation of adversarial examples that rely on the gradient because it uses the separate feedback path rather than weight transport.

Techniques to improve accuracy could similarly help harden models. For example, dropout can improve accuracy when used during training. It is particularly useful where there is limited training data and over-fitting is more likely to occur. Wang *et al.* [130] propose hardening DNN using defensive dropout at test time. Unfortunately, there is inherently a trade-off between defensive dropout and test accuracy; however, a relatively small decrease in test accuracy can significantly reduce the success rate of attacks. Such hardening techniques force successful attacks to use larger perturbations, which in turn may be more readily recognized as adversarial.

Defenses that block gradient-based attacks complicate the generation of adversarial examples; however, like adversarial training these defenses could be bypassed. In particular black-box attacks and transferability-based attacks are not blocked by such defenses. A more promising defense "Defensive dropout" can block both black-box and transferability-based attacks.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

5.3.4. Detecting Adversarial Examples

Much research has considered the best way to detect adversarial examples. If adversarial examples can be detected they could be more easily deflected, and perhaps even the original input could be salvaged and correctly classified. Grosse *et al*. [120] propose a statistical test to detect adversarial examples before they are input into machine learning models. They observe adversarial examples are unrepresentative of the distribution, and lie in unexpected regions of a model's output surface. Their proposed outlier detection system relies on the statistical separation of adversarial examples. They subsequently evaluate their model against adaptive strategies and strong black-box strategies.

Metzen *et al.* [121] propose a binary classifier "Detector Subnetwork" aiming to distinguish between genuine data and adversarial examples. The detection of adversarial examples does not unequivocally lead to correct classification; however, the effect of adversarial examples could perhaps be mitigated through fallback solutions. For example, by requesting human intervention. After successfully detecting adversarial examples in their experiments, they later bypassed their defenses by generating adversarial examples that fool both detector and classifier. They further propose a training procedure called 'dynamic adversary training' as a countermeasure to their attack against the detector.

Feiman *et al.* [122] also detect adversarial examples by considering which artifacts of adversarial examples could help detection. They consider two complementary features used to detect adversarial examples: Density estimates and Bayesian uncertainty estimates. They evaluate these features on CNNs trained on MNIST and CIFAR-10 datasets. They effectively detect adversarial examples with ROC-AUC of 92.6%. They further suggest that their method could be used in RNNs. This suggestion is bolstered by Gal and Ghahramani's [140] assertion that Bayesian approximation using dropout can be applied to RNN networks.

Meng *et al.* [123] propose a framework 'MagNET' to detect adversarial examples. This framework precedes the classifier it defends. The framework has two components: 1) A detector finds and discards any out-of-distribution examples (those significantly far from the manifold boundary). 2) A reformer that aims to find close approximations to inputs before forwarding the approximations to the classifier. Their system generalizes well because it learns to detect adversarial examples without knowledge of how they were generated. They propose a defense against gray-box attacks where the adversary has knowledge of the deployed defenses. The proposed defense trains a number of auto-encoders (or *reformers*). At test-time a single auto-encoder is selected at random.

Xu *et al.* [124] propose 'Feature Squeezing' as a strategy to detect adversarial examples by squeezing out unnecessary features in the input. Through comparing predictions of the original and feature squeezed inputs, adversarial examples are identified if the difference between the two predictions meets a threshold. Two feature-squeezing methods are used: 1) Reducing the colour bit-depth of the image. 2) Spatial smoothing. An adversary may adapt and circumvent this defense; however, the defense may frustrate the adversary because it changes the problem the adversary must overcome.

Rosenberg *et al.* [132] consider the feature squeezing defense designed for CNNs and propose 'Sequence Squeezing' which is adapted for RNNs. Adversarial examples are similarly detected by running the classifier twice: once on the original sequence, and once for the sequence-squeezed input. An input is identified as adversarial if the difference in the confidence scores meets a threshold value.

Zhang *et al.* [137] propose an image classifier based on a variational auto-encoder. They train two models each on half the dataset: a target model and a surrogate model. On the surrogate model they generate three types of strong transfer-based adversarial examples: L_0, L_2, and L_∞. Analysis of their model using the CIFAR-10, MNIST, and Fashion-MNIST datasets found their model achieves state-of-the-art accuracy with significantly better robustness. Their work is in the visual domain; however, perhaps their ideas can be applied to other domains such as intrusion detection.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

We have discussed some architectural defenses against adversarial examples. In particular, we have considered methods for detecting adversarial examples. Carlini & Wagner [141] show Adversarial examples are harder to detect and that adversarial examples do not exhibit intrinsic properties. Moreover, many detection methods can be broken by choosing good attacker-loss functions. Grosse et al. [120] note adversarial defenses exist within an arms race and that guarantees against future attacks are difficult because adversaries may adapt to the defenses by adopting new strategies. Meng *et al.* [123] advocate that defenses against adversarial examples should be indepedent of any particular attack. We have seen that Human-in-the-loop solutions could be useful where few cases need human intervention; however repeated requests might quickly overwhelm human operators given large numbers of adversarial examples. For example, as might be seen in network traffic analysis.

5.3.5. Defensive Testing

Adversarial examples cause unexpected behaviour. Recent research considers testing deep learning systems. Pei *et al.* [142] aim to discover unusual or unexpected behaviour of a neural network through systematic testing. They produce test data by solving a joint optimization problem. Their tests aim to trigger different behaviours and activate a high proportion of neurons in a neural network. Their method finds corner-cases where incorrect behaviour is exhibited. For example, malware masquerading as benign. They claim to expose more inputs and types of unexpected behaviour than adversarial examples. They further use the generated inputs to perform adversarial training. As a defense we question the practicability of triggering all neurons in larger neural networks; however, as an attack, their method could produce different types of adversarial inputs.

Other researchers are considering similar techniques to generate test data. Tian *et al.* [128] evaluate a tool for automatically detecting erroneous behaviour, generating test inputs designed to maximise the number of activated neurons using realistic driving conditions including: blurring, rain, and fog. Zhang *et al.* [129] propose a system to automatically synthesize large amounts of diverse driving scenes, including weather conditions using GANs. We consider GANs useful for generating adversarial inputs. GANs should implicitly learn domain constraints.

5.3.6. Multi-Classifier Systems

Biggio *et al.* [99] Highlight that robustness against adversarial examples can be improved by careful use of ensemble classifiers. For example by using rejection-based mechanisms. Indeed Biggio *et al.* had implemented a multi-classifier system (MCS) [143] which was hardened using randomisation. Randomising the decision boundary makes a classifier harder to evade. Since the attacker has less information on the exact position of a decision boundary, they must make too conservative or too risky choices when generating adversarial examples.

5.3.7. Game Theory

Zhou *et al.*[135] consider game theoretic modeling of adversarial machine learning problem. Many different models have been proposed. Some aim to optimise the feature set using a set of high-quality features, thus making adversarial attacks more difficult. Game theoretic models are proposed to address more complex situations with many adversaries of different types. Equilibrium strategies are acceptable to both players and neither has an incentive to change. Therefore, assuming rational adversaries, game theory-based approaches allowing a Nash equilibrium could potentially end the evolutionary arms race.

5.3.8. Adversarial Example Defenses in Cybersecurity Domains

We discussed domains in sections 5.2.4 and 5.2.5. Most research on defenses against adversarial examples has focused on the visual-domain. Comparatively little research has so far considered defenses in cybersecurity domains such as intrusion detection and

 doi:10.20944/preprints202202.0099.v1

malware analysis. Applying current defenses in the visual-domain to other domains might efficiently kick-start research into defenses for other domains. Effective defenses against adversarial examples could help enable the use of ML models in cybersecurity and other adversarial environments.

Different model types are more suited to domains. We consider that different model types may require different defenses. Again, we classify models into four types: MLP, CNN, RNN, and RF.

6. Discussion

We have conducted an extensive survey of the academic literature in relation to adversarial machine learning, and we have derived a classification based on both attack and defense. We now discuss the trends and research challenges as identified through our investigation, exploring topics where research is comprehensively covered, and research areas that are still in their infancy.

Machine learning and adversarial learning are becoming increasingly recognised by the research community, given the rapid uptake of ML models in a whole host of application domains. To put this in context, 2,975 papers were published on arXiv in the last 12 months (October 2020 - September 2021) related to machine learning and adversarial learning. Over recent years, the number of papers being published on this topic has grown substantially. According to Carlini who maintains a blog post 'A Complete List of All (arXiv) Adversarial Example Papers' [144] the cumulative number of adversarial example papers nears 4000 in the year 2021. It is therefore evident that there is a lot of interest and many researchers active in this area. Not all papers in this list are useful or relevant, we pass no judgement of their quality but merely aim to clarify the research landscape and draw important research to the fore. The majority of prior research has been applied to the visual domain. Seminal contributions have been made by Szegedy *et al.* [15], Goodfellow *et al.* [58], Carlini *et al.* [59], Papernot [73]. It is clear that the visual domain continues to be well researched.

We now discuss future research challenges. Few researchers have addressed the problem of transferability, which remains a key area of concern because hard-to-attack models are nevertheless susceptible to *transferable* adversarial examples generated against easy-to-attack models. Breaking the transferability of adversarial examples is a key challenge for the research community. Currently, defensive dropout [130] at test time is a promising defense. Adversarial example detection is also a useful area of research.

We consider the area of functionality-preserving adversarial examples is under-explored. Research into improving robustness against such adversarial examples is an area that requires urgent research. We suggest adapting defenses used in the visual domain and CNN models to other model types such as RNNs could offer potential solutions. Caution should be exercised when adapting defenses in the visual domain to other domains. For example, denoising defenses may not apply directly to discrete or noncontinuous data.

Constraints on adversarial examples are not limited to preserving the functionality of malware or IDS attacks. CPSs model the real world where linear and other physical constraints must be respected. Adversarial examples that do not respect domain constraints risk marking themselves as obvious anomalies.

Concept-drift is a real concern for cybersecurity [1], as new attacks and techniques are discovered daily. As the model and the current state of the art diverge, the model suffers from hidden technical debt. Therefore, the model must be retrained to reflect the current state-of-the-art attacks and new network traffic patterns [145]. Researchers might develop and use more up-to-date datasets. Further avenues for research include semi-supervised/unsupervised ML and active learning methods that continuously update the underlying model and do not rely on labelled datasets.

We prioritise the areas of future research, setting the agenda for research in this area. Critical areas of research include: breaking the transferability of adversarial examples that would hopefully be applicable across domains. Non-visual domains including cybersecurity and cyber physical systems have been under-explored and this oversight should

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0099.v1

be rectified urgently. Further research on transformations in nonvisual domains could provide useful knowledge. Detection of adversarial examples and pushing the fields of cybersecurity, intrusion detection, and cyber-physical systems will yield benefits beyond cybersecurity and may be applicable in other nonvisual domains. Moreover, research is required in areas beyond instance classifiers. Areas of RNNs and reinforcement learning have been under-explored. More research is required to understand the use of domain constraints and functionality-preserving adversarial examples. Further research is needed towards effective countermeasures.

6.1. Takeaways

We offer the following key takeaways: There is a need for better robustness metrics. Some researchers simply state accuracy, others might state the better F1-Score; however F1-Score is biased by unbalanced datasets which are widespread in intrusion detection partly due to large numbers of benign samples. Using F1-Score could lead to a false sense of security. Researchers should adopt stronger metrics such as CLEVER[54] or Empirical Robustness[57].

Adversarial machine learning is a critical area of research. If not addressed, there is increasing potential for novel attack strategies that seek to exploit the inherent weaknesses that exist within machine learning models; however, few works consider *'realisable'* perturbations that take account of domain and/or real-world constraints. Successful adversarial examples must be crafted to comply with domain and real-world constraints. This may be challenging since even small modifications may corrupt network packets that are likely to be dropped by firewalls. This necessitates functionality preservation in adversarial learning.

We propose that human perception may not be the best criterion for analyzing adversarial examples. In cybersecurity domains we propose adversarial examples must preserve functionality. Traditionally, adversarial examples are thought of as having imperceptible noise. That is, that humans cannot perceive the difference between the original and perturbed inputs. Indeed, human perception in some domains might be immaterial. For example, strategic attacks triggered at crucial moments might cause damage to CPS before any human could reasonably act.

In cyber-security domains traditional gradient descent algorithms may be insufficient; although JSMA may be reasonable because it perturbs few features. Stringent constraints exist in the cyber-security domain and extreme care must be taken to create valid adversarial examples. We offer some guidelines for generating functionality-preserving adversarial examples. Functionality-preserving adversarial examples should: only perform legitimate transformations; respect mathematical dependencies, real-world, and domain constraints; minimize the number of perturbed features, restrict modification to non-critical features; and where possible retain the original payload and/or packet order.

Defences against adversarial examples must consider that adversaries are likely to adapt by adopting new strategies. Many researchers propose adversarial training to improve robustness. Adversarial training is a simple method aiming to improve robustness; however, it is potentially a cosmetic solution: the problem of adversarial examples cannot be solved only through ever greater numbers of adversarial examples in the training data. Adversarial training, if used, must be bolstered by other defenses. Interesting defence strategies include randomisation: randomising decision boundaries makes evasion more difficult because attackers have less information on the exact position of a decision boundary. They must therefore make too conservative or too risky choices when generating adversarial examples.

Game theoretic models could be used to address more complex situations with many adversaries of different types as found in intrusion detection. Equilibrium strategies acceptable to both defender and adversary mean neither has an incentive to change. Therefore, assuming rational opponents, game theory-based approaches allowing a Nash equilibrium

 doi:10.20944/preprints202202.0099.v1

could potentially end the evolutionary arms race. Although it is difficult to conceive a world where no advantage is possible.

Current promising defenses like dropout exchange a relatively small decrease in accuracy for significant reduction of successful attacks, even successfully blocking black-box and transferability-based attacks. Hardening techniques force successful attacks to use larger perturbations, which in turn may be more readily recognized as adversarial.

In a broader cybersecurity context, risks arising from adversarial examples are not yet fully understood. Furthermore, algorithms and models from other domains may not readily apply because of distributed sensors and inherent real-world constraints. It is uncertain whether current defences are sufficient. Furthermore, adversarial example detectors must function efficiently in a real-time monitoring environment while maintaining low false alarm rates.

Many academic researchers use old datasets which do not fairly represent modern network traffic analysis problems due to concept-drift. Problems of labelling data and retraining systems provide an impetus to explore unsupervised and active learning. Unfortunately adversarial attacks are possible on active learning systems [146]. Lin et al. [76] describe an enchanting attack to lure a machine learning system to a target state through crafting a series of adversarial examples. It is conceivable that similar attacks could lure anomaly detection systems towards normalizing and accepting malicious traffic.

We offer these insights and hope that this survey offers other researchers a base for exploring the areas of robustness and functionality-preserving adversarial examples.

Author Contributions:

Conceptualization, Andrew McCarthy, Panagiotis Andriotis, Essam Ghadafi, and Phil Legg.; methodology, Andrew McCarthy.; formal analysis, Andrew McCarthy.; investigation, Andrew McCarthy.; writing—original draft preparation, Andrew McCarthy.; writing—review and editing, Andrew McCarthy, Panagiotis Andriotis, Essam Ghadafi, and Phil Legg. ; visualization, Andrew McCarthy.; supervision, Panagiotis Andriotis, Essam Ghadafi, and Phil Legg.; funding acquisition, Phil Legg. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Partnership PhD scheme at the University of the West of England in collaboration with Techmodal Ltd.

Data Availability Statement: N/A

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Andresini, G.; Pendlebury, F.; Pierazzi, F.; Loglisci, C.; Appice, A.; Cavallaro, L. INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection. In Proceedings of the Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec). ACM, 2021, p. 0.
2. Raghuraman, C.; Suresh, S.; Shivshankar, S.; Chapaneri, R. Static and dynamic malware analysis using machine learning. In Proceedings of the First International Conference on Sustainable Technologies for Computational Intelligence. Springer, 2020, pp. 793–806.
3. Berger, H.; Hajaj, C.; Dvir, A. Evasion Is Not Enough: A Case Study of Android Malware. In Proceedings of the International Symposium on Cyber Security Cryptography and Machine Learning. Springer, 2020, pp. 167–174.
4. Hou, R.; Xiang, X.; Zhang, Q.; Liu, J.; Huang, T. Universal Adversarial Perturbations of Malware. In Proceedings of the International Symposium on Cyberspace Safety and Security. Springer, 2020, pp. 9–19.
5. Parshutin, S.; Kirshners, A.; Kornijenko, Y.; Zabiniako, V.; Gasparovica-Asite, M.; Rozkalns, A. Classification with LSTM Networks in User Behaviour Analytics with Unbalanced Environment. *Automatic Control and Computer Sciences* **2021**, *55*, 85–91.
6. Le, D.C.; Zincir-Heywood, N. Exploring anomalous behaviour detection and classification for insider threat identification. *International Journal of Network Management* **2021**, *31*, e2109.
7. Biswal, S. Real-Time Intelligent Vishing Prediction and Awareness Model (RIVPAM). In Proceedings of the 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). IEEE, 2021, pp. 1–2.
8. Kumar, N.; Sonowal, S.; et al. Email Spam Detection Using Machine Learning Algorithms. In Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2020, pp. 108–113.
9. Kiela, D.; Firooz, H.; Mohan, A.; Goswami, V.; Singh, A.; Ringshia, P.; Testuggine, D. The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790* **2020**.

10. Bin Naeem, S.; Kamel Boulos, M.N. COVID-19 misinformation online and health literacy: A brief overview. *International Journal of Environmental Research and Public Health* **2021**, *18*, 8091.
11. Coan, T.; Boussalis, C.; Cook, J.; Nanko, M. Computer-assisted detection and classification of misinformation about climate change. *SocArXiv* **2021**.
12. Khanam, Z.; Alwasel, B.; Sirafi, H.; Rashid, M. Fake News Detection Using Machine Learning Approaches. In Proceedings of the IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, Vol. 1099, p. 012040.
13. Gu, X.; Easwaran, A. Towards Safe Machine Learning for CPS: Infer Uncertainty from Training Data. In Proceedings of the Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems; Association for Computing Machinery: New York, NY, USA, 2019; ICCPS '19, p. 249–258. doi:10.1145/3302509.3311038.
14. Ghafouri, A.; Vorobeychik, Y.; Koutsoukos, X. Adversarial regression for detecting attacks in cyber-physical systems. In Proceedings of the International Joint Conference on Artificial Intelligence, 2018.
15. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations, ICLR 2014, 2014, p. 0. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014.
16. Wardle, S.G.; Taubert, J.; Teichmann, L.; Baker, C.I. Rapid and dynamic processing of face pareidolia in the human brain. *Nature communications* **2020**, *11*, 1–14.
17. Summerfield, C.; Egner, T.; Mangels, J.; Hirsch, J. Mistaking a house for a face: neural correlates of misperception in healthy humans. *Cerebral cortex* **2006**, *16*, 500–508.
18. Huang, Y.; Verma, U.; Fralick, C.; Infantec-Lopez, G.; Kumar, B.; Woodward, C. Malware Evasion Attack and Defense. In Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2019, pp. 34–38. doi:10.1109/DSN-W.2019.00014.
19. Ayub, M.A.; Johnson, W.A.; Talbert, D.A.; Siraj, A. Model Evasion Attack on Intrusion Detection Systems using Adversarial Machine Learning. In Proceedings of the 2020 54th Annual Conference on Information Sciences and Systems (CISS), 2020, pp. 1–6. doi:10.1109/CISS48834.2020.1570617116.
20. Satter, R. Experts who wrestled with SolarWinds hackers say cleanup could take months - or longer, 2020.
21. Sirota, S. Air Force response to SolarWinds hack: Preserve commercial partnerships, improve transparency into security efforts. *Inside Cybersecurity* **2021**. Name - Department of Defense; Copyright - Copyright Inside Washington Publishers Jan 12, 2021; Last updated - 2021-01-13.
22. Corona, I.; Giacinto, G.; Roli, F. Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. *Information Sciences* **2013**, *239*, 201–225.
23. Hankin, C.; Barrère, M. Trustworthy Inter-connected Cyber-Physical Systems. In Proceedings of the International Conference on Critical Information Infrastructures Security. Springer, 2020, pp. 3–13.
24. Cho, J.H.; Xu, S.; Hurley, P.M.; Mackay, M.; Benjamin, T.; Beaumont, M. Stram: Measuring the trustworthiness of computer-based systems. *ACM Computing Surveys (CSUR)* **2019**, *51*, 1–47.
25. Papernot, N.; McDaniel, P.; Sinha, A.; Wellman, M.P. Sok: Security and privacy in machine learning. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2018, pp. 399–414.
26. Zhang, J.; Li, C. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems* **2019**.
27. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *Digital Threats: Research and Practice* **2021**, *0*. doi:10.1145/3469659.
28. Shannon, C.E. Communication theory of secrecy systems. *The Bell System Technical Journal* **1949**, *28*, 656–715. doi:10.1002/j.1538-7305.1949.tb00928.x.
29. Taran, O.; Rezaeifar, S.; Voloshynovskiy, S. Bridging machine learning and cryptography in defence against adversarial attacks. In Proceedings of the Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.
30. Wu, Y.; Wei, D.; Feng, J. Network attacks detection methods based on deep learning techniques: a survey. *Security and Communication Networks* **2020**, *2020*.
31. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE, 2009, pp. 1–6.
32. McHugh, J. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)* **2000**, *3*, 262–294.
33. Cerf, V.G. 2021 Internet Perspectives. *IEEE Network* **2021**, *35*, 3–3.
34. McKeay, M. Akamai state of the Internet / security: A Year in Review. http://akamai.com/soti, 2020.
35. Kok, S.; Abdullah, A.; Jhanjhi, N.; Supramaniam, M. A review of intrusion detection system using machine learning approach. *International Journal of Engineering Research and Technology* **2019**, *12*, 8–15.
36. Alatwi, H.A.; Morisset, C. Adversarial Machine Learning In Network Intrusion Detection Domain: A Systematic Review. *arXiv e-prints* **2021**, pp. arXiv–2112.
37. Revathi, S.; Malathi, A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)* **2013**, *2*, 1848–1853.
38. Gharaibeh, M.; Papadopoulos, C. DARPA 2009 intrusion detection dataset. *Colorado State Univ., Tech. Rep* **2014**.

39. Garcia, S.; Grill, M.; Stiborek, J.; Zunino, A. An empirical comparison of botnet detection methods. *computers & security* **2014**, *45*, 100–123.
40. Song, J.; Takakura, H.; Okabe, Y.; Eto, M.; Inoue, D.; Nakao, K. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In Proceedings of the Proceedings of the first workshop on building analysis datasets and gathering experience returns for security, 2011, pp. 29–36.
41. Moustafa, N.; Slay, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 military communications and information systems conference (MilCIS). IEEE, 2015, pp. 1–6.
42. Almomani, I.; Al-Kasasbeh, B.; Al-Akhras, M. WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors* **2016**, *2016*.
43. Niyaz, Q.; Sun, W.; Javaid, A.Y. A deep learning based DDoS detection system in software-defined networking (SDN). *arXiv preprint arXiv:1611.07400* **2016**.
44. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **2018**, *1*, 108–116.
45. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the mirai botnet. In Proceedings of the 26th {USENIX} security symposium ({USENIX} Security 17), 2017, pp. 1093–1110.
46. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems* **2019**, *100*, 779–796.
47. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089* **2018**.
48. Janusz, A.; Kałuza, D.; Chądzyńska-Krasowska, A.; Konarski, B.; Holland, J.; Ślęzak, D. IEEE BigData 2019 cup: suspicious network event recognition. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019, pp. 5881–5887.
49. Ferriyan, A.; Thamrin, A.H.; Takeda, K.; Murai, J. Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic. *Applied Sciences* **2021**, *11*. doi:10.3390/app11177868.
50. Martins, N.; Cruz, J.M.; Cruz, T.; Abreu, P.H. Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. *IEEE Access* **2020**, *8*, 35403–35419.
51. Shafique, M.; Naseer, M.; Theocharides, T.; Kyrkou, C.; Mutlu, O.; Orosa, L.; Choi, J. Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead. *IEEE Design & Test* **2020**, *37*, 30–57.
52. Labaca-Castro, R.; Biggio, B.; Dreo Rodosek, G. Poster: Attacking malware classifiers by crafting gradient-attacks that preserve functionality. In Proceedings of the Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2565–2567.
53. Bai, T.; Luo, J.; Zhao, J.; Wen, B. Recent Advances in Adversarial Training for Adversarial Robustness. *arXiv e-prints* **2021**, pp. arXiv–2102.
54. Weng, T.W.; Zhang, H.; Chen, P.Y.; Yi, J.; Su, D.; Gao, Y.; Hsieh, C.J.; Daniel, L. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578* **2018**.
55. Goodfellow, I. Gradient masking causes clever to overestimate adversarial perturbation size. *arXiv preprint arXiv:1804.07870* **2018**.
56. Weng, T.W.; Zhang, H.; Chen, P.Y.; Lozano, A.; Hsieh, C.J.; Daniel, L. On extensions of clever: A neural network robustness evaluation algorithm. In Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2018, pp. 1159–1163.
57. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, p. 0.
58. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**.
59. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 ieee symposium on security and privacy (sp). IEEE, 2017, pp. 39–57.
60. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; et al. Technical report on the cleverhans v2. 1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768* **2016**.
61. Rauber, J.; Brendel, W.; Bethge, M. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131* **2017**.
62. Nicolae, M.I.; Sinn, M.; Tran, M.N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; et al. Adversarial Robustness Toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069* **2018**.
63. Ding, G.W.; Wang, L.; Jin, X. AdverTorch v0. 1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623* **2019**.
64. Lashkari, A.H.; Zang, Y.; Owhuo, G.; Mamun, M.; Gil, G. CICFlowMeter. https://www.unb.ca/cic/research/applications.html, 2017.
65. Habibi Lashkari., A.; Draper Gil., G.; Mamun., M.S.I.; Ghorbani., A.A. Characterization of Tor Traffic using Time based Features. In Proceedings of the Proceedings of the 3rd International Conference on Information Systems Security and Privacy - ICISSP,. INSTICC, SciTePress, 2017, pp. 253–262. doi:10.5220/0006105602530262.

66. Draper-Gil., G.; Lashkari., A.H.; Mamun., M.S.I.; A. Ghorbani., A. Characterization of Encrypted and VPN Traffic using Time-related Features. In Proceedings of the Proceedings of the 2nd International Conference on Information Systems Security and Privacy - ICISSP,. INSTICC, SciTePress, 2016, pp. 407–414. doi:10.5220/0005740704070414.
67. Almomani, O. A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. *Symmetry* **2020**, *12*, 1046.
68. McCarthy, A.; Andriotis, P.; Ghadafi, E.; Legg, P. Feature Vulnerability and Robustness Assessment against Adversarial Machine Learning Attacks. In Proceedings of the 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2021, pp. 1–8. doi:10.1109/CyberSA52016.2021.9478199.
69. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE symposium on security and privacy (SP). IEEE, 2016, pp. 582–597.
70. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, [https://www.bmj.com/content/372/bmj.n71.full.pdf]. doi:10.1136/bmj.n71.
71. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion attacks against machine learning at test time. In Proceedings of the Joint European conference on machine learning and knowledge discovery in databases. Springer, 2013, pp. 387–402.
72. Papernot, N.; McDaniel, P.; Swami, A.; Harang, R. Crafting adversarial input sequences for recurrent neural networks. In Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference. IEEE, 2016, pp. 49–54.
73. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS P), 2016, pp. 372–387.
74. Jia, R.; Liang, P. Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2021–2031.
75. Zhao, Z.; Dua, D.; Singh, S. Generating Natural Adversarial Examples. In Proceedings of the International Conference on Learning Representations, 2018, p. 0.
76. Lin, Y.C.; Hong, Z.W.; Liao, Y.H.; Shih, M.L.; Liu, M.Y.; Sun, M. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748* **2017**.
77. Rigaki, M. Adversarial deep learning against intrusion detection classifiers, 2017.
78. Hu, W.; Tan, Y. Black-box attacks against RNN based malware detection algorithms. In Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, p. 0.
79. Homoliak, I.; Teknös, M.; Ochoa, M.; Breitenbacher, D.; Hosseini, S.; Hanacek, P. Improving Network Intrusion Detection Classifiers by Non-payload-Based Exploit-Independent Obfuscations: An Adversarial Approach. *EAI Endorsed Transactions on Security and Safety* **2018**, *5*.
80. Rosenberg, I.; Shabtai, A.; Rokach, L.; Elovici, Y. Generic black-box end-to-end attack against state of the art API call based malware classifiers. In Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, 2018, pp. 490–510.
81. Wang, Z. Deep learning-based intrusion detection with adversaries. *IEEE Access* **2018**, *6*, 38367–38384.
82. Warzyński, A.; Kołaczek, G. Intrusion detection systems vulnerability on adversarial examples. In Proceedings of the 2018 Innovations in Intelligent Systems and Applications (INISTA). IEEE, 2018, pp. 1–4.
83. Lin, Z.; Shi, Y.; Xue, Z. Idsgan: Generative adversarial networks for attack generation against intrusion detection. *arXiv preprint arXiv:1809.02077* **2018**.
84. Yang, K.; Liu, J.; Zhang, C.; Fang, Y. Adversarial examples against the deep learning based network intrusion detection systems. In Proceedings of the MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM). IEEE, 2018, pp. 559–564.
85. Su, J.; Vargas, D.V.; Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* **2019**, *23*, 828–841.
86. Kuppa, A.; Grzonkowski, S.; Asghar, M.R.; Le-Khac, N.A. Black box attacks on deep anomaly detectors. In Proceedings of the Proceedings of the 14th International Conference on Availability, Reliability and Security, 2019, pp. 1–10.
87. Ibitoye, O.; Shafiq, O.; Matrawy, A. Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019, pp. 1–6.
88. Usama, M.; Asim, M.; Latif, S.; Qadir, J.; et al. Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems. In Proceedings of the 2019 15th international wireless communications & mobile computing conference (IWCMC). IEEE, 2019, pp. 78–83.
89. Huang, W.; Peng, X.; Shi, Z.; Ma, Y. Adversarial Attack against LSTM-based DDoS Intrusion Detection System. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2020, pp. 686–693.
90. Ogawa, Y.; Kimura, T.; Cheng, J. Vulnerability Assessment for Machine Learning Based Network Anomaly Detection System. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2020, pp. 1–2.
91. Chen, J.; Gao, X.; Deng, R.; He, Y.; Fang, C.; Cheng, P. Generating Adversarial Examples against Machine Learning based Intrusion Detector in Industrial Control Systems. *IEEE Transactions on Dependable and Secure Computing* **2020**.
92. Demetrio, L.; Biggio, B.; Lagorio, G.; Roli, F.; Armando, A. Functionality-preserving black-box optimization of adversarial windows malware. *IEEE Transactions on Information Forensics and Security* **2021**, *16*, 3469–3478.

93. Labaca-Castro, R.; Muñoz-González, L.; Pendlebury, F.; Rodosek, G.D.; Pierazzi, F.; Cavallaro, L. Universal Adversarial Perturbations for Malware. *arXiv preprint arXiv:2102.06747* **2021**.
94. Cai, F.; Li, J.; Koutsoukos, X. Detecting adversarial examples in learning-enabled cyber-physical systems using variational autoencoder for regression. In Proceedings of the 2020 IEEE Security and Privacy Workshops (SPW). IEEE, 2020, pp. 208–214.
95. Li, J.; Yang, Y.; Sun, J.S.; Tomsovic, K.; Qi, H. Conaml: Constrained adversarial machine learning for cyber-physical systems. In Proceedings of the Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, 2021, pp. 52–66.
96. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, 2018, p. 0.
97. Kurakin, A.; Goodfellow, I.; Bengio, S.; Dong, Y.; Liao, F.; Liang, M.; Pang, T.; Zhu, J.; Hu, X.; Xie, C.; et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*; Springer, 2018; pp. 195–231.
98. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. A General Framework for Adversarial Examples with Objectives. *ACM Trans. Priv. Secur.* **2019**, *22*. doi:10.1145/3317611.
99. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* **2018**, *84*, 317–331.
100. Gilmer, J.; Adams, R.P.; Goodfellow, I.; Andersen, D.; Dahl, G.E. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732* **2018**.
101. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *Digital Threats: Research and Practice* **2021**.
102. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1765–1773.
103. Sheatsley, R.; Papernot, N.; Weisman, M.; Verma, G.; McDaniel, P. Adversarial Examples in Constrained Domains. *arXiv preprint arXiv:2011.01183* **2020**.
104. Chernikova, A.; Oprea, A. Fence: Feasible evasion attacks on neural networks in constrained environments. *arXiv preprint arXiv:1909.10480* **2019**.
105. Apruzzese, G.; Colajanni, M.; Marchetti, M. Evaluating the effectiveness of adversarial attacks against botnet detectors. In Proceedings of the 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA). IEEE, 2019, pp. 1–8.
106. Hashemi, M.J.; Cusack, G.; Keller, E. Towards evaluation of nidss in adversarial setting. In Proceedings of the Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, 2019, pp. 14–21.
107. Papadopoulos, P.; Essen, O.T.v.; Pitropakis, N.; Chrysoulas, C.; Mylonas, A.; Buchanan, W.J. Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT. *Journal of Cybersecurity and Privacy* **2021**, *1*, 252–273.
108. Anthi, E.; Williams, L.; Rhode, M.; Burnap, P.; Wedgbury, A. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *Journal of Information Security and Applications* **2021**, *58*, 102717.
109. Han, D.; Wang, Z.; Zhong, Y.; Chen, W.; Yang, J.; Lu, S.; Shi, X.; Yin, X. Evaluating and Improving Adversarial Robustness of Machine Learning-Based Network Intrusion Detectors. *IEEE Journal on Selected Areas in Communications* **2021**.
110. Duy, P.T.; Khoa, N.H.; Nguyen, A.G.T.; Pham, V.H.; et al. DIGFuPAS: Deceive IDS with GAN and Function-Preserving on Adversarial Samples in SDN-enabled networks. *Computers & Security* **2021**, p. 102367.
111. Alhajjar, E.; Maxwell, P.; Bastian, N. Adversarial machine learning in network intrusion detection systems. *Expert Systems with Applications* **2021**, *186*, 115782.
112. Han, D.; Wang, Z.; Zhong, Y.; Chen, W.; Yang, J.; Lu, S.; Shi, X.; Yin, X. Practical traffic-space adversarial attacks on learning-based nidss.
113. Wang, J.; Pan, J.; AlQerm, I.; Liu, Y. Def-IDS: An Ensemble Defense Mechanism Against Adversarial Attacks for Deep Learning-based Network Intrusion Detection. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN). IEEE, 2021, pp. 1–9.
114. Ali, H.; Khan, M.S.; AlGhadhban, A.; Alazmi, M.; Alzamil, A.; Al-utaibi, K.; Qadir, J. Analyzing the Robustness of Fake-news Detectors under Black-box Adversarial Attacks. *IEEE Access* **2021**.
115. Chenou, J.; Hsieh, G.; Fields, T. Radial Basis Function Network: Its Robustness and Ability to Mitigate Adversarial Examples. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2019, pp. 102–106.
116. Wei, W.; Liu, L.; Loper, M.; Truex, S.; Yu, L.; Gursoy, M.E.; Wu, Y. Adversarial examples in deep learning: Characterization and divergence. *arXiv preprint arXiv:1807.00051* **2018**.
117. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing machine learning models via prediction apis. In Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 601–618.
118. Lillicrap, T.P.; Cownden, D.; Tweed, D.B.; Akerman, C.J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications* **2016**, *7*, 1–10.
119. Graese, A.; Rozsa, A.; Boult, T.E. Assessing Threat of Adversarial Examples on Deep Neural Networks. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 69–74. doi:10.1109/ICMLA.2016.0020.

120. Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280* **2017**.
121. Metzen, J.H.; Genewein, T.; Fischer, V.; Bischoff, B. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267* **2017**.
122. Feinman, R.; Curtin, R.R.; Shintre, S.; Gardner, A.B. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410* **2017**.
123. Meng, D.; Chen, H. Magnet: a two-pronged defense against adversarial examples. In Proceedings of the Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, 2017, pp. 135–147.
124. Xu, W.; Evans, D.; Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155* **2017**.
125. Lee, H.; Han, S.; Lee, J. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387* **2017**.
126. Samangouei, P.; Kabkab, M.; Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605* **2018**.
127. Dhillon, G.S.; Azizzadenesheli, K.; Lipton, Z.C.; Bernstein, J.; Kossaifi, J.; Khanna, A.; Anandkumar, A. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442* **2018**.
128. Tian, Y.; Pei, K.; Jana, S.; Ray, B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the Proceedings of the 40th international conference on software engineering, 2018, pp. 303–314.
129. Zhang, M.; Zhang, Y.; Zhang, L.; Liu, C.; Khurshid, S. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In Proceedings of the 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2018, pp. 132–142.
130. Wang, S.; Wang, X.; Zhao, P.; Wen, W.; Kaeli, D.; Chin, P.; Lin, X. Defensive dropout for hardening deep neural networks under adversarial attacks. In Proceedings of the Proceedings of the International Conference on Computer-Aided Design, 2018, pp. 1–8.
131. Amer, M.; Maul, T. Weight Map Layer for Noise and Adversarial Attack Robustness. *arXiv preprint arXiv:1905.00568* **2019**.
132. Rosenberg, I.; Shabtai, A.; Elovici, Y.; Rokach, L. Defense methods against adversarial examples for recurrent neural networks. *arXiv preprint arXiv:1901.09963* **2019**.
133. Apruzzese, G.; Andreolini, M.; Marchetti, M.; Venturi, A.; Colajanni, M. Deep reinforcement adversarial learning against botnet evasion attacks. *IEEE Transactions on Network and Service Management* **2020**, *17*, 1975–1987.
134. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Marchetti, M. Addressing adversarial attacks against security systems based on machine learning. In Proceedings of the 2019 11th International Conference on Cyber Conflict (CyCon). IEEE, 2019, Vol. 900, pp. 1–18.
135. Zhou, Y.; Kantarcioglu, M.; Xi, B. A survey of game theoretic approach for adversarial machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2019**, *9*, e1259.
136. Apruzzese, G.; Andreolini, M.; Colajanni, M.; Marchetti, M. Hardening random forest cyber detectors against adversarial attacks. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2020**, *4*, 427–439.
137. Zhang, C.; Tang, Z.; Zuo, Y.; Li, K.; Li, K. A robust generative classifier against transfer attacks based on variational auto-encoders. *Information Sciences* **2021**, *550*, 57–70.
138. Wang, N.; Chen, Y.; Hu, Y.; Lou, W.; Hou, Y.T. MANDA: On Adversarial Example Detection for Network Intrusion Detection System. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE, 2021, pp. 1–10.
139. Song, D.; Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Tramer, F.; Prakash, A.; Kohno, T. Physical adversarial examples for object detectors. In Proceedings of the 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18), 2018, p. 0.
140. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the international conference on machine learning. PMLR, 2016, pp. 1050–1059.
141. Carlini, N.; Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 3–14.
142. Pei, K.; Cao, Y.; Yang, J.; Jana, S. Deepxplore: Automated whitebox testing of deep learning systems. In Proceedings of the proceedings of the 26th Symposium on Operating Systems Principles, 2017, pp. 1–18.
143. Biggio, B.; Fumera, G.; Roli, F. Adversarial pattern classification using multiple classifiers and randomisation. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, 2008, pp. 500–509.
144. Carlini, N. A complete list of all (arxiv) adversarial example papers. https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html, 2019.
145. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.F.; Dennison, D. Hidden technical debt in machine learning systems. In Proceedings of the Advances in neural information processing systems, 2015, pp. 2503–2511.
146. Shu, D.; Leslie, N.O.; Kamhoua, C.A.; Tucker, C.S. Generative adversarial attacks against intrusion detection systems using active learning. In Proceedings of the Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, 2020, pp. 1–6.