Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE

Review

The Threat of Adversarial Attacks against Machine Learning in Network Security: A Survey

Olakunle Ibitoye[1], Rana Abou-Khamis[1], Mohamed elShehaby[2*], Ashraf Matrawy[1], M. Omair Shafiq[1]

[1] School of Information Technology, Carleton University, Ottawa, ON, Canada
[2] Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada
E-mail: MohamedelShehaby@cmail.carleton.ca

Received: 18 September 2024; **Revised:** 6 December 2024; **Accepted:** 18 December 2024

Abstract: Machine learning models have made many decision support systems to be faster, more accurate and more efficient. However, applications of machine learning in network security face more disproportionate threat of active adversarial attacks compared to other domains. This is because machine learning applications in network security such as malware detection, intrusion detection, and spam filtering are by themselves adversarial in nature. In what could be considered an arm's race between attackers and defenders, adversaries constantly probe machine learning systems with inputs which are explicitly designed to bypass the system and induce a wrong prediction. In this survey, we first provide a taxonomy of machine learning techniques, tasks, and depth. We then introduce a classification of machine learning in network security applications. Next, we examine various adversarial attacks against machine learning in network security and introduce two classification approaches for adversarial attacks in network security. First, we classify adversarial attacks in network security based on a taxonomy of network security applications. Secondly, we categorize adversarial attacks in network security into a problem space vs. feature space dimensional classification model. We then analyze the various defenses against adversarial attacks on machine learning-based network security applications. We conclude by introducing an adversarial risk grid map and evaluate several existing adversarial attacks against machine learning in network security using the risk grid map. We also identify where each attack classification resides within the adversarial risk grid map.

Keywords: machine learning, adversarial samples, network security

1. Introduction

There has been an ever-increasing application of machine learning and deep learning techniques in network security. One key advantage of machine learning is that it makes optimal decisions more feasible.

It, however, introduces a new challenge since security and robustness of these models is usually not a huge consideration for machine learning algorithm designers who are more focused on designing effective and efficient models. This creates room for various forms of attack models against machine learning-based network security applications.

Researchers [1, 2, 3, 4] have shown that the presence of adversarial samples can easily fool machine learning systems. Adversarial samples are specially crafted inputs that cause a machine learning model to classify an input wrongly. Machine learning systems typically take in input data in two distinct phases. The training data which is fed into the learning algorithm

DOI: https://doi.org/10.37256/jeee.4120255738

during the training phase, and the new or test data which is fed into the learned model during the prediction phase. If the attacker can manipulate the input data in either phase, it is possible to induce a wrong prediction from the machine learning model.

In this survey, we provide a brief introduction to machine learning using a three-dimensional classification method. We classify the various machine learning approaches based on the learning tasks, learning techniques and learning depth. We further organize the various applications of machine learning in network security based on a taxonomy of security tasks. Contrary to the survey by Corona et al. [5], our work focuses on adversarial attacks that are strictly machine learning based. Next, we classify the various adversarial attacks based on the applications in network security. We identify five main categories of machine learning applications in network security for our classification method. Finally, we classify adversarial attacks against machine learning based on a taxonomy of network security applications.

Our contribution is threefold. First, we introduce a new method for classifying adversarial attacks in network security based on a taxonomy of network security applications. We also introduce the concept of problem space and feature space dimensional classification of adversarial attacks in network security.

Secondly, we introduce the concept of adversarial risk in computer and network security. We provide a new risk mapping for evaluating the risk of adversarial attacks in network security based on the discriminative or directive autonomy of the machine learning tasks and techniques respectively.

Lastly, we evaluate several adversarial attacks against machine learning in network security applications as proposed by various researchers and classify the attacks based on an adversarial threat attack taxonomy shown in Figure 7.

As we outline in Section 2, prior adversarial attacks surveys [6, 7, 8] mainly covered them in the computer vision domain. Nevertheless, some surveys tackled adversarial attacks on cybersecurity [9, 10, 11, 12], but to the best of our knowledge, there is currently no prior work that has reviewed adversarial attacks in network security based on a classification of network security applications. No prior work has also reviewed the concept of problem space vs. feature space dimensional classification of adversarial attacks in network security. Also, this is the first work to propose an adversarial machine learning risk grid map in the field of network security based on the directive or discriminative autonomy of the machine learning algorithms.

Our proposed taxonomy provides a structured perspective for classifying adversarial attacks based on the characteristics of network security applications. By incorporating the analysis of problem-space and feature-space, the taxonomy enables a deeper understanding of how adversarial attacks operate within these domains. Additionally, introducing the adversarial risk grid map constitutes a novel contribution to the field, offering a systematic approach to assess and quantify the risks posed by adversarial attacks in network security. This mapping enhances the understanding of vulnerabilities across various network security contexts. Furthermore, the comprehensive evaluation of adversarial attacks and their classification using the proposed taxonomy provide valuable insights for researchers and practitioners, illustrating how different attack methods correspond to specific threat scenarios and offering practical knowledge for effectively mitigating these threats.

As illustrated in Figure 1, We structure the remainder of the paper as follows. In Section 2, we survey some related work. In Section 3, we discuss some applications of machine learning in network security. In Section 4, we begin with a brief background about adversarial machine learning followed by a description of our adversarial attack taxonomy. We also review different adversarial attack methods and algorithms. In Section 5, we introduce a classification method for adversarial attacks in network security based on the network security CIA goals of confidentiality, integrity and availability. In Section 6, we discuss and evaluate adversarial risk in machine learning. In Section 7, we review various approaches for defending against adversarial attacks. In Section 8, we provide some discussion and lessons learnt. Finally, in Section 9, we add a conclusion for our survey with guidance for future work.

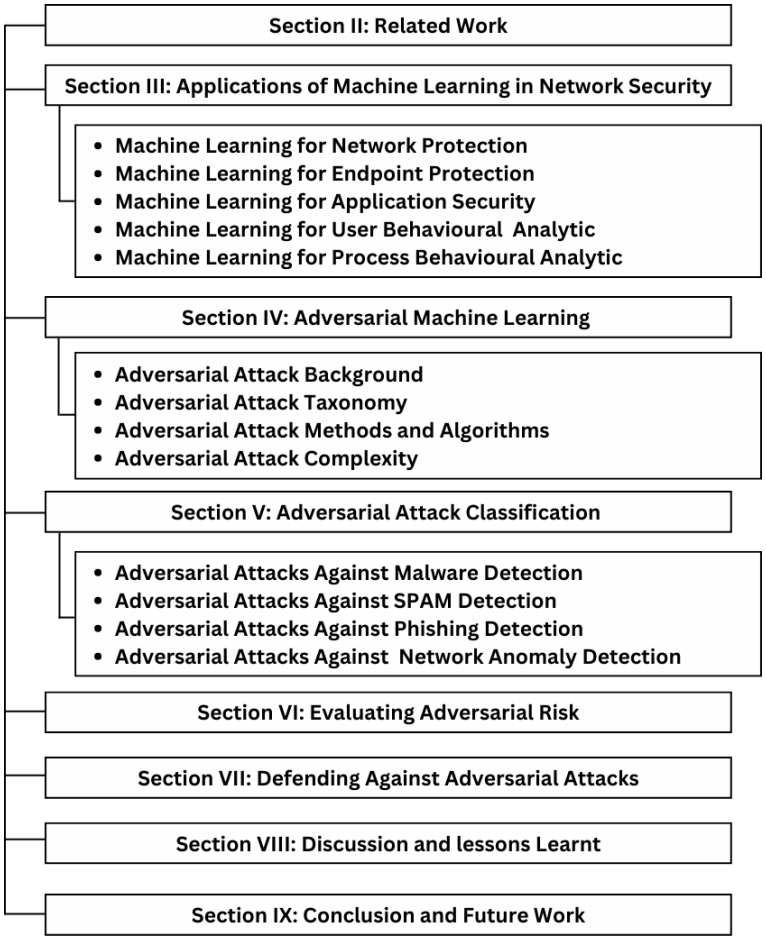

Figure 1. Structure of the paper

2. Related work

Adversarial attacks have been widely studied in the field of computer vision [6, 7, 8] with several attack methods and techniques developed mostly for image recognition tasks. Researchers have discussed the public safety concern of adversarial attacks such as in self-driving cars which could be fooled into mis-classifying a stop sign resulting in a potentially fatal outcome [13]. In network security, the consequences of adversarial attacks are equally significant [14] especially in areas such as intrusion detection [15] and malware detection [16] where there have been rapid progress in the adoption of machine learning for such tasks. Even though adversarial machine learning has recently been widely researched in network security, to the best of our knowledge, there is currently no publication that has surveyed the vast number of growing research work on adversarial machine learning in this field. Some existing survey papers we reviewed include Akhtar et al. [17] which reviewed adversarial attacks against deep learning in computer vision. Qui et al [18] provided a generalized survey on adversarial attacks in artificial intelligence, with a brief discussion on cloud security, malware detection and intrusion detection. Liu et al. [19] reviewed security threats and corresponding defensive techniques of machine learning focusing on the threats in the learning algorithms. Rosenberg et al. [9] provided a general review on adversarial attacks on cyber security domains like; Intrusion detection systems, URL Detection systems, Biometric Systems, CPSs (Cyber-Physical Systems), and Industrial Control Systems. Unlike their work, our review only concentrates on network security and uses different approaches to classify adversarial attacks and defenses. Duddu et al. [10] discussed various research work on adversarial machine learning in cyberwarfare, with some mention of adversarial attacks against malware classifiers. Martins et al. [20] conducted a systematic review of adversarial attacks and found that the practicality

of many of these attacks in the context of network security had not been tested in intrusion scenarios. However, unlike our work, their focus was primarily on intrusion and malware detection scenarios rather than encompassing the broader spectrum of network security domains. Zhang et al. [11] discussed adversarial attacks as a limitation of deep learning in mobile and wireless networking but did not consider deep learning in the context of network security applications. Buczak et al. [21] in their survey on machine learning-based cybersecurity intrusion detection focused on complexity and challenges of machine learning in cybersecurity but did not review adversarial attacks in their study. Biggio and Roli [12] provided an historical timeline of adversarial machine learning in the context of computer vision and cybersecurity but their work did not provide a detailed review in the context of network security. Gardiner et al. [22] in their survey on the security of machine learning in malware detection, focused on reviewing the Call and Control (C & C) detection techniques. They also identified the weaknesses and explained the limitations of secure machine learning algorithms in malware detection systems. Domain specific surveys on adversarial machine learning have also been published including Hao et al. [23] in which various adversarial attacks and defenses in images, graphs and texts were reviewed. In the field of natural language processing, zhang et al. [24] reviewed various publications in which deep adversarial attacks and defenses were proposed. Sun et al. [25] published a survey on adversarial machine learning in graph data. Akhtar et al. [17] computer vision, Duddu et al. [10] cyber warfare.

2.1 *Research gap*

With growing interest in the use of machine learning for network security applications, the significance of adversarial attacks against such machine learning-based application have become more prevalent. With continued increase in the amount of work in this field, there have been recent attempts to review these publications into a survey work. In the field of network security, We identified nine survey papers which attempt to discuss adversarial machine learning from the context of network security. None of these previous survey papers have however explored the vast amount of research work currently ongoing on the topic of adversarial machine learning in network security in a manner that categorizes them based on security applications, problem and feature space dimensional classification and adversarial risk grid map.

Our survey more importantly seeks to distinguish between adversarial attacks in general, and adversarial machine learning in context. We note that an adversary may seek to compromise network security applications in various ways and this may not be related to adversarial machine learning. For example in [5] where adversarial attacks in Intrusion detection systems was reviewed. In our context, adversarial machine learning specifically addresses the optimization problem in which a machine learning based network security solution is being attacked. Many network security solutions are strictly rules based or hard programming dependent and do not implement machine learning techniques. Our survey work does not refer to such adversarial attacks, since they do not capture the real context of adversarial machine learning in principle.

3. Applications of machine learning in network security

Today's network as well as next generation network architectures have become quite complex, and new innovations of network security solutions are required to protect against the growing landscape of cyber threats. Machine learning techniques have been increasingly used to carry out a wide range of tasks in network security [26] incorporating several layers of defenses both within the network and at the edge of the network. In this section, we review and highlight some applications of machine learning in network security by classifying them into five categories as illustrated in Figure 2.

3.1 *Machine learning for network protection*

Intrusion Detection Systems (IDS) are essential solutions for monitoring events dynamically in a computer network or system. Essentially there are two types of IDS (signature based and anomaly based) [27]. Signature based IDS detects attacks based on the repository of attacks signatures with no false alarm [28]. However, zero-day attacks can easily bypass signature-based IDS. Anomaly IDS [28] uses machine learning and can detect a new type of attacks and anomalies. A typical disadvantage of anomaly IDS is the tendency to generate a significant number of false positive alarms.

Figure 2. Machine learning applications in network security

- Hybrid Approach for Alarm Verification Sima et al. [29] designed and built Hybrid Alarm Verification System that requires processing a significant number of real-time alarms, high accuracy in classifying false alarms, perform historical data analysis. The proposed system consists of three components: Machine Learning, Stream processing and Batch processing (Alarm History). Machine learning model trained offline and used for verification service that can immediately classify true or false alarms. They used different machine learning algorithms in the experiments to show the effectiveness of their system where the accuracy achieves more than 90% in a stream of 30K alarms per second [29].
- Learning Intrusion Detection Laskov et al. [30] worked in developing a framework to compare the supervised learning (classification) and unsupervised learning (clustering) techniques for detecting intrusions and malicious. They used different methods in supervised learning to evaluate the work include k-Nearest Neighbor (kNN), decision trees, Support Vector Machines (SVM) and Multi-Layer Perception (MLP). Also, k-means clustering was utilized, with single linkage clustering as unsupervised algorithms. The evaluation was ran under two scenarios to evaluate how much the IDS could generalize its knowledge to new malicious activities. The supervised algorithms showed better classification with the known attacks. The best result among the supervised algorithm was the decision tree algorithm whiched achieved 95% true positive and 1% false positive rate, followed by MLP, SVM and then KNN. If there were new attacks not previously seen in the training data, the accuracy decreases significantly. However, the unsupervised algorithms performed better for unseen attacks and did not show significant difference in accuracy for seen and unseen attacks [30].

3.2 *Machine learning for endpoint protection*

Malware detection is a significant part of endpoint security including workstations, servers, cloud instances, and mobile devices. Malware detection is used to detect and identify malicious activities caused by malware. With the increase in the variety of malware activities, the need for automatic detection and classifier amplifies as well. The signature-based malware detection system is commonly used for existing malware that has a signature but it not suitable for unknown malware or zero-day malware. Machine learning can cope with this increase and discover underlying patterns in large-scale datasets [31].

- Automatic Analysis of Malware Behavior Rieck et al. [32] successfully proposed a framework for analyzing malware behavior automatically using various machine learning techniques. The framework allows clustering similar malware behaviors into classes and assigns new malware to these discovered classes. They designed an incremental approach for the behavior analysis that can process various malware behaviors and reduce the run-time defense against malware development comparing to other analysis methods and provide accurate discovery of novel malware. To implement this automatic framework, they collected a large number of malware samples and monitored their behaviors using a sandbox environment and learn those behaviors using Clustering and Classification algorithms [32].

- Automated Multi-level Malware Detection System In [33], authors proposed Advanced Virtual Machine Monitor-based guest-assisted Automated Multilevel Malware Detection System (AMMDS) that affect both Virtual Machine Introspection (VMI) and Memory Forensic Analysis (MFA) techniques to mitigate in real time symptoms of stealthily hidden processes on guest OS [33]. They use different machine learning techniques such as Logistic Regression, Random Forest, Naive Bayes, Random Tree, Sequential Minimal Optimization (SMO), and J48 to evaluate the AMMDS and the results achieve 100%.

- Classification of Malware System Call Sequences Kolosnjaji et al. [31] focused on the utilization of neural networks by stacking layers according to deep learning to improve the classification of newly retrieved malware samples into a predefined set of malware classes. They constructed Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) layers for modeling System Call Sequences. The sequences used by the CNN layers was based on a set of n-grams. The presence of the n-grams and their relation were counted in a behavioral trace. The RNN on the other hand used sequential information to train the model. A dependence between the system call appearance and the system call sequence was however maintained. If this model was trained properly, it usually provided better accuracy on subsequent data and most often captured more training set information. This deep learning technique for capturing the relation between the n-grams in the system call sequences was deemed to be relatively efficient as it achieved 90% average accuracy, precision and recall for most of the malware families [31].

- A Hybrid Malicious Code Detection Method Li et al. [34] proposed a hybrid malicious code detection scheme based on AutoEncoder and Deep Belief Networks (DBN). They used the AutoEncoder to reduce the dimensionality of data by extracting the main features. Then they used the DBN that composed multilayer Restricted Boltzmann Machines (RBM) and a layer of BP neural network to detect malicious code. The BP neural network has an input vector from the last layer of RBM based on unsupervised learning and then use supervised learning in the BP neural network. They achieved the Optimal hybrid model. The experiment results that are verified by KDDCUP'99 dataset show higher accuracy compared to a single DBN and reduce the time complexity [34].

3.3 *Machine learning for application security*

Various machine learning tasks used for application security including malicious web attack detection, phishing detection and spam detection.

- Detection of Phishing Attacks Basnet et al. [35] studied and compared the effectiveness of using different machine learning algorithms for classification of phishing emails using many novel input features that helps in detecting phishing attacks. The training dataset is labeled with phishing or legitimate email. They used unsupervised learning to extract features without prior training directly and provides fast and reliable knowledge from the dataset. They used 4000 emails in total, A total of 2000 emails used for testing. They used Support Vector Machines (SVM), Leave One Model Out, Biased SVM, Neural Networks, Self Organizing Maps (SOMs) and K-Means on the dataset. Consistently, Support Vector Machine achieved the best results. The Biased Support Vector Machine (BSVM) and NN have an accuracy of 97.99% [35].

- Adaptively Detecting Malicious Queries in Web Attacks Don et al. [36] proposed a new system called AMODS and learning strategy called SVM HYBRID for detecting web attacks. AMODS is an adaptive system that aims to periodically update the detection model to detect the latest web attacks. The SVM HYBRID is an adaptive learning strategy which was implemented primarily for reducing manual work. The detection model was trained using dataset which was obtained from an academic institute's web server logs. The proposed detection model outperformed existing web attack detection methods with an FP rate of 0.09% and 94.79% F-value. The SVM Hybrid system obtained a total number of malicious queries equal to 2.78 times by the popular SVM method. Also, the Web Application Firewall (WAF) can use malicious queries to update the signature library. The significant queries were used for updating the detection model which consisted of a meta-classifier as well as other three base classifiers [36].

- URLNet -Learning a URL Representation with Deep Learning for Malicious URL Detection Le et al. [37] proposed an end-to-end deep learning framework which did not require sophisticated feature. URLNet was introduced to address several limitations which was found with the other model approaches. This framework learns from the URL directly how to perform a nonlinear URL embedding which then enabled it to successfully detect various Malicious URLs. Convolutional Neural Networks (CNN) were applied to both the characters and words of each URL to discover the URL embedding method. They also proposed advanced word-embedding techniques to deal with uncommon words, which was a limitation being experienced by other malicious URL detection systems. The framework then learns from unknown works at testing phase [37].

3.4 *Machine learning for user behavior analytic*

User behavior analytics is a cybersecurity process which involves analyzing patterns in human behaviors and detecting anomalies that give an indication of fraudulent activities or insider threats. Machine learning algorithms are used to detect such anomalies in user actions such as unusual login tries and to infer useful knowledge from those patterns.

- Authentication with Keystroke Dynamics Revett et al. [38] proposed a system using Probabilistic Neural Network (PNN) for keystroke dynamics that captures the typing style of a user. A system comprising of 50 user login credential keystrokes was evaluated. The authors [38] used eight attributes to monitor the enrollment and authentication attempts. An accuracy of 90% was obtained in classifying legitimate users from imposters. A comparison of the training time between the PNN system and a Multi-Layer Perception Neural Network (MLPNN) showed that the PNN was four times faster.

- Text-based CAPTCHA Strengths and Weaknesses Bursztein et al. [39] in a study showed that several well known websites still implemented technologies that have been proven to be vulnerable to cyber attacks. In the study, an automated Decaptcha tool was tested on numerous websites including well known names such as eBay, Google and Wikipedia. It was observed that 13 out of 15 widely used web technologies were vulnerable to their automated attack. They had a significant success rate for most of the websites. Only Google and Recaptacha were able to resist to the automated attack. Their study revealed the need for more robust CAPTCHA designs in most of the widely used schemes. Authors recommended that the schemes should not rely on segmentation alone because it did not provide sufficient defense against automated attacks.

- Social Network Spam Detection K. Lee et al. [40] proposed social network spam detection that gathers legitimate and spam profiles and feeds them to Support Vector Machine (SVM) model. The authors selected two social networks: Twitter and MySpace to evaluate the proposed machine learning system. They collected data over months and feed them to the SVM classifier. The dataset contains 388 legitimate profiles and 627 spam profiles collected from MySpace, and 104 legitimate profiles and 168 profiles between promoters and spammers collected from Twitter. The system achieved a low false positive rate and high precision up to 70% for MySpace and 82% for Twitter.

3.5 *Machine learning for process behavior analytic*

Machine learning applications usually necessitate the need to learn and have some domain knowledge about business process behaviors in order to detect anomalous behaviors. Machine learning could be used for determining fraudulent transactions within banking systems. Also it has been successfully used for identifying outliers, classifying types of fraud and for clustering various business processes.

- Anomaly detection in Industrial Control Systems Kravch et al. [41] performed a successful study on SecureWater Treatment Testeb (SWat) using Deep Convolutional Neural Networks CNN to detect most of attacks on Industrial Control System (ICS) with a low false positive. The anomaly detection method was based on the statistical deviation measurement of the predicted value. They performed the study using 36 different attacks from SWat. The authors in [41] proofed that using 1D convolutional networks in anomaly detection in ICS outperformed the recurrent networks.

- Detecting Credit Card Fraud Traditionally, the Fraud Detection System uses old transactions data to predict a new transaction. Fraud Detection System (FDS) should encounter various potential challenges and difficulties to achieve high accuracy and performance [42]. The traditional detection method does not solve all problems and challenges including imbalanced data where there is a small chance of transactions are fraudulent. Wrong classification and overlapping data and Fraud detection cost are other major challenges [42]. Chen et al. [43] proposed an approach to solving the listed challenges and problems for Credit Card fraud. They introduced a system to prevent fraud from the initial use of credit cards by collecting user data from online questionnaire based on consumer behavior surveys. They used various classifiers models: decision tree (C5.0, CandRT, CHAID) and SVM (linear and radial basis, Kernels of polynomial, sigmoid). They use three datasets to develop questionnaire-responded transaction (QRT) model to predict new transaction.

- Deep Learning Techniques for Side-Channel Analysis Prouff et al. [44] defined Side-Channel Analysis as a type of attack that attempts to leak information from a system by exploiting some parameters from the physical environment [44]. This attack was utilizing the running-time of some cryptographic computation, especially in the block ciphers. The capability of a system to resist side-channel attacks (SCA) requires an evaluation strategy that focuses on deducing the relationship between the device behavior and the sensitivity of the information that is common in classical cryptography. The authors in [44] focused on proposing an extensive study of using deep learning algorithms in the Side-Channel Analysis. Also, they focused on the hyper-parameters selection to help in designing new deep learning classifier and models. They confirmed that the Convolutional Neural Networks (CNN) models are better in detecting SCA. Their proposal system outperformed the other tested models on highly desynchronized traces and had the best performance as well on small desynchronized trace [44].

4. Adversarial machine learning

4.1 *Adversarial attack background*

Adversarial attacks have been studied for more than a decade now [12]. However, the first notable discovery in adversarial attacks for computer vision was by Szegedy et al. [45] who reported that a small perturbation in the form of a carefully crafted input could confuse a deep neural network to misclassify an image object. Other researchers have demonstrated the use of adversarial attacks beyond image classification [46, 47, 48, 49].

In adversarial machine learning, an adversary seeks to confuse a machine learning model into making a wrong decision. The adversary achieves this by modifying the input data that is fed to the machine learning model either during the training phase (poisoning attack) [50] or during the inference phase (evasion attack) [51].

The reason behind adversarial examples has been linked to the fact that most machine learning models remain overtly attached to the superficial statistics of the input data [52, 53]. This attachment to the input data makes the machine learning highly sensitive to distribution shift, resulting in a disparity between semantic changes and a decision change [1].

We consider the security model for use of machine learning in network security as a combination of four components namely the attack surface, threat model, adversarial framework and adversarial risk. An alternative adversarial model was proposed in [54] which modeled the adversary using a threefold approach based on knowledge, goals and capability. The attack surface identifies the various attack vectors along a typical machine learning data processing pipeline in network security related applications. The threat model provides a system abstraction for profiling the adversary's capabilities and the potential threats that are associated. The adversarial framework details our approach for classifying the various attacks and defenses within each network security domain and lastly the adversarial risk provides an evaluation of the likelihood and severity of adversarial attacks within a network security system.

A major component of an adversarial attack is the adversarial sample. As illustrated in Figure 3, an adversarial sample consists of an input to a machine learning model which has been perturbed. For a particular dataset with features x and label y, a corresponding adversarial sample is a specific data point x' which causes a classifier c to predict a different label on x' other than y, but x' is almost indistinguishable from x. The adversarial samples are created using one of many

optimization methods known as adversarial attack methods. Crafting adversarial samples involves solving an optimization problem to determine the minimum perturbation which maximizes the loss for the neural network.

Figure 3. Adversarial machine learning

Considering an input x, and a classifier f, the optimization goal for the adversary is to compute such perturbation with a small norm, measured w.r.t some distance metric, that would modify the output of the classifier such that

$$f(x+\delta) \neq f(x)$$

where δ is the perturbation. If δ is applied to all of the input data (all of the image's pixels, for example), it is considered a **dense adversarial attack**. However, if just partial positions are perturbed, it is called a **sparse adversarial attack** [55].

Adversarial machine learning in network security is typically an arms race between two agents. The first agent is an adversary whose objective is to intrude a network with a malicious payload. The other agent is one whose role is to protect the network from the consequences of the malicious payload.

We start with a view of the different type of data that traverses a network during any given time.

4.2 *Adversarial attack taxonomy*

We examine the Adversarial Attack Taxonomy in Table 1 to consider the goals and capabilities of any adversary for a machine learning system. We base our threat framework from the original model in [8, 54] and adapt it within the context of adversarial attacks in network security domain. Within this context, adversarial attack threats in network security may be considered based on the attacker's knowledge, attack space, attacker's strategy, attacker's goal and attack target. As mentioned in Section 1, to the best of our knowledge, this is the first review to add the idea of the space dimension in the classification of adversarial attacks in network security.

Table 1. Adversarial attack taxonomy

	Types	References
Knowledge	Black box	[7, 56]
	White box	[57, 1]
	Gray box	[58]
Space	Feature space	[15]
	Problem space	[59]
Strategy	Evasion	[58, 60]
	Poisoning	[61, 62, 63]
	Oracle	[64, 61]
Goal	Availability	[65, 66]
	Integrity	[65, 67]
	Confidentiality	[65, 68]
Target	Physical Domain	[69, 70]
	ML Model	[1]

4.2.1 *Knowledge*

The knowledge component of the adversarial threat model describes the extent to which the adversary knows about the machine system as a whole. This could be classified as **White-box**, **Gray-box** or **Black-box** attacks.

- In white-box attacks, it is assumed that the attacker has complete knowledge of the training data, the learning algorithm, the learned model as well as the parameters which were used while training model. A white-box attack represents an adversary who has the exact information that is held by the owner or creator of the machine learning system which is being under attack. In the majority of real world adversarial attack settings, this is usually not feasible.

- A Gray-box attacks assumes a more realistic approach, and considers that there could be varying degrees information accessible to the adversary [58]. For example, an adversary may have partial information about the model queries, or limited access to the training data. For a gray-box attack, the adversary does not have the exact knowledge which the creator of the model possesses, but has sufficient information to attack the machine learning system to cause the machine learning system to fail.

- A black-box attack assumes that the adversary is totally unaware of the machine learning system. in this type of attack, the adversary has no knowledge about either the learning algorithm or the learned model. It may be argued that a truly black-box attack is impossible. this is because it is assumed that the adversary must at least have some specific information, for example the location of the model before it can attack the model. The severity of blackbox attacks poses a greater threat in practice. The model for real-world systems may be more restrictive than a theoretical black-box model where the adversary can understand the full output of the neural network on inputs that have been chosen arbitrarily. In [71], an analysis of three threat models were proposed. These models, defined as, the query-limited setting, the partial information setting, and the label-only setting, provide a more accurate characterization of real-world classifiers. As such, a representation of black box adversarial attacks was proposed, such that, it would be possible to fool classifiers under these more restrictive threat models, whereas, it might have been impractical or ineffective.

4.2.2 *Space*

In the field of adversarial machine learning, the input space can be defined as a dimensional representation of all the possible configurations of the objects in determination context. We categorize this as **Feature Space** and **Problem Space**.

- Feature space modeling of an adversarial sample is a method in which an optimization algorithm is used to find the ideal value out of a finite number of arbitrary changes made to the features. In a feature space adversarial attack, the attacker's objective is to remain benign without generating a new instance. Conversely, a feature space is defined as the n dimensional space in which all variables in the input dataset are represented. We take as an example an intrusion detection dataset with 70 variables, this represents a 70-dimensional feature space. A feature space adversarial attack in the context above will seek to alter the feature space by making changes within the 70-dimensional feature space. A feature space attack modifies the features in the instance directly. Using an example of malware adversarial attacks, a feature space adversarial malware attack will only modify the feature vectors but no new malware is created.

- The problem space refers to an input space in which the objects e.g., image, file, etc. resides. A problem space adversarial malware attack will modify the actual instance from the source to produce a new instance of the malware. Typically, a problem space adversarial attack tends to generate new objects in domains such as malware detection whereby there is no clear inverse mapping to the feature space [59]. A typical difference between a problem space adversarial attack, and a feature space adversarial attack is that a feature space attack does not generate a new sample but only creates a new feature vector. A problem space adversarial attack modifies the actual instance itself to create an entirely new object.

4.2.3 *Strategy*

Attacker's strategy implies the phases of operation in which the adversary launches the attack. Three main strategies which an adversary may use in adversarial attacks are **Evasion**, **Poisoning** and **Oracle**.

- Evasion attacks, also known as exploratory attack or attack at decision time, during the testing or inference phase. The attacker aims to confuse the decision of the machine learning model after it has been learned as shown in Figure 4. Evasion attacks typically involve an arithmetic computation of an optimization problem. The objective of the optimization problem is to compute a tiny perturbation *sigma* which would cause an increase in the loss function. The change in loss function would then be significant enough to result in a wrong prediction by the machine learning model. Evasion attacks are classified as gradient-based attacks or gradient-free attacks.

  Gradient-based attacks are further classified based on the frequency with which the adversarial samples are updated or optimized. These are **iterative** or **One-shot** attacks. Iterative attacks provide tighter control of the perturbation in order to generate more convincing adversarial samples [61]. This however results in higher computational costs. Alternative to iterative attacks are one-shot attacks which adopt a single-step approach without iterations. One-shot or one-time attacks are attacks in which the adversarial samples are optimized just once. Iterative attacks, however, involve updating the adversarial samples multiple times. By updating the adversarial samples multiple times, the samples are better optimized and perform better compared to one-shot attacks. However, iterative attacks cost more computational time to generate.

  Adversarial attacks against certain machine learning techniques which are computationally intensive such as reinforcement learning usually demand one-shot attacks as the only feasible approach [60].

  Gradient-free attacks [58], unlike gradient-based attacks do not require knowledge of the model. Gradient-free attacks can generate potent attacks against a machine learning model with knowledge of only the confidence values of the model.

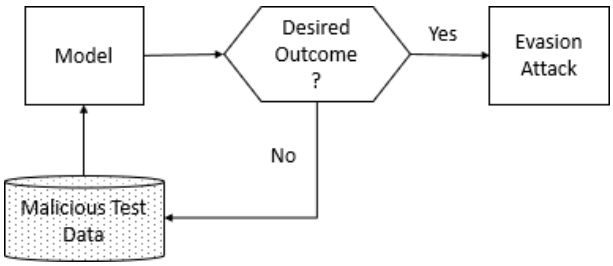

Figure 4. Evasion attack

- Poisoning attacks, also known as causative attack, involves adversarial corruption of the training data or model logic during the training phase to induce a wrong prediction from the machine learning mode as shown in Figure 5. Poisoning attacks may be carried out by data injection, data manipulation or logic corruption [61]. Data injection occurs when the adversary inserts adversarial inputs to alter the data distribution while preserving the original input features and data labels. Data manipulation refers to a situation in which either the input features or data labels of the original training data are modified by the adversary. Logic corruption is an attempt by the adversary to model structure.

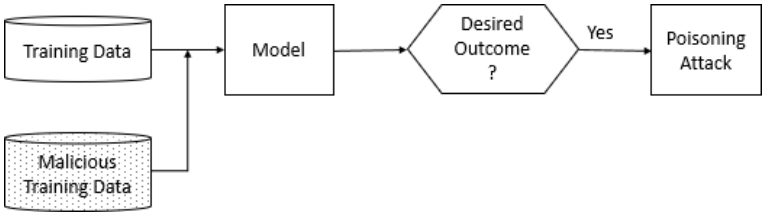

Figure 5. Poisoning attack

- Oracle attacks occur when an adversary leverages the access to the Application Programming Interface of a model, to create a substitute model with malicious intent. The substitute model typically preserves a significant part of the functionality of the original model [64]. As a result, the substitute model can then be used for other types of attacks such as evasion attacks [61]. Oracle attacks can be further subdivided into **Extraction**, **Inversion** and **Inference** attacks. The objective of an extraction attack is to deduce model architectural details such as parameters and weights from an observation of the model's output predictions and class probabilities [72]. Inversion attacks occurs when adversary attempts to reconstruct the training data. An inference attacks allows the adversary to identify specific data points with the distribution of the training dataset [73].

4.2.4 *Goal*

Traditionally in the field of computer vision, adversarial attacks are regarded in terms of targeted or reliability attacks [17]. In targeted attacks, the attacker has a specific goal with regard to the model decision. Most commonly, the attacker would aim to induce a definite prediction from the machine learning model. On the other hand, a reliability attack occurs when the attacker only seeks to maximize the prediction error of the machine learning model without necessarily inducing a specific outcome. Yevgeny et al. [14] have noted that the distinction between reliability and targeted attacks becomes blurred in attacks on binary classification tasks such as malware binary classification. As such, these conventional paradigms of attacker goal classification is not optimal for consideration in network security. We choose to adopt the CIA triad in this context and find that it is more suitable for adversarial classification of the adversary goals in network security domain.

- Confidentiality attack refers to the goal of the attacker to intercept communication between two parties A and B, to gain access to private information being exchanged. This happens within the context of adversarial machine learning, whereby machine learning techniques are being used to carry out network security tasks.
- Integrity attack seeks to cause a misclassification, different from the actual output class which the machine learning model was trained to predict. Integrity attack could result in a targeted misclassification or a reliability attack. A targeted misclassification attempts to make the machine learning model to produce a specific wrong prediction. A reliability attack results in either a confidence reduction or a misclassification to any arbitrary class apart from the correct class.
- Availability Attack results in a denial of service situation for the machine learning model. as a result, the machine earning model becomes either totally unavailable to the user, or the quality is significantly degraded to the extent that the machine learning system becomes unusable to the end users.

4.2.5 *Target*

In our surveyed work, adversarial attacks are targeted against a specific machine learning technique. Several successful attempts have been made towards the transferability of adversarial attacks [74, 75]. However, attacks that have been targeted towards a specific machine learning technique for example unsupervised learning, have not been successfully transfered towards a another technique for example supervised learning. Regarding the physical domain, it includes input sensors, cameras and output actions.

4.3 *Adversarial attack methods and algorithms*

We recall that adversarial attacks could be deployed either during decision time (evasion attacks) or during training time (poisoning attacks). In each case, the training algorithm (for poisoning attacks) or the learned model (for evasion attacks) is being manipulated with some form of carefully crafted input known as the adversarial samples. A common trend among the attack methods below reveals that the robustness of a machine learning model to a large extent depends on the ability of an attacker to find an adversarial sample that is as close as possible to the original input. In this section, we evaluate the primary methods for generating adversarial samples. It should be noted that recent research has shown the limitations of some earlier methods that are still listed here for reference even though more effective methods have been introduced.

In the previous Section 4.2, we described our threat model for adversarial attacks in network security. In this section, we introduce a classification method for the various adversarial attack algorithms. As seen in Figure 6 our classification method is based on the adversary strategy described in Section 4.2.3.

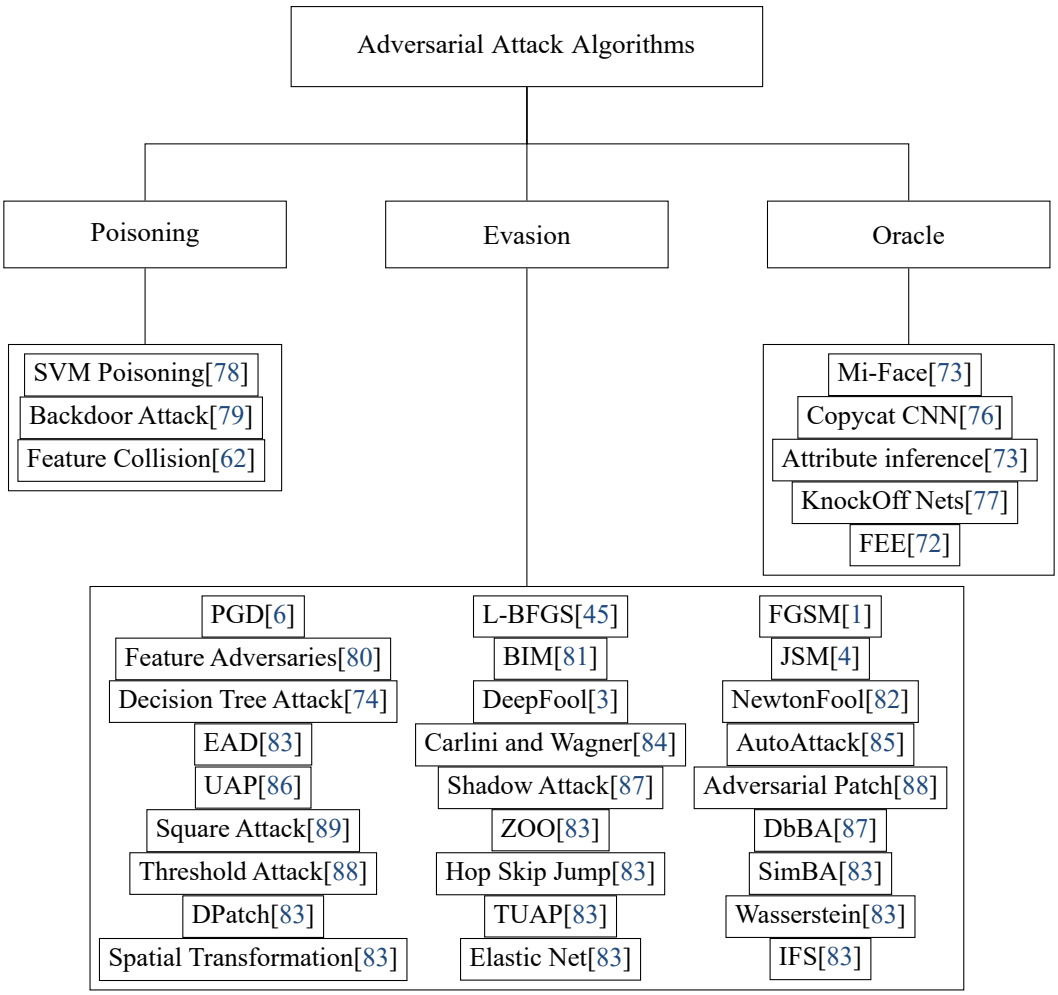

Figure 6. Adversarial attack algorithms

4.3.1 *Evasion attacks*

Evasion attacks attempt to mislead the machine learning system during the testing or inference phase. Below we highlight adversarial attack methods that fall within this category of evasion attacks. The attacks are further divided into **Gradient-based** and **Gradient-free** attacks.

- Gradient-based attacks: Szegedy et al. [45] studied how adversarial samples could be generated against neural networks for image classification. The L-BFGS (Limited Broyden-Fletcher- Goldfarb-Shanno) method was then introduced, which used an expensive linear search method to find the optimal values of the adversarial samples. In a different approach proposed by Goodfellow et al. [1] called the Fast Gradient Sign Method (FGSM), adversarial samples are created by finding the maximal direction of positive change in the loss. This is a faster method than the L-BFGS method since only a one-step gradient update is performed along the direction of the sign gradient at each level. A major limitation of the Fast Gradient Sign Method and similar attack methods is that they work based on the assumption that the adversarial samples can be fed directly into the machine learning model. This is far from being practical since most attackers would seek to access the machine learning models through devices such as sensors [90]. The Basic Iterative Method (BIM) proposed in [81] overcomes this limitation by running the gradient update in multiple iterations.

  The Jacobian-based Saliency Map Attack (JSMA) was introduced by Papernot et al. [4]. For the attack, the Jacobian matrix of a given sample is computed to find the input features of that sample which most significantly impacts the output. Subsequently, a small perturbation is created based on that input feature for generating the adversarial attack. DeepFool was proposed by Moosavi et al. [3] as a method for creating adversarial samples by finding out the closest distance between original input and the decision boundary for adversarial samples. They were able to determine that by using a related classifier, the closest distance which would correspond to the minimal perturbation for creating an adversarial sample will be the distance to the hyperplane of the related classifier.

  Jang et al. [82] presented the NewtonFool attack, an algorithm that is based on gradient-descent to find adversarial samples. This attack is similar to Deepfool [3] but more effective in producing good adversarial samples and reduces the confidence probability of the correct class. They exploit the softmax layer and control the step size and how small the perturbation could be. Carlini et al. [84] developed Carlini and Wagner Attack, a targeted attack specifically for existing adversarial defense methods. It was discovered that defenses such as defensive distillation [91] were ineffective towards the Carlini and Wagner attack. Madry et al. [6] proposed the Projected Gradient Descent (PGD) adversarial attacks that is more robust than FGSM. This form of attack utilizes a multi-step approach with a negative loss function. It overcomes the network overfit problem, and shortly comes of FGSM adversarial samples. It is more robust than FGSM, which utilizes the first-order network information, and it works well in large-scale constraints. In ℓ_∞-ball, PGD iterate to explore the maximum loss.

  Croce et al. [85] proposed Auto Attack, an attack that overcomes and remedy the weaknesses of Projected Gradient Descent (PGD) [6] that lead to model robustness false outcomes. First PGD attack use fixed step size with cross-entropy as a loss function that causes the failure as identity by [92]. In [85], they use a new gradient-based scheme without step size selection with different loss function. With these two changes, two versions of PGD produced with free parameters in the number of iteration. They also integrate the new PGD versions with FAB-attack [93] and Square attack [89] to produce a parameter-free attack called AutoAttack. The authors also integrated two Auto Attack and were tested on a large scale on 40 classifiers.

  Sabour et al. [80] proposed a new adversarial image attack that not only focus on the class label but in the internal representations. The attack, known as Feature Adversaries enables the possibility to deceive a trained DNN to mystify any source image with other target image by finding a small perturbation from the source image that create similar internal representation to the target image and not related to the source image. The authors however take into consideration that such adversaries are not outliers. Universal Perturbation [86] was proposed by Moosavi et al. as an algorithm to calculate a universal small image perturbation to misclassify a state-of-the-art deep neural network classifier. The main focus of this algorithm was to find the perturbation vector that deceives classifier on all

data point samples. This fix perturbation is existed to lead changes in image label gradually to build the universal perturbation.

- Gradient-free Attacks: Decision Tree Attack was proposed by Papernot et al. [74] this type of black-box attacks use transferability of adversarial samples between and within different classifiers, including Deep neural network. Logistic regression, decision trees, support vector machines (SVM), ensembles, and nearest neighbors. They demonstrated that black box attacks are feasible to a machine learning algorithm that not using deep neural networks and adversarial samples works well between and across models using the same and different machine learning techniques. Chen et al. [83] proposed an adversarial attack algorithm to attack DNN based on elastic-net regularization in feature L_1 and L_2 called elastic-net attacks to DNNs (EAD). EAD considers state-of-the-are L_2 and L_∞ Authors demonstrated that EAD could break undefended and defensively distilled DNNs. They also improve the transferability of attacks and adversarial training. Shadow Attack was proposed by Ghiasi et al. [87] which is a new method for attacking systems that rely on certificates and fool certified robust networks to assign the wrong label to an image and produce a spoofed secure robustness certificate for the adversarial example. Adversarial Patch, proposed by Brown et al. [88] present universal, robust, and targeted adversarial patches for the real world that do not require any knowledge about what image they are attacking. Those adversarial samples can be used to attack any classifier, and they work with many transformations that exist defense methods may not be robust to such a massive transformation. The adversarial patch leads the classifier to switch class labels to any target class. Chen et al. [94] develop HopSkipJumpAttack based on a decision-based attack that is a type pf black-box attack. This algorithm generates iterative targeted and untargeted adversarial samples with minimum distance. This attack demonstrates superior efficiency over various state-of-the-art decision-based attacks. The iteration in the algorithm is based on gradient direction, step size, and boundary search.

4.3.2 *Poisoning attacks*

A poisoning attack also known as causative attack, uses direct or indirect means to alter the data or the model. Poisoning attacks occurs either by injecting false data, manipulating the original data, or corrupting the model logic.

- Data Injection: Biggio et al. [78] proposed a gradient ascent based attack based on SVM that attacks the input data that lead to maximize the non-convex surface error and increase classifier classification at the test time. Gu et al. [79] proposed BadNets, which perform adversarial attacks by discovering the backdoored neural network or BadNet. The attack is based on a full or partial outsourced training process where attacker provides the user with a trained model with a backdoor that causes a targeted misclassification and degrade in the accuracy in some cases called backdoor trigger. For example, in autonomous driving, an attacker provides the user with a street sign detector that is backdoored, which classify stop sign well in most cases except when the stop signs have a particular sticker in classifying it as speed limit signs. This type of attack occurs under two scenarios user outsource trained model or download a pre-trained model.

- Data Manipulation: Feature Collision Attack proposed by Shafahi et al. [62] presents a watermarking poisoning attack based on optimization-based to craft a clean label attack to target the behavior of a neural network classifier on a specific instance. This attack uses enhanced preservation techniques to make it difficult to be detected.

4.3.3 *Oracle attacks*

In an oracle type adversarial attack, an adversary who has been given a oracle prediction access to a model, steals a copy of a remotely deployed machine learning model. This enables the adversary to duplicate the functionality of the model, i.e., “steal the model” [64]. This attack has become increasingly common due to the increase in Machine Learning as a Service “MLaaS” offerings where several companies that offer cloud-based Machine Learning services e.g., Google, Amazon, and BigML, provide easy-to-use web APIs to manage client interaction.

- Inversion Attacks: Fredrikson et al. [73] exposed the privacy issues with providing access to machine learning API. Their study demonstrated how an adversary could utilize the confidence information of a model to result in model inversion attacks. The attack, which is implemented as a function called MI-Face attack, enables an adversary to extract pictures of subjects from a trained machine learning model.
- Inference Attacks: Fredrikson et al. [73] proposed the attribute inference attack which could be launched either as a white-box or black-box attack.
- Extraction Attacks: Correia-Silva et al. [76] demonstrated how an adversary could create a substitute model from a black-box convolutional neural network (CNN) model by querying the black-box model with random non-labeled data. A more intriguing aspect of this oracle type of extraction attack is the fact that dataset used to persuade the model was not related to original problem domain. Orekondy et al. [77] proposed Knockoff Nets which are capable of stealing the functionality of a fully trained model using a two-step approach. The adversary first obtains predictions from the model by querying a set of input data, then the data-prediction pairs are used to create a substitute model known as a "knock-off" model. Their approach uses a reinforcement learning approach with demonstrated query efficiency and performance gains, compared to other oracle type attacks. Jagielski et al. [72] proposed the Functionally Equivalent Extraction (FEE) attacks which explore accuracy and fidelity objectives within the space of model extraction by improving the query efficiency of learning attacks. Their method is demonstrated to be practical for high parameter models in the range of millions. In their attack method, an adversarial model is produced whose architecture and weights are identical to the oracle.

4.4 *Adversarial attack complexity*

The complexity of adversarial attacks depends on multiple factors, including the attack's implementation, type, the attacker's knowledge and objectives, and the characteristics of the targeted model, application, or domain. These factors result in substantial differences in time and space complexities across various adversarial attack methods.

For time complexity, evasion attacks generally measure the time required to generate adversarial perturbations, which can vary significantly based on the attack approach. For example, gradient-based attacks, such as FGSM [1], PGD [6], and BIM [81], typically have a time complexity that scales with the number of model parameters and the steps involved in gradient calculation. Iterative methods, like PGD and BIM, add computational cost with each iteration, while non-iterative methods, like FGSM, usually demonstrate lower time complexities. Query-based black-box attacks, which operate without gradient access, can be particularly time-intensive as they rely on repeated input modifications and model queries to infer adversarial directions, making time complexity highly dependent on the number of allowable queries. Poisoning attacks, in turn, can be computationally intensive due to stages like data selection, modification, and model retraining, with time complexity influenced by factors such as dataset size and the number of poisoning iterations.

Similar variations are observed in space complexity, where different types of adversarial attacks exhibit significant differences. For instance, gradient-based evasion attacks often require the storage of gradient information for each model layer, while black-box attacks sometimes necessitate creating surrogate models, which require additional storage. In summary, the complexity of adversarial attacks depends on numerous factors; however, adversarial attacks are generally complex tasks. That said, time and space complexities are not the sole determinants of the practicality of adversarial attacks in network security. Other contributing factors will be discussed in Section 8.6.

5. Adversarial attack classification

Multiple studies [95, 96] have sought to differentiate the different domains of network security into multiple fragmented domains. A common approach for example make attempts at differentiating malware and spam detection from intrusion detection [9]. We find that this attempt of fine grained classification results in redundancy, since the task of malware or phishing detection in a network could be considered an intrusion detection task. As such, in this survey, we consider cyber attacks against a network as an attempt by an adversary to intrude the network with a malicious payload. We identify

malicious payload in a network to consist of three broad types: malicious files (malware), malicious text (spam) and malicious url links (phishing). We note that attackers may use a combination of all three payloads in most cyber attacks. For example, a spam email may also contain a link to a malicious url or contain a malicious file attachment. This payload approach becomes even more crucial in our study on adversarial attacks within the network security domain. We realise from our study that this distinction plays an important role in providing an accurate classification of adversarial attacks within the network security domain, as compared to other domains such as computer vision.

In this section, we introduce a classification method for adversarial attacks in network security based on network security task. Our classification approach considers the data object which is being manipulated by the adversary. The feature scope of the adversarial attack corresponds to the data object as shown in Figure 7.

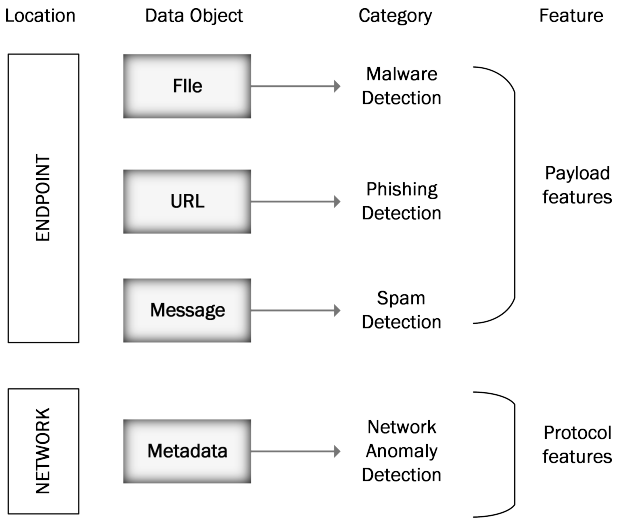

Figure 7. Adversarial attack classification

For the scope of this study, we consider adversarial attacks based on the actual payload which is being attacked in context. When a message is being transmitted from a sender to a receiver, the payload represents the portion of the transmitted data that is actually the intended message. For example, when an email is sent, the payload consist of the message body, attachments, and URL links. Headers and metadata which help to facilitate the delivery of the payload are not considered as part of the payload, within the context of our study. Hence, the protocol overhead is not considered as part of the actual data.

Our approach for classifying adversarial attacks in network security is based off this approach, as shown in Figure 7. This is known as feature scope based classification, which refers to what features are being manipulated or perturbed by the adversary in other to generate an adversarial sample. Adversarial attacks against malware detection, phishing detection and spam detection applications try to perturb the payload features such as a binary file, a URL, or an email message. These attacks are categorized as adversarial attacks against endpoint protection systems. Conversely, we also have adversarial attacks against network anomaly detection applications and these type of attacks will seek to perturb protocol features such as the network metadata or protocol headers. We categorize these attacks as adversarial attacks against network protection systems.

Network security domain that utilize machine learning techniques fall into four broad categories namely malware detection, phishing detection, spam detection and network anomaly detection. We illustrate this categorization in Figure 7. The first three categories of network security tasks are considered as endpoint based protection. Machine learning applications within this endpoint based protection category are typically initiated with payload features. Network protection primarily constitutes network anomaly detection and machine learning applications within this category are typically initiated with protocol features. Our study only considers active attacks against a network, and passive attacks such as

eavesdropping are not within the scope of this study. Adversarial attacks hence seek to generate adversarial samples using specific data objects.

In contrast to adversarial attacks in the field of image processing or computer vision, network security's adversarial learning is more challenging. This occurs because even very slight modifications to URLs, spam, packets, or malware bytes of the binary files can significantly alter the functionality of the data. In computer vision, the addition of tiny perturbations to an image sample does not alter the human perception of the image and same as in speech processing. Text processing and network security filtering techniques are similar in this regard since a very slight change in the input such as a word or a byte will alter the meaning of the text or the data functionality. Hence, approaches for generating adversarial samples in the domain of machine learning-based network security filtering systems need to occur in such a way that the malicious functionality is not distorted. Several approaches for achieving these adversarial attacks have been researched and are discussed in the sections below.

5.1 *Adversarial attacks against malware detection*

A major component of endpoint protection in network security is malware detection. Yet, malware detection remains a challenging problem in network security. Between 2009 and 2019, the number of new malware digital signatures has increased by over 2000 percent [97]. Therefore, traditional malware detection systems that rely solely on digital signatures have become less effective. Significant effort has been made in the use of machine learning to protect against malware attacks. Several researches have shown the vulnerability of these machine learning models to adversarial attacks. The most common approach is the addition of selected sequence of bytes to the binary file. Several approaches have been considered for synthesizing this sequence of bytes as discussed below.

Malware detection may be based on static analysis, in which the malware is detected without executing the code. Alternatively, dynamic analysis for malware detection typically executes a suspicious malware sample in a sandbox in an attempt to discover dynamic behavioural patterns such as API call sequences.

5.1.1 *Iagodroid*

One of the earliest attacks against machine learning based malware detection systems was the Iagodroid attack [98]. Iagodroid uses a method to induce mislabelling of malware families during the triaging process of malware samples. Their evasion rate reached 97 percent.

5.1.2 *Stingray*

Suciu et al [99] proposed an adversarial attack against malware using the 'FAIL' model. Their study focuses on constraints of obscurity and transferability in order to realize a targeted poisoning attack. StingRay succeeded in half of the test cases.

5.1.3 *Texture perturbation attacks*

Researchers have deployed visualization techniques similar to computer vision and adapted it for malware classification [100]. This involves conversion of malware binary code into image data. The Adversarial Texture Malware Perturbation Attack (ATMPA) achieved a 100 percent effectiveness in defeating visualization based machine learning malware detection system and also resulted in 88.7 percent transfer-ability rate [16]. The attack model for ATMPA works by allowing the attacker to distort the malware image data during the visualization process.

5.1.4 *Android malware attack in problem space*

[59] et al. formalized an approach for problem space adversarial evasion attacks against machine learning based android malware detection systems. Their study identified four main contraints which are characteristic of any problem space attack. Their study adopted a technique which automates the generation thousands of realistic and inconspicuous

adversarial malware samples, further buttressing the notion of adversarial malware as a service as a real threat in network security. Their attack led to a misclassification rate of 100.0 percent on the successfully generated samples.

5.1.5 *EvadeDroid*

Bostani et al. [101] presented EvadeDroid, another problem space Android evasion attack. EvadeDroid is a query-efficient black-box attack, that can fool ML-based Android malware detectors without altering the functionality of the original malware samples. It uses an n-gram-based similarity method to select candidate donors for gadget extraction to change malware samples into benign ones through an iterative and incremental manipulation technique. Their experimental results demonstrated that EvadeDroid's evasion rates are 81, 73, 75, and 79 percent for DREBIN, Sec-SVM, MaMaDroid, and ADE-MA, respectively.

5.1.6 *EvnAttack*

EvnAttack is an evasion attack model that was proposed in [48] which manipulates an optimal portion of the features of a malware executable file in a bi-directional way such that the malware is able to evade detection from a machine learning model based on the observation that the API calls differently contribute to the classification of malware and benign files. The detection model's false negative ratio almost reached 1 (100 percent), which means almost all malware samples are misclassified.

5.1.7 *AdvAttack*

AdvAttack was proposed in [46] as a novel attack method to evade detection with the adversarial cost as low as possible. This is achieved by manipulating the API calls by injecting more of those features which are most relevant to benign files and removing those features with higher relevance scores to malware. AdvAttack increased the classifier's false negative ratio to 71 percent while degrade the accuracy of the classifier to 58.5 percent.

5.1.8 *MalGAN*

To combat the limitations of traditional gradient-based adversarial sample generation, the use of a generative adversarial network (GAN) based algorithm for generating adversarial samples has been proposed. Generative models have been mostly used for input reconstruction by encoding an original image into a lower-dimensional latent representation [2]. The latent representation of the original input can be used to distort the initial input to create an adversarial sample. MalGAN proposed by [102] leverages on generative modeling techniques to evade black-box malware detection systems with a detection rate close to zero.

5.1.9 *GAPGAN*

Yuan et al. [103] introduced GAPGAN, an adversarial attack framework that generates adversarial examples against binaries-based malware detection through GANs. Adversarial perturbations are appended to the original malware binaries to maintain its malicious functionality. They tested GAPGAN on deep learning and MalConv detectors. GAPGAN's success rate reached 100 percent attack with appending payloads of 2.5 percent of the total length of the original data.

5.1.10 *Black-box attacks against RNN based malware detection algorithms*

Hu et al. [56] implemented a generative recurrent neural network (RNN) which generates sequential adversarial samples. In their study, the Gumbel-Softmax approach is used to approximate generated discrete API's. Before their attack, the victim's RNN malware detection rates ranged from 90.74 to 93.87 percent. After their adversarial attack, the detection rates on adversarial examples ranged from 0.44 to 3.03 percent.

5.1.11 *Adversarial deep learning for robust detection of binary encoded malware*

Al-Dujaili et al. [104] proposed a method of generating adversarial malware samples with a focus on preserving the malicious functionality of the binary encoded files. They also introduce a mitigation framework known as SLEIPNIR which employs the saddle-point optimization technique to learn malware detection models.

5.1.12 *Deceiving end-to-end deep learning malware detectors using adversarial examples*

The authors Kreuk et al. [105] introduced a novel approach for creating adversarial malware samples by injecting a small sequence of bytes to the binary file. The approach was also found to be transferable across different malware files and families. In their study, they evaluated the effectiveness of adversarial malware samples based on five metrics namely (1) File transferability, (2) Spatial Invariance (3) payload size, (4) entropy (5) Functionality preservation. Their study was based on only white box attacks and was not evaluated as white box scenarios. Their injection procedure resulted in an evasion rate of 99.21 and 98.83 percent.

5.1.13 *Adversarial examples on discrete sequences for beating whole-binary malware detection*

The authors [106] focus on adversarial attacks against Convolutional Neural Network (CNN) based end to end malware detectors. End to end malware detectors such as Malconv [107] function quite different from most deep learning based malware detectors in the sense that they take the whole malware binary file as an input. To achieve their aim, a loss function was which functions as a surrogate loss function proposed which enforces the modifications in the embedding space. Thus, the authors were able to modify the embedding vector in order to reconstruct the modified binary, which becomes the adversarial malware sample. To preserve the functionality of the malware binary, a unique section of payload bytes is perturbed and appended to the original malware binary file instead of perturbing the original binary file. Thus by adding perturbations in the embedding vector space and reconstructing new binary files from the adversarial example. This attack's evasion rate reached 100 percent.

5.1.14 *Adversarial-example attacks toward android malware detection system*

MalGAN [102] proposed a black-box adversarial-example attacks toward Android malware detection, in which adversarial examples are generated using a generative adversarial network (GAN) without requiring the knowledge about the target. Unfortunately, the effectiveness of Malgan is affected, if a firewall is incorporated into the malware detection system. Adversarial attacks were also studied against cloud-based Android malware detection systems. Li et al. proposed a bi-objective GAN type adversarial attack against android malware detection systems. Their technique has the novelty of implementing a GAN with two discriminators in which one discriminator contends against the firewall while the other discriminator contends against the malware detector. This study was the first study to target a firewall-equipped Android malware detection system.

5.1.15 *Adversarial malware sample generation method based on the prototype of deep learning detector*

Qiaoa et al.[108] presented a method for generating adversarial malware to fool the deep learning-based malware detection systems. The post-hoc interpretability of deep learning is used by the authors to direct the malware file's updates. Based on their experiments, the time to generate their adversarial malware is less than other attacks. The fooling rate of this attack reached 92 percent.

5.1.16 *Slack attacks*

A byte-based convolutional neural network (MalConv) was introduced by Raff et al. [109]. Unlike image perturbation attacks [45], where the fidelity of the image is of little concern, attacks that alter the binaries of malware files must maintain the semantic fidelity of the original file because altering the bytes of the malware arbitrarily could affect the malicious effect of the malware. This problem could be solved by appending adversarial noise to the end of the binary [49]. This prevents the added noise from affecting the malware functionality. The Random Append attack and Gradient Append

attacks are two types of append attacks which work by appending byte values from a uniform distribution sample and gradually modifying the appended byte values using the input gradient value. Two additional variations of append attacks; the benign append and the FGM Append were introduced by Suciu et al. [110] which improves the long convergence time experienced in previous attacks. When malware binaries have exceeded the model's maximum size, it is impossible to append additional bytes to them. Hence a slack attack proposed by Suciu et al. [110] exploits the existing bytes of the malware binaries. The most common form of the slack attack is the Slack FGM Attack which defines a set of slack bytes that can be freely modified without breaking the malware functionality.

5.1.17 *Attack and defense of dynamic analysis-based, adversarial neural malware detection models*

Stokes et al. [111] proposed adversarial attacks against dynamic analysis-based malware detection systems. Their work focuses on different strategies of crafting adversarial samples for deep learning based dynamic analysis of malware samples. Their study is motivated in the fact that static analysis based deep learning malware classifiers only classify the content of the unknown file without execution, and become less effective when faced with packed or encrypted malware files. In addition, they propose a defense mechanism known as the weight defense mechanism. The compare their defence technique to existing defenses such as distillation and ensemble defenses. They however did not compare their study to the more popular approach of adversarial training, which is a proven method for reducing the vulnerability deep learning classifiers to adversarial samples. Their study also indicates that adding more hidden layers to the neural network significantly improves the robustness of the deep learning based malware classifier to adversarial samples.

5.2 *Adversarial attacks on spam detection*

Spam detection is a significant endpoint protection component, used to protect users from unsolicited digital communications. Machine learning techniques are widely used for current spam filtering applications, most of which utilize supervised learning methods [112]. Multiple adversarial attacks on machine learning-based spam detection systems are discussed below.

5.2.1 *Adversarial classification*

Dalvi et al [113] were the first to introduce a formal framework with corresponding algorithms to describe the problem of adversarial attacks against machine learning based spam detectors. In their study, they seek the minimum cost camouflage (MCC) of a data sample x to generate an adversarial sample MCC(x) with the minimum cost, for which the classifier outputs a negative sample. Similar studies [114] had considered adversarial attacks against spam detectors albeit not machine learning based.

5.2.2 *Attacks on statistical spam filters*

Several spam filters such as SpamAssasin, SpamBayes, Bogofilter are based on the popular Naive Bayes Machine learning algorithm which was first applied to filtering junk email in 1998 [115]. A variety of good word attacks introduced by Lowd [114] were successfully evading the machine learning models from detecting spam or junk emails. Using these attacks, an attacker can get 50 percent of currently blocked spam past a typical spam filter.

5.2.3 *Exploiting machine learning to subvert your spam filter*

Nelson et al. [116] showed in 2008 that an attacker could effectively disable the SpamBayes spam filter with small information and little control over training data. Their introduced Usenet dictionary poisoning attack caused misclassification of 36 percent of ham messages with only 1 percent control over the training data. They have also presented a new class of focused attacks that stop victims from receiving specific email messages. With knowledge of only 30 percent of the target's tokens, their focused attack altered the classification of the target email 60 percent of the time.

5.2.4 *Attacks against crowd-turfing detection systems*

Machine learning techniques are used to identify misbehavior includes fake users in social networks and detect users who pays for sites to have fake accounts. Malicious crowdsourcing or crowd-turfing systems are used to connect users who are willing to pay, with workers who carry out malicious activities such as generation and distribution of fake news, or malicious political campaigns. Machine learning models have been used to detect crowdturfing activity with up to 95 percent accuracy particularly in detecting the accounts of crowdturfing workers [117]. However, malicious crowdsourcing detection systems are highly vulnerable to adversarial evasion and poisoning attacks.

5.2.5 *Attacks against ML for keystroke dynamics*

Negi et al. [118] created adversarial keystroke samples that misled an otherwise accurate classifier into accepting the artificially generated keystroke samples as belonging to an authentic user. Almost 50 percent of the tested users were compromised after their attack.

5.2.6 *Attacks against ML for credit card fraud detection*

Zeager et al. [119] examined how a logistic regression classifier used as a fraud detection mechanism, could be adversarially attacked to cause a number of fraudulent transactions to go undetected. Previous studies have similar models which are based on game theory to investigate adversarial attacks against credit card fraud detection and email spam detectors. However, the authors introduced a new framework which successfully produced an improved AUC score on multiple iterations of the validation sets compared to the performance of the models which credit card companies had previously used.

5.2.7 *Crafting adversarial email content against machine learning based spam email detection*

Wang et al. [120] proposed two methods to create adversarial email content to bypass spam detectors. The first approach approximates the Term Frequency—Inverse Document Frequency) TF-IDF values in the resultant adversarial examples and the second method recognizes and adds a group of significant words to fool the detectors. They tested their work on multiple machine language models like; KNN, SVM, decision tree, and logistic regression, in both white-box and black-box attack scenarios. Their attacks' success rates ranged from 2.2 to 98.9 percent, which is inconclusive. However, they concluded that the second method is more effective.

5.2.8 *Marginal attacks of generating adversarial examples for spam filtering*

Zhaiquan et al. [121] created the marginal attack, which generates adversarial samples that can deceive naive bayesian spam filters by selecting sensitive words from a sentence and then add them at the end of the sentence. Their experiments showed that adding just one word to the message could reduce the model's accuracy from 93.6 to 55.8 percent. They also tested the transferability of the generated adversarial samples against standard machine learning filters like logic regression, decision tree, and linear support vector. In some cases, the accuracy of these filters could drop from 100 to 1.5 percent.

5.2.9 *Universal adversarial perturbations and image spam classifiers*

Phung et al. [122] evaluated numerous adversarial attack methods against deep learning-based image spam classifiers, and they found that the universal perturbation method is the most harmful. So they used this approach to create a novel transformation-based adversarial attack that was capable of creating tailored "natural perturbations" in image spam. In some cases, their suggested attack can lower the model's accuracy to reach 23.7 percent.

5.3 *Adversarial attacks against phishing detection*

Phishing detection is a critical endpoint protection element aimed to save the users from serious fraudulent actions like; money stealing and accessing private information. There are multiple techniques for phishing detection like [123];

List-base approach, Visual similarity-base approach, and Heuristics and machine learning-based approach, which is the most popular method now. Several adversarial attacks on machine learning-based phishing detection systems are discussed below.

5.3.1 *FIGA*

Gressel et al. [124] proposed the Feature Importance Guided Attack (FIGA) to fool phishing detection models by perturbing the most effective features of the input in the direction of the target class. It is a model-agnostic gray-box attack that needs knowledge of the feature representation of the victim model. FIGA was tested on eight different phishing detection models, and it reduced the F1-score of the models from 0.96 to 0.41 on average.

5.3.2 *Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks*

AlEroud et al. [125] presented an evasion technique that attacks URL phishing detection systems via Generative Adversarial Networks (GAN). Their generated samples can deceive Blackbox phishing detectors even when those detectors are created using refined methods like those relying on intra-URL similarities. Their experiments revealed that some classifiers were unable to identify any of the adversarial examples leading to zero true positive rates. At the same time, the false positive rates are increased, which indicates the percentage of benign examples classified as phishing.

5.3.3 *Generating optimal attack paths in generative adversarial phishing*

Al-Qurashi et al. [126] proposed a method that creates adversarial phishing attacks by discovering optimal subsets of features that lead to a higher evasion rate. To achieve this, multiple feature engineering techniques are used, such as Recursive Feature Elimination, Lasso, and Cancel Out. Their experiments revealed that their attack has better evasion capability than Generative Adversarial Deep Neural Network (GAN) which randomly perturbs features.

5.3.4 *Advanced evasion attacks and mitigations on practical ML-based phishing website classifiers*

Song et al. [127] introduced multiple mutation-based techniques, differing in the knowledge of the target classifier (white, gray, and black boxes). They also proposed a sample-based collision attack to acquire the knowledge of the target model, in the cases of white- and gray-box scenarios. Their evasion attacks fooled the classifiers without changing the functionalities and appearance of the samples. Their attack's success rate varied depending on the knowledge and the attacked model. Attacks on Google's phishing page filter achieved a 100 percent attack success rate. Their transferability attack on BitDe-fender's industrial phishing page classifier, TrafficLight, achieved 81.25 and 50 percent transferability attack rates in the black- and gray-box scenarios.

5.4 *Adversarial attacks against network anomaly detection*

Network anomaly detection devices learn network activity patterns and detect irregularities. They must continuously scan the network, analyze encrypted data, and spot anomalies in real-time. Machine learning ticks all these boxes, that's why it is used extensively in modern Network anomaly detection tools, however, researches have found some ways to attack them. Multiple of these adversarial attacks are discussed below.

5.4.1 *IDSGAN*

IDSGAN was proposed by Lin et al. [128] for generating adversarial attacks targeted towards intrusion detection systems. IDSGAN is based on the Wasserstein GAN [129] which uses a generator, discriminator and a black-box. The discriminator is used to imitate the black-box intrusion detection system and at the same time provide the malicious traffic samples. IDSGAN can lower the detection rates of some IDS models to approximately zero percent.

5.4.2 *TCP obfuscation techniques*

Another method for evading machine learning based intrusion detection systems is the use of obfuscation techniques. Homolial et al. [130] proposed the modification of various properties of network connections to obfuscate a TCP communication which successfully evades a wide variety of intrusion detection classifiers.

5.4.3 *Deep adversarial learning in intrusion detection: A data augmentation enhanced framework*

Zhang et al. [131] proposed a framework which incorporates deep adversarial learning with statistical learning in a manner which exploits learning-based data-augmentation. In the study, the Poisson-Gamma joint probabilistic generative model is used to synthesize adversarial samples.

5.4.4 *Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems*

A Generative adversarial network (GAN)—based adversarial attack was proposed by Usama et al. [67]. Their method was the first attempt to utilize GAN-based adversarial attacks against a black box Intrusion detection system (IDS) while still preserving the functional behavior of the network traffic. In some cases, their attack dropped the accuracy of the detection model from 84.3 to 43.4 percent.

5.4.5 *Adversarial deep learning for robust detection of binary encoded malware*

Al et al. [104], developed four adversarial attack methods to generate an adversarial example of a binary malware file that preserves its functionality (rFGSM, dFGSM, BCA, and BGA). They developed a framework for training robust malware detection models by utilizing the saddle-point formulation that consists of the inner maximization and outer maximization problems. The inner maximization approach is used to generate powerful adversarial examples that maximize the loss, and then they inject them in the training time. In some conditions, their attack's evasion rate exceeded 99 percent.

5.4.6 *Investigating adversarial attacks against network intrusion detection systems in SDNs*

With the increasing deployment of ML-based NIDSs which leverage the global network visibility offered by SDNs, the threat of vulnerability of the ML algorithms to adversarial attacks is also considered. Their study considered a use-case example of a SYN Flood DDoS attack, in which they demonstrated the ability to reduce the NIDS detection accuracy from 100% to 0% on multiple classifiers using evasion attacks. This was one of the most successful attempts of adversarial attacks against Network Intrusion Detections Systems, proposed by Aiken et al. [132]. Their experimental platform was based on ML based NIDS for Software defined networks called Neptune. In their study, they demonstrated that with the perturbation of a few features, the detection accuracy of a specific SYN flood Distributed Denial of Service (DDoS) attack by Neptune decreases from 100% to 0% across a number of classifiers. Furthermore, they proposed an adversarial test suite named Hydra to evaluate the impact of adversarial evasion classifiers against an anomaly-based NIDS—Neptune. Their study considered several classifiers and machine learning algorithms, proving that clustering algorithms were more robust to adversarial samples compared to other ML types. Specifically, KNN proved to be the most robust classifier against the adversarial attacks performed within their research, with only one combination of feature perturbations halving the detection accuracy from 100% to 50%. In contrast, Random forest, LR, and Support vector machines were generally vulnerable to the same perturbations resulting in similar detection accuracy reductions. The concept of attack generalization was also studied in this publication, using their Neptune NIDS framework as the adversarial target and which was capable of implementing multiple classifiers.

5.4.7 *IoT network security from the perspective of adversarial deep learning*

The effect of adversarial attacks on wireless sensor networks was studied by Sagduyu et al. [133]. The study experimented with adversarial attacks within the context of three types of over-the-air (OTA) wireless attacks, namely

within the jamming, spectrum poisoning, and priority violation attack. Their study demonstrated how adversarial attacks can lead to significant loss in throughput, by fooling an IoT transmitter into making a wrong transmit decision in the test phase. This was also an evasion attack against the machine learning model. In their study, they considered an IoT network where an IoT transmitter predicts if a channel status is idle or busy, by using deep learning algorithms. Their study showed that deep learning was effective in performing this task. Then, adversarial machine learning as applied in three contexts—jamming, spectrum poisoning and priority violation attakcs. A defense system based on stackelberg game showed to be an effective mitigation against adversarial machine learning against ioT networks. This defense technique is however considered not transferable as it was not proven to be generalizable across multiple adversarial attack scenarios.

Several uses of deep learning for anomaly detection in wireless communication systems have been commonly implemented including channel decoding [134], wireless resource allocation [135, 136] and radio signal (modulation) classification [137]. Uses of Machine Learning in IoT include anomaly detection [138], device identification [139, 140], and signal authentication [141].

5.4.8 *Adversarial attacks on deep-learning based radio signal classification*

The robustness of deep learning based algorithms for the wireless physical layer was also studied within the context of radio radio signal (modulation) classification tasks. Sadeghi [142] investigated the use of convolutional neural networks in which they developed both white-box and blackbox adversarial attacks for a DL based modulation classification. In their study, a VT-CNN was used as the classifier. The outcome of their research showed that Significantly less transmit power is required by the attacker in order to cause misclassification in the case of adversarial machine learning, as compared to the case of conventional jamming (where the attacker transmits only random noise). Hence, adversarial machine learning is an alternative to signal jamming with random noise, with less resource required in terms of transmit power. Their research also created a a computational efficient algorithm for crafting universal adversarial perturbations (UAP), which can cause a misclasification of the deep learning model irrespective of the input provided to the model. Furthermore, their study revealed an interesting property known as the Shift invariant Property of their attack method, which makes the attack generalizable across various deep learning models, without having any knowledge of the nature of the model, thus implying a black-box attack. Their tests showed that after applying these attacks, the targeted model accuracy could drop from 75 to 0 percent in the cases of a high perturbation-to-noise ratio (ratio of the perturbation power to the noise power).

5.4.9 *Addressing adversarial attacks against security systems based on machine learning*

Apruzzese et al. [143] proposed an attack and defense method against several types machine learning algorithms in for network intrusion detection systems. In their study, they evaluated both poisoning and evasive adversarial attacks against three supervised machine learning algorithms. The three algorithms namely Random forest, K-nearest neighbour and Artificial Neural Network (multi-layer perceptron) MLP were used to develop a network intrusion detection system. Their poisoning and evasion attack severity averaged 70.1 and 66.4 percent, respectively. They also demonstrated that adversarial training was effective in improving the robustness of deep learning based network intrusion detection systems.

5.4.10 *Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies*

Shi et al. [144] proposed an adversarial machine learning approach to launch jamming attacks on wireless communications and introduces a defense strategy. The study bases on the premise that in a cognitive radio network, a typical transmitter workflow includes the task of sensing available channels, identifying spectrum opportunities, and then transmitting data to the receiver in idle channels. As machine learning techniques have been progressively applied in this context, such as implementing a deep learning classifier for the classification of channels as either idle or busy, attackers seek to compromise the machine learning classifier. Even though the attacker has no knowledge of the deep learning classifier, i.e this is a black box attack. Their experiments showed that their adversarial deep learning attack reduced the transmission success rate from 73.79 to 2.91 percent. The authors also propose a defense technique for the deep learning classifier that works by allowing the transmitter to deliberately takes wrong actions in predetermined time slots in order to mislead the adversary.

5.4.11 *Performance evaluation of physical attacks against E2E autoencoder over rayleigh fading channel*

Albaseer et al. [69] investigated the vulnerabilities of autoencoder E2E with Rayleigh channel mode. Their study demonstrated the vulnerability of auntoencoder deep learning models to adversarial samples when used in end-to-end wireless communication systems. Both white-box and black box attacks were launched against and e2e model that was based on a realistic channel model. Their results showed that adversarial attacks had more significant impacts compared to jamming attack.

5.4.12 *Physical adversarial attacks against end-to-end autoencoder communication systems*

Sadeghi et al. [70] also showed that end to end learning of wireless communication systems are vulnerable to physical adversarial attacks. Similar to the work of Albaseer et al. [69], their study demonstrates that adversarial attacks are more destructive than jamming attacks.

5.4.13 *Targeted adversarial examples against RF deep classifiers*

Kokalj-Filipovic et al. [145] studied the effect of adversarial samples on machine learning based classifiers for radio frequency signals. The goal of their research was to verify if adversarial samples against machine learning based classification in of radio frequency signals was as effects in the physical world (i.e., when launched over the air—OTA) as it was in theoretical settings.

5.4.14 *Deep learning-based intrusion detection with adversaries*

Wang et al. [15] evaluated the vulnerabilities of deep learning-based IDS among state-of-the-art adversarial attack algorithms, including FGSM, JSMA, Deepfool, and CW using NSL-KDD dataset. They recognize feature patterns for the attack algorithms, and they demonstrated that modifying a limited number of features is better for most of the adversaries, such as JSMA attacks. JSMA attacks distinguish adversaries in terms of applicability. They noticed how feature selection to be perturbed by an adversary varies depending on the degree of significance.

5.4.15 *Evaluating deep learning-based network intrusion detection system in adversarial environment*

Peng et al. [146] evaluated the developed scalable ENIDS framework robustness in the adversarial environment against various attacks (MI-FGSM, L-BFGS, PGD, and SPSA) using NSD-KDD dataset. They compare different well-known models, including SVM, RF, and LR, with the proposed framework under adversarial attacks. They use different metrics to compare the model robustness, including accuracy (ACC), Precision Rate (PR), Recall Rate (RR), F-Sorce (FS), and Success Rate (SR).

5.4.16 *Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks*

Ibitoye et al. [147] studied the adversarial samples effectiveness against deep learning-based Intrusion Detection System (IDS) within the context of an IoT network. The authors provide a comprehensive comparison between two different deep learning model, a Self-normalizing Neural Network (SNN) and a Feed-forward Neural Network (FNN). They utilize and study input features normalization in a deep learning-based IDS in an adversarial environment. It increases the robustness of the deep learning model against various adversarial attacks (FGSM, BIM, and PGD).

5.4.17 *Online anomaly detection under adversarial impact*

Kloft et al. [148] studied the effect of a poisoning attack of training data on online centroid anomaly detection (IDS) with a finite sliding window. They study the poising attack with limited and full control of the training dataset using real HTTP traffic from a web server of Fraunhofer FIRST institute. This study shows if the attacker has full control of the data, is it easy to attack while when applying additional constraints to have limited control of the training data by assuming that attacker can inject a small fraction of the training dataset, the attack fails. Therefore, adding those constraints adds

protection approaches against poising attacks. Their results show that they cannot consider their method secure if the attacker has full control of the dataset.

5.4.18 *Security evaluation of pattern classifiers under attack*

Biggio et al. [149] proposed a framework for empirical security evaluation that can be applied in different three real-life applications, including Intrusion detection system, spam filtering, Biometric Authentication. They proposed an algorithm to sample training and testing sets. They evaluate their framework performance under causative adversarial attack using SVM and LR algorithm. For IDS, they used a public data set of a web-server with 205 malicious samples collected in five days in 2006. Authors recommend the designer of classifiers to follow to use their framework to evaluate the security of the classifier.

5.4.19 *Evading machine learning botnet detection models via deep reinforcement learning*

Wu et al. [150] introduced a generic black-box attack against botnet detection machine learning models. The authors of this paper use deep reinforcement learning (DRL) to generate adversarial traffic flows to deceive the detection models. A reinforcement learning agent updates the adversarial samples to change the temporal and spatial features of the traffic flows without altering the original functionality and executability. Their attack's evasion rate ranged from 69.3 to 80.4 percent.

5.4.20 *Attack-GAN*

Cheng et al. [151] proposed Attack-GAN to generate malicious adversarial raw packets that can mislead current machine learning network intrusion detection systems in the internet of things. Each byte in a packet is represented with word embedding. Feedback from the victim NIDS is needed by this black box attack to update the parameters of the generator. The attack success rate depends on multiple factors like the machine model and the modes of byte embedding, but it reached 98.42 percent in the best case.

5.4.21 *Fooling intrusion detection systems using adversarially autoencoder*

Chen et al. [152] introduced AIDAE (Anti-Intrusion Detection AutoEncoder) framework against IDSs. AIDAE can produce features matching normal feature distribution, it also keeps the correlation between the generated continuous and discrete features. They used Evasion Increase Rate (EIR) to evaluate their attack. The EIR reflects the evasion power by comparing the adversarial detection rate with the original, i.e., 1-(adversarial detection rate/original detection rate). EIR was higher than 0.9 in all their experiments.

5.4.22 *TANTRA*

Sharon et al. [153] presented TANTRA (Timing-Based Adversarial Network Traffic Reshaping) which deceives NIDSs by reshaping attack network traffic using the timestamp attribute. Based on the authors' evaluation, TANTRA had an extremely high success rate (99.99 percent). However, when TANTRA was tested after training the NIDSs with both benign and reshaped traffic, its success rate decreased.

6. Evaluating adversarial risk

In discussing adversarial risk, we introduce the concept of discriminative and directive autonomy of machine learning models. The two-fold goal of an adversarial risk grid mapping is to evaluate the likelihood of success of an adversarial attack against a machine learning model, and the consequence of that attack if successful. Adversarial risk often seek to measure the performance of a machine learning model based on worst case inputs [154]. We present in this paper, an adversarial risk grid map shown in Figure 8 based on the level of autonomy of the machine learning model with respect to

the learning technique and task. The concept of discriminative autonomy and directive autonomy of the machine learning models represents a novel approach for evaluating the relative adversarial risk of a machine learning model.

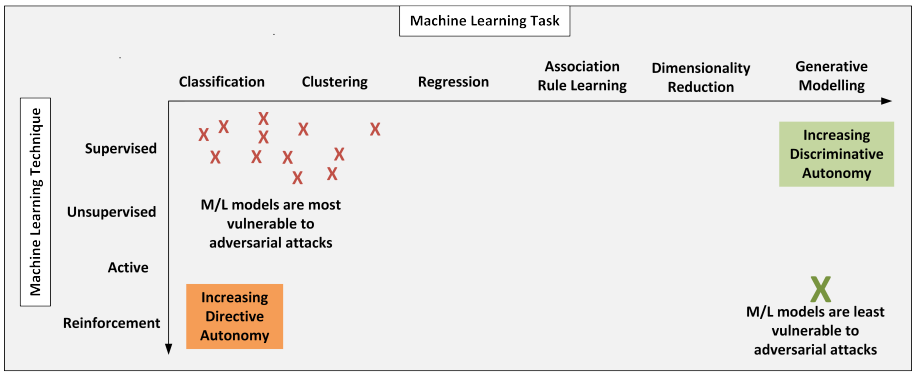

Figure 8. Adversarial risk grid map

6.1 *Security by obscurity in adversarial risk*

The notion of security by obscurity in adversarial context, in which defenses are proposed based on obscurity to an adversary does not truly reflect the nature of adversarial risk in machine learning-based network security applications. The prevalence of black box adversarial attacks which fool classifiers without having direct access to the model further demonstrate the weakness in the obscurity approach to adversarial risk.

As adversarial attacks continue to emerge into real world production systems, the ability to computationally evaluate and even optimize adversarial risk becomes invaluable. While both adversarial risk and obscurity have been impossible to compute directly [154], frameworks for adversarial risk based on the concept of obscurity have been proposed [155].

6.2 *Adversarial risk grid map*

A modified notion of adversarial risk was proposed in [156] which suggested that certain classifiers inherently have low adversarial risk. Other works [157, 158] have suggested a trade-off between standard risks and adversarial risk. This indicates that with increase in standard accuracy of the classifier, the adversarial risk of the classifier increases. Based on our review, a grid map based on the autonomy of the machine learning model is proposed. We term this as model autonomy adversarial risk approach since it is based on the directive and discriminative autonomy of the machine learning models. The map is shown in Figure 8.

- *Discriminative Autonomy:* The discriminative autonomy is directly related to the type of task being performed by the machine learning model. Machine learning tasks such as classification are highly dependent on the input data. As such, they have lower discriminative or conditional autonomy compared to tasks such as generative modeling which depend less on the input data when predicting an outcome.
- *Directive autonomy:* The directive autonomy of a machine learning model is a function of the machine learning technique. In supervised machine learning, there is less directive autonomy since the model needs to be first learned with some form of labeled data. Machine learning techniques such as reinforcement learning depend less on a model being learned with any form of training data and posses much higher directive autonomy.

6.3 *Cross model vs cross dataset attack*

In discussing adversarial risk, the notion of transferability becomes pertinent. Transferability refers to the fact in which an adversarial example which is crafted for a specific deep learning model, is found to be effective in causing a

misclassification in a different model. This is known as cross-model adversarial samples. in a similar situation, when the adversarial sample that was generated by altering a particular dataset. If that sample is used to attack a deep learning system that was trained using a different dataset, that is called a Cross-dataset adversarial sample.

7. Defending against adversarial attacks

Numerous researchers have aimed to review and classify defenses against adversarial attacks. Barreno et al. [8] first proposed three broad approaches for defending machine learning algorithms against adversarial attacks. Regularization, Randomization, and Information hiding. Yuan et al. [90] classified the defenses into two broad strategies. Proactive strategies and reactive strategies. Rosenberg et al. [9] organized the defenses based on the cyber security sub-domains (malware detection, spam detection, biometric systems, etc.), in our work, we classify the defenses based on generalized ML approaches.

Since adversarial examples represent a worst-case scenario of a distribution shift, the task of generating an adversarial sample is a non-convex optimization problem that can only be approximately solved. Adversarial attack methods are mostly optimization algorithms in search for a lower boundary perturbation that corresponds to an adversarial sample [159]. These optimization algorithms often result in high frequency outputs [160]. This however makes the defense methods against this adversarial samples vulnerable to adversarial samples that are generated within a low-frequency subspace.

In this section, we provide the most common defense methods in use today and classify them based on the strategy and approach. The reviewed defense methods are shown in Figure 9.

7.1 *Gradient masking*

Since most method of adversarial attacks are based on the using of gradient, the gradient masking method modifies a machine learning model in an attempt to obscure its gradient from an attacker. Nayebi et al. [161] demonstrated the effect of gradient masking by saturating the sigmoid network which results in a vanishing gradient effect in gradient-based attacks. Authors force the neural networks to works in nonlinear saturating system. By using Jacobian regularization for each network layer including the output layer, the model becomes non sensitive of perturbations that are generated using fast gradient sign method (FGSM) and iterative adversarial attacks [161]. However, [162] indicate that gradient masking react as over-fitting in their experiments.

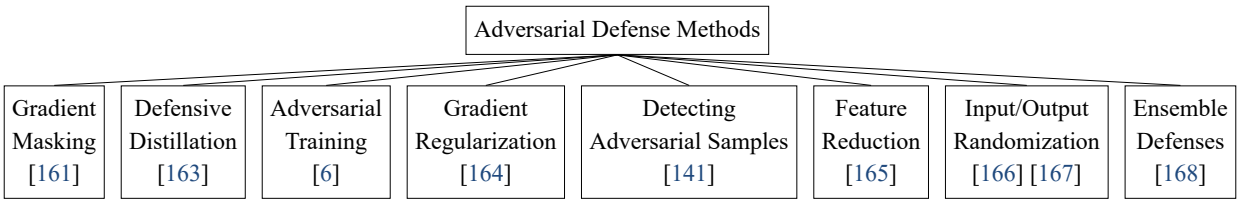

Figure 9. Adversarial defense methods

7.2 *Defensive distillation*

Distillation technique was originally proposed by Hinton et al. [163] for transferring knowledge from large neural networks to smaller ones. To implement the distillation approach, Hinton et al. authors built 10 DNN models with same architecture and training method and use soft targets to avoid overfitting that occur when using hard targets. They proved in their experiments that ensemble model is able to transfer knowledge to the distilled model better than individual models. However, ensemble requires large computation models that have large networks and large datasets. Therefore, they use learning specialist models that each use a subset of dataset classes to reduce the amount of computation [163]. Also, it was adapted by Papernot et al. [91] to defend against adversarial crafting by using the output of the original neural network

to train a smaller network rather than using the distillation as originally proposed by Hinton. Defensive distillation was initially tested against adversarial attacks in computer vision, but further research is required to determine its effectiveness in other applications such as malware detection.

7.3 *Adversarial training*

Adversarial training [6] is a method that aims to increase the robustness of a machine learning model to adversarial samples by minimizing the loss L on data/label pairs $\{X_i, y_i\}$ while maximizing the corresponding loss function. Szegedy et al. [45] originally proposed a three-step method known as adversarial training for defending against adversarial attacks. 1, Train the classifier on the original dataset 2, Generate adversarial samples 3, Iterate additional training epochs using the adversarial samples. Generally, adversarial training is based on min-max formulation that solves two problems: attacks as an inner maximization problem and defenses as an outer minimization problem to achieve optimization [6]. The inner maximization intents to generate adversarial samples version that results to maximize the model loss. Where the outer minimization intents to minimize the loss by finding model parameters that build a more robust model with less adversarial loss [6]. Numerous researchers tested and evaluated the effect of adversarial training in the network security domain [169, 170]. They concluded that it improves the classification performance of the machine learning model and makes it more resilient to adversarial crafting.

However, adversarial training has certain limitations particularly in the context of adversarial machine learning in network security. First, the adversary may implement a different attack method other than the one which was used in training the network. Secondly, the adversary may design adversarial perturbations for a deep learning model that already has been trained with adversarial training, and craft new adversarial perturbations which would make the previous adversarial training ineffective. It has also been shown that adversarial training can reduce the performance of the deep learning models on clean inputs as discussed in [70].

7.4 *Gradient regularization*

Gradient regularization is a technique that penalizes large changes in the output of some neural network layer, to adjust machine learning models, minimize the loss function, increase model robustness and prevent overfitting or underfitting. Many researchers tested this approach as a defense against adversarial attacks, like Ros et al. [164] who found that training DNNs with gradient regularization improves the robustness to adversarial perturbations as much or more than adversarial training. They have also found that combining both approaches (gradient regularization and adversarial training) achieves greater robustness. The main drawback of Gradient regularization is that it doubles the training time per batch.

7.5 *Detecting adversarial samples*

Several approaches are used to detect the presence of adversarial samples in the training phase of a machine-learning model. One of such approaches proposed by [141] works on the premise that adversarial samples have a higher uncertainty than clean data and uses a Bayesian neural network that is in dropout layers of neural networks to estimate the extent of uncertainty in the input data to detect the adversarial samples. Other approaches include the use of probability divergence proposed by [171] as well as the use of an auxiliary network of the original network introduced by Metzen et al. in [172]. Ren et al. [173] also proposed adversarial attack detection and adversarial sample recognition methods by using the causal inference technique to establish a causal model to describe the generation and performance of adversarial samples that attack DNNs.

7.6 *Feature reduction*

Other potential defenses for adversarial attacks have been proposed. Simple feature reduction was evaluated by Grosse et al. [165] but was found inadequate in defending against adversarial attacks. A more advanced approach was explored by elShehaby et al. [174], who assessed the “Perturb-ability” of features and assigned a score to each, reflecting

its susceptibility to perturbation in the problem-space of network intrusion detection systems. During the feature selection phase, they removed the susceptible features and reported promising results.

7.7 *Input/output randomization*

Some researchers have attempted randomization techniques on model inputs as a defense against adversarial attacks on machine learning. For instance, Xie et al. [166] explored random resizing and adding random padding to inputs, and their experiments demonstrated the effectiveness of this approach. Similarly, Zhang et al. [57] proposed injecting random Gaussian noise, offering advantages like simplicity, low computational complexity, and no requirement for additional training.

The primary limitation of input randomization in the network security domain's problem-space (e.g., executables, packets) is that it may alter input functionality. However, applying such randomization in the feature space could prove effective. We believe this method deserves further evaluation in the network security domain.

Meanwhile, other defenses have investigated output/confidence score randomization. In many black-box attacks, only the confidence values (outputs given by the model for a specific input) are known, while the model structure remains hidden. Kwon et al. [167] proposed a method called AdvGuard, which aims to prevent the creation of adversarial examples by adding noise to the softmax layer, where confidence values are generated. By providing random confidence values, AdvGuard aims to make black-box attacks that rely on confidence scores infeasible.

7.8 *Ensemble defenses*

Similar to the idea of ensemble learning which combines one or more machine learning techniques, researchers have also proposed the use of multiple defense strategies as a defense technique against adversarial samples. PixelDefend was proposed by [168] to combine adversarial detecting techniques with one or more other methods for creating a more robust defense against adversarial attacks. Another example of combining multiple defense techniques is Adaptive Continuous Adversarial Training (ACAT) [175, 176]. This defense merges adversarial training and detection by incorporating detected adversarial examples into continuous adversarial training sessions.

8. Discussion and lessons learnt

This section discusses several key lessons learnt through our survey on adversarial attacks against ML in network security.

8.1 *Increased adversarial risk*

We observed an increased risk of adversarial vulnerability of machine learning models in network security with reduced discriminative autonomy and directive autonomy. Similarly, we observed a reduced risk of adversarial vulnerability with increased discriminative autonomy and directive autonomy. As illustrated in the adversarial risk grid map shown in Figure 8, the discriminative autonomy directly relates to the machine learning tasks while the directive autonomy relates to the machine learning technique. The reason for the adversarial sensitivity of the machine learning models to the discriminative and directive autonomy based risk grid map is still an area of open research.

Previous approaches on making machine learning in network security more secure have advocated the development of machine learning models that are resilient to adversarial attacks. In this survey, we introduced the concept of an element of reduced risk of adversarial attacks based on an adversarial risk grid map. Our findings suggest that the adversarial risk grid map provides a promising future for the security of artificial intelligence and machine learning in network security. Machine learning based network security applications that are more resilient to adversarial attacks can be designed by leveraging on the adversarial risk grid map. We observed that the misclassification achieved by an adversarial attack is dependent significantly on the design of the adversarial attack algorithm with the context of each specific attack. White-box, Evasion attacks against endpoint protection systems (malware detection) are the most common attacks. While there is

limited research in adversarial attacks against process behavior and user behavior analysis, use cases of machine learning in network security, endpoint protection, network protection and application security have been well researched.

8.2 *Transferability with regards to machine learning technique*

Transferability of adversarial samples [74, 177] has been shown to be more effective with targeted adversarial samples [75]. This implies that non-targeted adversarial samples (reliability attacks) which are solely aimed at causing a misclassification, are more likely to transfer from one model to the other. In furtherance to this phenomenon, we observe that adversarial attacks in network security are less likely to transfer from one machine learning technique to another. Transferability of adversarial defences in network security in also impacted by to the heterogenous nature of the perturbed features. While this is has a positive side with regards to preventing transferable defenses, it also makes it more difficult in real world situations. From our observation, adversarial attacks in problem space are more difficult to generate, more difficult to defend against and less chances of being transferable.

In our research, we observed that a significant amount of features are perturbed in the process of generating the adversarial sample. This is a sub optimal approach. There is currently no publication which has explored the challenge of finding a way to identify the ideal features that need to be perturbed for creating adversarial samples. In the field of computer vision, Guo et al. [160] restricted the search for adversarial samples to the low frequency domain, thereby reducing query complexity.

We reviewed defenses against adversarial attacks on machine learning applications in network security. We note that there are two major limitations in the existing research on adversarial defenses. Firstly, most defenses are designed to protect against attacks on machine learning applications in computer vision. Secondly, the defenses studied are usually designed for a specific attack or a part of the attack. A generalized defense model against adversarial attacks is at best still theoretical as research on generalized defense models is in early stages [178]. Furthermore, our findings indicate that defenses against adversarial attacks are specific to a particular type of attack and are not necessarily transferable. Recent research [74] have studied the transferability in malware machine learning models in machine learning applications such as malware detection.

8.3 *Malware detection approaches*

In the majority of cases, Android malware detection is posed as a binary classification problem in which a classifier is used to determine whether an app is malicious or not. Malware detection take three general approaches which are dynamic, static, or hybrid. Significant overhead is usually required in order to extract dynamic features because it requires monitoring the behavior of apps at run time. Several of the studies we examined have focused on instances in which static features were extracted, including required permissions, actions, and application programming interface (API) calls. In our literature review, we did not come across any work in which adversarial attacks were successfully carried out against machine learning based malware detection systems in which dynamic features were extracted.

8.4 *Quantitative evaluation of adversarial attacks*

In network security, majority of the adversarial attacks reported target the integrity aspect of the CIA triad, with the intent of causing a misclassification. A quantitative analysis of the attacks' efficiency for the four reviewed categories (malware detection, phishing detection, spam detection, and network anomaly detection) was observed. After calculating the average attack success rate per class, we have found that the most significant adversarial effect was in the malware detection and the network anomaly detection domains, in which the adversarial attacks' success rates averaged more than 90 percent. It is worth mentioning that we think that a number of these attacks are theoretical and need more investigation to deploy them in practical settings, thus the quantitative effect of some of the reviewed attacks could be exaggerated. However, we find these results as a good indication of the malicious potential of adversarial attacks on network security domains.

The challenge of quantifying the efficiency of adversarial example generation, is an emerging field and several approaches have been proposed in recent literature. In [179] a new performance metric was proposed, called effective

generation rate (EGR) which is the ratio between n′ and n, i.e., n′/n. Where n represents the number of adversarial examples generated by an attacker and n′ denotes the number of adversarial examples that successfully evades both malware and adversarial example detection.

8.5 *Difference between adversarial attacks in network security and computer vision*

- In image recognition, the primary feature used in adversarial perturbation is the pixels of the image. However, in network security, there is a great variation in the types of features which may be used, and as such, the perturbation scope for adversarial attacks becomes largely increased.
- Adversarial attacks in network security differs from computer vision since data objects are considered rather than images. As a result, the perturbed features are more diverse and heterogeneous. The consequence is that defending against adversarial attacks in network security becomes more challenging due to the heterogeneity of the features, which in turn affects transferability and the effectiveness of universal defenses. Notably, significant progress has been made in computer vision in developing universal defenses; however, this remains an emerging research area in network security. Additionally, features can vary significantly depending on the network security application. In most cases, the features used for machine learning classification are the same ones perturbed when generating adversarial samples.

8.6 *Practicality of adversarial attacks against network security*

Recent studies have highlighted several challenges in launching practical adversarial attacks against certain network security systems. These challenges include:

- Attackers' limited access to the precise feature vectors used by some network security systems, such as NIDS, along with the impractical assumption that attackers possess detailed knowledge of model architecture, feature extraction techniques, or the ability to directly and freely query the ML models [180, 181]. Grosse et al. [182] highlighted that researchers often make overly generous assumptions about attackers' power and access to information, which may not reflect real-world conditions.
- Perturbations introduced in the problem space may alter the intended malicious behavior or interfere with the normal functioning of network traffic, posing a significant barrier to creating practical adversarial examples [180].
- The dynamic and complex nature of modern ML systems, with their shifting features and decision boundaries, presents substantial obstacles for adversarial attacks [180].
- The challenge of translating perturbations from feature space to meaningful modifications in the problem space (e.g., network packets)-known as the Inverse Feature-Mapping Problem [59]. This issue often limits the direct application of gradient-based feature-space attacks for generating effective problem-space adversarial examples. Additionally, there is no guarantee that a solution optimized in the problem space will closely approximate the intended adversarial feature vector [183].
- Minimal problem-space modifications can significantly alter numerous feature-space characteristics. These manipulations often introduce unintended side effects [59], known as collateral damage [174]. This collateral damage does not follow the gradient direction, adding unpredictability [180]. Consequently, these unintentional feature perturbations may negatively impact attack success, undermining the adversary's objectives [174].
- The predominant focus in current evaluations on the evasiveness of adversarial attacks rather than their real-world maliciousness and impact [183].
- Treating adversarial attacks on network security systems similarly to those in computer vision, where small, imperceptible perturbations are introduced. However, for network data, such minor modifications often have

minimal importance, unlike in computer vision, where perceptual similarity is crucial. In network security, the similarity constraint from computer vision is instead applied to the semantics of the attack, aiming to preserve network functionality and the malicious nature of the flow [174].

- Some problem-space perturbations can even be counterproductive, such as introducing delays between packets to evade intrusion detection systems, which may ultimately undermine the attack's effectiveness [174].

That said, the practicality of adversarial attacks against ML-based network security systems is not dismissed by any means. Factors such as the attacker's objective and their access to the target model's output can influence feasibility. For instance, attacks on domains like spam filters, where the attacker can query the model and modify input text without altering network functionality, may be more practical than attacks on intrusion detection systems with stricter constraints [175]. In summary, while adversarial attacks on ML-based network security systems remain an active research area, studies highlight some practical challenges that may limit their straightforward application in real-world network security scenarios. Careful consideration of the attack context and target domain is necessary to assess the true practicality of these techniques.

8.7 *Adversarial attacks against federated learning*

Federated Learning (FL) [184], distinct from distributed computation, allows each client to perform machine learning computations locally without transmitting data to the cloud. This approach enhances privacy and confidentiality compared to centralized learning, as the cloud provider lacks a complete view of the machine learning model. FL shows great promise in strengthening cybersecurity by enabling collaborative learning across decentralized devices [185]. By sharing timely insights into emerging threats like spoofing, intrusions, anomalies, and DDoS attacks, FL facilitates the development and refinement of robust defense models and mechanisms, strengthening cybersecurity at both device and network levels. However, despite its inherent support for privacy and security, FL remains susceptible to adversarial attacks [186]. Byzantine attacks disrupt model convergence by sending random messages or conspiring to create faulty models [187], with even a single compromised client significantly degrading accuracy. Sybil attacks involve creating fake or compromised clients to poison or manipulate the shared global model [188], particularly in open systems where devices can join and leave freely. Based on adversarial intent, attacks can be classified as semi-honest, where attackers passively observe data to extract private information, or malicious, involving active protocol deviations that result in data manipulation or playback. Furthermore, attacks may occur during the training phase, where adversaries use data or model poisoning to compromise learning, or during the inference phase, where attackers aim to deceive models into making incorrect predictions. The decentralized nature of FL, especially model broadcasting, amplifies vulnerabilities, particularly in evasion attacks at both server and client levels.

9. Conclusions and future work

We present a first of its kind survey on adversarial attacks on machine learning in network security. The previous survey [17] that we reviewed had only discussed adversarial attacks against deep learning in computer vision. We introduced a new classification for adversarial attacks based on applications of machine learning in network security and developed a matrix to correlate the various types of adversarial attacks with a taxonomy-based classification to determine their effectiveness in causing a misclassification. We also presented a novel idea of the concept of an adversarial risk grid map for machine learning in network security.

In our review on defenses against adversarial attacks, although there were numerous proposed defenses against specific adversarial attacks, research on generalized defenses against adversarial attacks is still not well established [178]. In our future work, we would study generalized defenses against adversarial attacks to understand if a generalized approach towards adversarial defenses will be effectively attainable. In addition, we would examine the interpretability of the adversarial risk to further understand why the reduced adversarial vulnerability occurs, and its implications for other applications of machine learning such as computer vision and natural language processing.

Future Work: Based on our research, most adversarial attacks to date have been conducted on data at rest, with relatively few successful attempts targeting data in transit, or streaming data, as seen in studies such as [189, 190]. However, in the domain of network security, particularly in areas like intrusion detection, adversarial attacks are more likely to occur on data in transit. This makes it critical to investigate how adversarial perturbations can affect streaming data in real-time, as these attacks may have significantly different characteristics and impacts compared to those on static datasets. Given the dynamic nature of network traffic and the low-latency requirements of real-time systems, defending against adversarial attacks on data in transit presents unique challenges that are not fully addressed in current literature. Consequently, further research is needed to explore the specific risks posed by adversarial attacks on data in transit, including the development of effective detection and defense techniques that can mitigate these threats without compromising system performance.

Certified Robustness is a defense approach against adversarial attacks. It is an essential consideration for deploying models in critical applications in domains like network security. Certified Robustness offers guaranteed security for neural networks operating in adversarial environments [191]. The primary goal of neural network robustness certification is to assess whether a neural network alters its predictions when changes are introduced to its inputs [192].

Defending against adversarial attacks in network security remains a complex challenge due to the heterogeneity of features across different applications, which impacts both the transferability of attacks and the effectiveness of universal defenses. While significant strides have been made in developing universal defenses for adversarial attacks in computer vision, this remains an emerging and relatively under-explored area within network security. Additionally, the features used for machine learning classification in network security systems can vary significantly depending on the specific application, further complicating defense strategies. In most cases, the features that are essential for classification are also the ones targeted and perturbed during the generation of adversarial samples. This creates a unique set of obstacles that need to be addressed, highlighting the need for more tailored defense mechanisms that account for the diverse and dynamic nature of network security features.

Adversarial attacks were demonstrated to affect only classifier and clustering tasks in network security. From the reviewed literature of over fifty attacks against machine learning in network security, there has been no attempt to implement adversarial attacks against any other task in network security except classification and clustering tasks. This is consistent with our adversarial risk grid map illustrated in Figure 8 in which we posit that adversarial risk increases based on the type of network security task which is being performed. Our study notes that there are diverse adversaries in network security compared to computer vision. as such, there is even more relevant arms race situation in network security than in computer vision

Several authors have shown that deep learning can be performed on data that is encrypted [193, 194, 195]. But in our study, we observe that encrypted data has not been adversarial defeated. Even though, most data in network security is encrypted, adversarial attacks or the ability to generate adversarial samples against encrypted data is an area of open research. As such, it is a promising idea, subject to future research, to stipulate that performing encryption before applying machine learning to the data, is a trusted and proven defense against adversarial machine learning in network security.

The use of deep learning as a technique for encryption is quite restrictive [196]. This is mostly due to the computational costs of deep learning. Research is also required to understand the effects of adversarial attacks against deep learning for encryption.

Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the NSERC Discovery Grant program.

Conflict of interests

There is no conflict of interest declared by the authors.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv*, 2014, arXiv:1412.6572.

[2] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *Proc. IEEE Sec. Priv. Wks. (SPW)*, San Francisco, CA, USA, May 24, 2018, pp. 36–42.

[3] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comp. Vis. Pat. Recog.*, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 2574–2582.

[4] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Sec. Priv. (EuroS&P)*, Saarbruecken, Germany, Mar. 21–24, 2016, pp. 372–387.

[5] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Inf. Sci.*, vol. 239, pp. 201–225, 2013.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv*, 2017, arXiv:1706.06083.

[7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comp. Comm. Sec.*, Abu Dhabi, United Arab Emirates, Apr. 2–6, 2017, pp. 506–519.

[8] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf., Comp. Comm. Sec.*, Taipei, Taiwan, Mar. 21–24, 2006, pp. 16–25.

[9] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, 2021.

[10] V. Duddu, "A survey of adversarial machine learning in cyber warfare," *Def. Sci. J.*, vol. 68, no. 4, pp. 356–366, 2018.

[11] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surv. & Tutor.*, vol. 21, no. 3, pp. 2224–2287, 2019.

[12] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, 2018.

[13] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv*, 2016, arXiv:1606.06565.

[14] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synthesis Lect. Artif. Intell. Mach. Learn.*, vol. 12, no. 3, pp. 1–169, 2018.

[15] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38,367–38,384, 2018.

[16] X. Liu, Y. Lin, H. Li, and J. Zhang, "Adversarial examples: Attacks on machine learning-based malware visualization detection methods," *arXiv*, 2018, arXiv:1808.01546.

[17] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *arXiv*, 2018, arXiv:1801.00553.

[18] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Appl. Sci.*, vol. 9, no. 5, p. 909, 2019.

[19] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6, pp. 12,103–12,117, 2018.

[20] N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: a systematic review," *IEEE Access*, vol. 8, pp. 35,403–35,419, 2020.

[21] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 2, pp. 1153–1176, 2015.

[22] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," *ACM Comput. Surv.*, vol. 49, no. 3, p. 59, 2016.

[23] H. Xu, Y. Ma, H. C. Liu, D. Deb, H. Liu, J. L. Tang, et al., "Adversarial attacks and defenses in images, graphs and text: A review," *Int. J. Autom. Comput.*, vol. 17, no. 2, pp. 151–178, 2020.

[24] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–41, 2020.

[25] L. Sun, M. Tan, and Z. Zhou, "A survey of practical adversarial example attacks," *Cybersecurity*, vol. 1, no. 1, p. 9, 2018.

[26] V. Ford and A. Siraj, "Applications of machine learning in cyber security," in *Proc. 27th Int. Conf. Comp. Appl. Ind. Eng.*, New Orleans, LA, USA, Oct. 13–15, 2014.

[27] J. Singh and M. J. Nene, "A survey on machine learning techniques for intrusion detection systems," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 11, pp. 4349–4355, 2013.

[28] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Proc. IEEE 15th Int. Symp. Intell. Syst. Inform. (SISY)*, Subotica, Serbia, Sept. 14–16, 2017, pp. 277–282.

[29] A.-C. Sima, K. Stockinger, K. Affolter, M. Braschler, P. Monte, and L. Kaiser, "A hybrid approach for alarm verification using stream processing, machine learning and text analytics," in *Proc. Int. Conf. Ext. Database Tech. (EDBT)*, Vienna, Austria, Mar. 26–29, 2018.

[30] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?" in *Proc. Int. Conf. Image Anal. Process.*, Cagliari, Italy, Sept. 6–8, 2005, pp. 50–57.

[31] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Proc. Australasian Joint Conf. Artif. Intell.*, Hobart, TAS, Australia, Dec. 5–8, 2016, pp. 137–149.

[32] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *J. Comput. Secur.*, vol. 19, no. 4, pp. 639–668, 2011.

[33] A. Kumara and C. Jaidhar, "Automated multi-level malware detection system based on reconstructed semantic view of executables using machine learning techniques at vmm," *Future Gener. Comput. Syst.*, vol. 79, pp. 431–446, 2018.

[34] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Methods*, vol. 9, no. 5, 2015.

[35] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," in *Soft Computing Applications in Industry*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 373–383.

[36] Y. Dong, Y. Zhang, H. Ma, Q. Wu, Q. Liu, K. Wang, et al., "An adaptive system for detecting malicious queries in web attacks," *Sci. China Inf. Sci.*, vol. 61, no. 3, p. 032114, 2018.

[37] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "Urlnet: Learning a url representation with deep learning for malicious url detection," *arXiv*, 2018, arXiv:1802.03162.

[38] K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, P. S. T. Magalhães, and H. D. d. Santos, "A machine learning approach to keystroke dynamics based user authentication," *Int. J. Electron. Sec. Digit. Forensics.*, vol. 1, no. 1, pp. 55–70, 2007.

[39] E. Bursztein, M. Martin, and J. Mitchell, "Text-based captcha strengths and weaknesses," in *Proc. 18th ACM Conf. Comp. Comm. Sec.*, Chicago, IL, USA, Oct. 17–21, 2011, pp. 125–138.

[40] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, Geneva, Switzerland, Jul. 19–23, 2010, pp. 435–442.

[41] M. Kravchik and A. Shabtai, "Anomaly detection; industrial control systems; convolutional neural networks," *arXiv*, 2018, arXiv:1806.08110.

[42] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *Proc. 2004 IEEE Int. Conf. Netw. Sens. Control.*, Taipei, Taiwan, Mar. 21–23, 2004, pp. 749–754.

[43] Y. G. Şahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in. *Proc. Int. MultiConf. Eng. Comp. Sci. 2011*, Hong Kong, Mar. 16–18, 2011, vol. I.

[44] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ascad database," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 53, 2018.

[45] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, et al., "Intriguing properties of neural networks," *arXiv*, 2013, arXiv:1312.6199.

[46] L. Chen and Y. Ye, "Secmd: Make machine learning more secure against adversarial malware attacks," in *Proc. Australasian Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, Aug. 19–20, 2017, pp. 76–89.

[47] L. Chen, S. Hou, Y. Ye, and S. Xu, "Droideye: Fortifying security of learning-based classifier against adversarial android malware attacks," in *Proc. 2018 IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining. (ASONAM)*, Barcelona, Spain, Aug. 28–31, 2018, pp. 782–789.

[48] L. Chen, Y. Ye, and T. Bourlai, "Adversarial machine learning in malware detection: Arms race between evasion attack and defense," in *Proc. 2017 Eur. Intell. Sec. Inform. Conf. (EISIC).*, Athens, Greece, Sept. 11–13, 2017, pp. 99–106.

[49] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, et al., "Adversarial malware binaries: Evading deep learning for malware detection in executables," *arXiv*, 2018, arXiv:1803.04173.

[50] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, et al., "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proc. 10th ACM Work. Artif. Intell. Sec.*, Dallas, TX, USA, Nov. 3, 2017, pp. 27–38.

[51] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, et al., "Evasion attacks against machine learning at test time," in *Proc. Jt. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Prague, Czech Republic, Sept. 23–27, 2013, pp. 387–402.

[52] J. Jo and Y. Bengio, "Measuring the tendency of cnns to learn surface statistical regularities," *arXiv*, 2017, arXiv:1711.11561.

[53] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv*, 2019, arXiv:1903.12261.

[54] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, "Adversarial machine learning," in *Proc. 4th ACM Work. Sec. Artif. Intell.*, Chicago, IL, USA, Oct. 21, 2011, pp. 43–58.

[55] Y. Fan, B. Wu, T. Li, Y. Zhang, M. Li, Z. Li, et al., "Sparse adversarial attack via perturbation factorization," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, UK, Aug. 23–28, 2020, pp. 35–50.

[56] W. Hu and Y. Tan, "Black-box attacks against RNN based malware detection algorithms," in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2–7, 2018.

[57] Y. Zhang and P. Liang, "Defending against whitebox adversarial attacks via randomized discretization," in *Proc. 22nd Int. Conf. Artif. Intell. Stat.*, Okinawa, Japan, Apr. 16–18, 2019, pp. 684–693.

[58] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, et al., "On evaluating adversarial robustness," *arXiv*, 2019, arXiv:1902.06705.

[59] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ML attacks in the problem space," in *Proc. 2020 IEEE Symp. Secur. Priv. (SP)*, San Francisco, CA, USA, May 18–21, 2020, pp. 1332–1349.

[60] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," *arXiv*, 2017, arXiv:1712.03632.

[61] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, "A taxonomy and terminology of adversarial machine learning," *NIST IR*, vol. 2019, pp. 1–29, 2019.

[62] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, et al., "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, Quebec, Canada, Dec. 2–8, 2018, pp. 6103–6113.

[63] H. Kwon, H. Yoon, and K.-W. Park, "Selective poisoning attack on deep neural networks," *Symmetry*, vol. 11, no. 7, p. 892, 2019.

[64] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proc. 25th {USENIX} Security Symposium ({USENIX} Security 16)*, Austin, TX, USA, Aug. 10–12, 2016, pp. 601–618.

[65] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "Sok: Security and privacy in machine learning," in *Proc. 2018 IEEE Eur. Symp. Secur. Priv. (EuroS&P)*, London, UK, Apr. 24–26, 2018, pp. 399–414.

[66] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," in *Proc. 2020 54th Ann. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 18–20, 2020, pp. 1–6.

[67] M. Usama, M. Asim, S. Latif, J. Qadir, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *Proc. 2019 15th Int. Wirel. Commun. Mob. Comput. Conf. (IWCMC)*, Tangier, Morocco, Jun. 24–28, 2019, pp. 78–83.

[68] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: protecting against dnn model stealing attacks," in *Proc. 2019 IEEE Eur. Symp. Secur. Priv. (EuroS&P)*, Stockholm, Sweden, Jun. 17–19, 2019, pp. 512–527.

[69] A. Albaseer, B. S. Ciftler, and M. M. Abdallah, "Performance evaluation of physical attacks against e2e autoencoder over rayleigh fading channel," in *Proc. 2020 IEEE Int. Conf. Informat. IoT Enabl. Technol. (ICIoT)*, Doha, Qatar, Feb. 2–5, 2020, pp. 177–182.

[70] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 847–850, 2019.

[71] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *arXiv*, 2018, arXiv:1804.08598.

[72] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *Proc. 29th {USENIX} Security Symposium ({USENIX} Security 20)*, Boston, MA, USA, Aug. 12–14, 2020.

[73] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Denver, CO, USA, Oct. 12–16, 2015, pp. 1322–1333.

[74] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv*, 2016, arXiv:1605.07277.

[75] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv*, 2016, arXiv:1611.02770.

[76] J. R. Correia-Silva, R. F. Berriel, C. Badue, A. F. de Souza, and T. Oliveira-Santos, "Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data," in *Proc. 2018 Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 8–13, 2018, pp. 1–8.

[77] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 15–20, 2019, pp. 4954–4963.

[78] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv*, 2012, arXiv:1206.6389.

[79] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv*, 2017, arXiv:1708.06733.

[80] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," *arXiv*, 2015, arXiv:1511.05122.

[81] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv*, 2016, arXiv:1607.02533.

[82] U. Jang, X. Wu, and S. Jha, "Objective metrics and gradient descent algorithms for adversarial examples in machine learning," in *Proc. 33rd Ann. Comput. Secur. Appl. Conf.*, Orlando, FL, USA, Dec. 4–8, 2017, pp. 262–277.

[83] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2–7, 2018.

[84] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. 2017 IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 22–26, 2017, pp. 39–57.

[85] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," *arXiv*, 2020, arXiv:2003.01690.

[86] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 21–26, 2017, pp. 1765–1773.

[87] A. Ghiasi, A. Shafahi, and T. Goldstein, "Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates," *arXiv*, 2020, arXiv:2003.08937.

[88] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv*, 2017, arXiv:1712.09665.

[89] F. Croce, M. Andriushchenko, and M. Hein, "Provable robustness of relu networks via maximization of linear regions," in *Proc. 22nd Int. Conf. Artif. Intell. Stat.*, Apr. 16–18, 2019, pp. 2057–2066.

[90] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *arXiv*, 2017, arXiv:1712.07107.

[91] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. 2016 IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 22–26, 2016, pp. 582–597.

[92] M. Mosbach, M. Andriushchenko, T. Trost, M. Hein, and D. Klakow, "Logit pairing methods can fool gradient-based attacks," *arXiv*, 2018, arXiv:1810.12042.

[93] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," *arXiv*, 2019, arXiv:1907.02044.

[94] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," *arXiv*, 2019, arXiv:1904.02144.

[95] A. K. Das, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for internet of things," *Future Gener. Comput. Syst.*, vol. 89, pp. 110–125, 2018.

[96] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Comput. Sec.*, vol. 24, no. 1, pp. 31–43, 2005.

[97] Total malware. Accessed: Dec. 6, 2024. [Online]. Available: https://www.av-test.org/en/statistics/malware/.

[98] A. Calleja, A. Martín, H. D. Menéndez, J. Tapiador, and D. Clark, "Picking on the family: Disrupting android malware triage by forcing misclassification," *Exp. Sys. Appl.*, vol. 95, pp. 113–126, 2018.

[99] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks," in *Proc. 27th {USENIX} Sec. Symp. ({USENIX} Sec. 18)*, Baltimore, MD, USA, Aug. 15–17, 2018, pp. 1299–1316.

[100] K. S. Han, J. H. Lim, B. Kang, and E. G. Im, "Malware analysis using visualized images and entropy graphs," *Int. J. Inf. Sec.*, vol. 14, no. 1, pp. 1–14, 2015.

[101] H. Bostani and V. Moonsamy, "Evadedroid: A practical evasion attack on machine learning for black-box android malware detection," *arXiv*, 2021, arXiv:2110.03301.

[102] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv*, 2017, arXiv:1702.05983.

[103] J. Yuan, S. Zhou, L. Lin, F. Wang, and J. Cui, "Black-box adversarial attacks against deep learning based malware binaries detection with gan," in *ECAI 2020*. Amsterdam, The Netherlands: IOS Press, 2020, pp. 2536–2542.

[104] A. Al-Dujaili, A. Huang, E. Hemberg, U. M. O'Reilly, "Adversarial Deep Learning for Robust Detection of Binary Encoded Malware," in *Proc. 2018 IEEE Sec. Priv. Wks. (SPW)*, San Francisco, CA, USA, May 24, 2018, pp. 76–82.

[105] F. Kreuk, A. Barak, S. Aviv-Reuven, M. Baruch, B. Pinkas, and J. Keshet, "Deceiving end-to-end deep learning malware detectors using adversarial examples," *arXiv*, 2018, arXiv:1802.04528.

[106] F. Kreuk, A. Barak, S. Aviv-Reuven, M. Baruch, B. Pinkas, J. Keshet, "Adversarial examples on discrete sequences for beating whole-binary malware detection," *arXiv*, 2018, pp. 490–510, arXiv:1802.04528.

[107] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole exe," in *Proc. Wks. 32nd AAAI Conf. AI*, New Orleans, LA, USA, Feb. 2–7, 2018.

[108] Y. Qiao, W. Zhang, Z. Tian, L. T. Yang, Y. Liu, and M. Alazab, "Adversarial malware sample generation method based on the prototype of deep learning detector," *Comput. Secur.*, vol. 119, p. 102762, 2022.

[109] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas, "Malware detection by eating a whole exe," *arXiv*, 2017, arXiv:1710.09435.

[110] O. Suciu, S. E. Coull, and J. Johns, "Exploring adversarial examples in malware detection," *arXiv*, 2018, arXiv:1810.08280.

[111] J. W. Stokes, D. Wang, M. Marinescu, M. Marino, and B. Bussone, "Attack and defense of dynamic analysis-based, adversarial neural malware detection models," in *Proc. MILCOM 2018—2018 IEEE Mil. Comm. Conf. (MILCOM)*, Los Angeles, CA, USA, Oct. 29–31, 2018, pp. 1–8.

[112] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *J. Big Data*, vol. 2, no. 1, pp. 1–24, 2015.

[113] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, "Adversarial classification," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Seattle, WA, USA, Aug. 22–25, 2004, pp. 99–108.

[114] D. Lowd and C. Meek, "Good word attacks on statistical spam filters." in *Proc. Second Conf. Email Anti-Spam (CEAS)*, Stanford, CA, USA, Jul. 21–22, 2005.

[115] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*. AAAI Technical Report WS-98-05; Palo Alto, CA, USA: AAAI, 1998, pp. 98–105.

[116] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, et al., "Exploiting machine learning to subvert your spam filter," *LEET*, vol. 8, no. 1, p. 9, 2008.

[117] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers," in *Proc. 23rd {USENIX} Sec. Symp. ({USENIX} Sec. 14)*, San Diego, CA, USA, Aug. 20–22, 2014, pp. 239–254.

[118] P. Negi, A. Sharma, and C. Robustness, "Adversarial machine learning against keystroke dynamics," 2017. Accessed: Dec. 6, 2024. [Online]. Available: https://cs229.stanford.edu/proj2016/report/NegiSharma-AdversarialMachineLearningAgainstKeystrokeDynamics-report2.pdf.

[119] M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, "Adversarial learning in credit card fraud detection," in *Proc. 2017 Syst. Inform. Eng. Des. Symp. (SIEDS)*, Charlottesville, VA, USA, Apr. 28, 2017, pp. 112–116.

[120] C. Wang, D. Zhang, S. Huang, X. Li, and L. Ding, "Crafting adversarial email content against machine learning based spam email detection," in *Proc. 2021 Int. Symp. Adv. Sec. Softw. Syst.*, Virtual Event, Hong Kong, Jun. 7, 2021, pp. 23–28.

[121] G. Zhaoquan, X. Yushun, H. Weixiong, Y. Lihua, H. Yi, and T. Zhihong, "Marginal attacks of generating adversarial examples for spam filtering," *Chin. J. Electron.*, vol. 30, no. 4, pp. 595–602, 2021.

[122] A. Phung and M. Stamp, "Universal Adversarial Perturbations and Image Spam Classifiers," in *Malware Analysis Using Artificial Intelligence and Deep Learning*. Heidelberg, Germany: Springer, 2021, pp. 633–651.

[123] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *arXiv*, 2020, arXiv:2009.11116.

[124] G. Gressel, N. Hegde, A. Sreekumar, and M. Darling, "Feature importance guided attack: A model agnostic adversarial attack," *arXiv*, 2021, arXiv:2106.14815.

[125] A. AlEroud and G. Karabatis, "Bypassing detection of url-based phishing attacks using generative adversarial deep neural networks," in *Proc. 6th Int. Works. Secur. Privacy Analyt.*, New Orleans, LA, USA, Mar. 18, 2020, pp. 53–60.

[126] R. Al-Qurashi, A. AlEroud, A. A. Saifan, M. Alsmadi, and I. Alsmadi, "Generating optimal attack paths in generative adversarial phishing," in *Proc. 2021 IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, San Antonio, TX, USA, Nov. 2–3, 2021, pp. 1–6.

[127] F. Song, Y. Lei, S. Chen, L. Fan, and Y. Liu, "Advanced evasion attacks and mitigations on practical ml-based phishing website classifiers," *Int. J. Intell. Syst.*, vol. 36, no. 9, pp. 5210–5240, 2021.

[128] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," *arXiv*, 2018, arXiv:1809.02077.

[129] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv*, 2017, arXiv:1701.07875.

[130] I. Homoliak, M. Teknos, M. Ochoa, D. Breitenbacher, S. Hosseini, and P. Hanacek, "Improving network intrusion detection classifiers by non-payload-based exploit-independent obfuscations: An adversarial approach," *arXiv*, 2018, arXiv:1805.02684.

[131] H. Zhang, X. Yu, P. Ren, C. Luo, and G. Min, "Deep adversarial learning in intrusion detection: A data augmentation enhanced framework," *arXiv*, 2019, arXiv:1901.07949.

[132] J. Aiken and S. Scott-Hayward, "Investigating adversarial attacks against network intrusion detection systems in sdns," in *Proc. 2019 IEEE Conf. Netw. Funct. Virtualiz. Softw. Defined Netw. (NFV-SDN)*, Dallas, TX, USA, Nov. 12–14, 2019, pp. 1–7.

[133] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Iot network security from the perspective of adversarial deep learning," in *Proc. 2019 16th Annu. IEEE Int. Conf. Sens., Commun. Netw. (SECON)*, Boston, MA, USA, Jun. 10–13, 2019, pp. 1–9.

[134] F. Liang, C. Shen, and F. Wu, "An iterative bp-cnn architecture for channel decoding," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 144–159, 2018.

[135] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. 2017 IEEE 18th Int. Works. Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 3–6, 2017, pp. 1–6.

[136] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, Aberdeen, UK, Sept. 2–5, 2016, pp. 213–226.

[137] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, 2018.

[138] J. Canedo and A. Skjellum, "Using machine learning to secure iot systems," in *Proc. 2016 14th Annu. Conf. Privacy, Secur. Trust (PST)*, Auckland, New Zealand, Dec. 12–14, 2016, pp. 219–222.

[139] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, et al., "Profiliot: a machine learning approach for iot device identification based on network traffic analysis," in *Proc. Symp. Appl. Comput.*, Marrakech, Morocco, Apr. 3–7, 2017, pp. 506–509.

[140] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *Proc. 2017 IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 5–8, 2017, pp. 2177–2184.

[141] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv*, 2017, arXiv:1703.00410.

[142] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 1, pp. 213–216, 2018.

[143] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *Proc. 2019 11th Int. Conf. Cyber Conf. (CyCon)*, Tallinn, Estonia, May 28–31, 2019, pp. 1–18.

[144] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *Proc. 2018 IEEE Int. Conf. on Comm. Works. (ICC Works.)*, Kansas City, MO, USA, May 20–24, 2018, pp. 1–6.

[145] S. Kokalj-Filipovic, R. Miller, and J. Morman, "Targeted adversarial examples against rf deep classifiers," in *Proc. ACM Works. Wireless Sec. Mach. Learn.*, Miami, FL, USA, May 15–17, 2019, pp. 6–11.

[146] Y. Peng, J. Su, X. Shi, and B. Zhao, "Evaluating deep learning based network intrusion detection system in adversarial environment," in *Proc. 2019 IEEE 9th Int. Conf. Electron. Info. Emerg. Comm. (ICEIEC)*, Beijing, China, Jul. 12–14, 2019, pp. 61–66.

[147] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," in *Proc. 2019 IEEE Global Comm. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 9–13, 2019, pp. 1–6.

[148] M. Kloft and P. Laskov, "Online anomaly detection under adversarial impact," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, Sardinia, Italy, May 13–15, 2010, pp. 405–412.

[149] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, 2013.

[150] D. Wu, B. Fang, J. Wang, Q. Liu, and X. Cui, "Evading machine learning botnet detection models via deep reinforcement learning," in *Proc. ICC 2019—2019 IEEE Int. Conf. Comm. (ICC)*, Shanghai, China, May 20–24, 2019, pp. 1–6.

[151] Q. Cheng, S. Zhou, Y. Shen, D. Kong, and C. Wu, "Packet-level adversarial network traffic crafting using sequence generative adversarial networks," *arXiv*, arXiv:2103.04794, 2021.

[152] J. Chen, D. Wu, Y. Zhao, N. Sharma, M. Blumenstein, and S. Yu, "Fooling intrusion detection systems using adversarially autoencoder," *Digit. Comm. Netw.*, vol. 7, no. 3, pp. 453–460, 2021.

[153] Y. Sharon, D. Berend, Y. Liu, A. Shabtai, and Y. Elovici, "Tantra: timing-based adversarial network traffic reshaping attack," *arXiv*, 2021, arXiv:2103.06297.

[154] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli, "Adversarial risk and the dangers of evaluating against weak attacks," *arXiv*, 2018, arXiv:1802.05666.

[155] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 18–23, 2018, pp. 1778–1787.

[156] A. S. Suggala, A. Prasad, V. Nagarajan, and P. Ravikumar, "Revisiting adversarial risk," in *Proc. 22nd Int. Conf. Artif. Intell. Stat.*, Naha, Japan, Apr. 16–18, 2019, pp. 2331–2339.

[157] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," in *Proc. Adv. Neural Inform. Process. Syst.*, Palais des Congrès de Montréal, Canada, Dec. 2–8, 2018, pp. 1178–1187.

[158] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "There is no free lunch in adversarial robustness (but there are unexpected benefits)," *arXiv*, vol. 2, no. 3, 2018, arXiv:1805.12152.

[159] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv*, 2017, arXiv:1712.04248.

[160] C. Guo, J. S. Frank, and K. Q. Weinberger, "Low frequency adversarial perturbation," *arXiv*, 2018, arXiv:1809.08758.

[161] A. Nayebi and S. Ganguli, "Biologically inspired protection of deep networks from adversarial attacks," *arXiv*, 2017, arXiv:1703.09202.
[162] Y. Yanagita and M. Yamamura, "Gradient masking is a type of overfitting," *Int. J. Mach. Learn. Comput.*, vol. 8, no. 3, pp. 203–207, 2018.
[163] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv*, 2015, arXiv:1503.02531.
[164] A. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2–7, 2018.
[165] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv*, 2016, arXiv:1606.04435.
[166] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv*, 2017, arXiv:1711.01991.
[167] H. Kwon and J. Lee, "Advguard: fortifying deep neural networks against optimized adversarial example attack," *IEEE Access*, vol. 12, pp. 5345–5356, 2020.
[168] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv*, 2017, arXiv:1710.10766.
[169] R. Abou Khamis, M. O. Shafiq, A. Matrawy, "Investigating resistance of deep learning-based ids against adversaries using min-max optimization," in*Proc. ICC 2020—2020 IEEE Int. Conf. Comm. (ICC)*, Dublin, Ireland, Jun. 7–11, 2019.
[170] R. Abou Khamis and A. Matrawy, "Evaluation of adversarial training on different types of neural networks in deep learning-based idss," in *Proc. 2020 Int. Symp. Netw., Comput. Comm. (ISNCC)*, Montreal, QC, Canada, Oct. 20–22, 2020, pp. 1–6.
[171] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proc. 2017 ACM SIGSAC Conf. Comput. Comm. Secur.*, Dallas, TX, USA, Oct. 30–Nov. 3, 2017, pp. 135–147.
[172] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv*, arXiv:1702.04267, 2017.
[173] M. Ren, Y.-L. Wang, and Z.-F. He, "Towards interpretable defense against adversarial attacks via causal inference," *Mach. Intell. Res.*, vol. 19, no. 3, pp. 209–226, 2022.
[174] M. elShehaby and A. Matrawy, "Introducing perturb-ability score (ps) to enhance robustness against evasion adversarial attacks on ml-nids," *arXiv*, 2024, arXiv:2409.07448.
[175] M. elShehaby, A. Kotha, and A. Matrawy, "Introducing adaptive continuous adversarial training (acat) to enhance machine learning robustness," *IEEE Netw. Lett.*, vol. 6, no. 3, pp. 208–212, 2024.
[176] A. Kotha, A. Matrawy, "Adaptive continuous adversarial training (acat) to enhance ml-nids robustness," Accessed: Nov. 2024. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.173144803.35072777/v1.
[177] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv*, arXiv:1704.03453, 2017.
[178] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Proc. Adv. Neural Inform. Process. Syst.*, Palais des Congrès de Montréal, Canada, Dec. 2–8, 2018, pp. 5014–5026.
[179] H. Li, S. Zhou, W. Yuan, J. Li, and H. Leung, "Adversarial-example attacks toward android malware detection system," *IEEE Syst. J.*, vol. 14, no. 1, pp. 653–656, 2019.
[180] M. e. Shehaby and A. Matrawy, "Adversarial evasion attacks practicality in networks: Testing the impact of dynamic learning," *arXiv*, 2023, arXiv:2306.05494.
[181] M. El Shehaby and A. Matrawy, "The impact of dynamic learning on adversarial attacks in networks (ieee cns 23 poster)," in *Proc. 2023 IEEE Conf. Comm. Netw. Secur. (CNS)*, Orlando, FL, USA, Oct. 2–5, 2023, pp. 1–2.
[182] K. Grosse, L. Bieringer, T. R. Besold, and A. M. Alahi, "Towards more practical threat models in artificial intelligence security," in *Proc. 33rd USENIX Secur. Symp. (USENIX Secur. 24)*, Philadelphia, PA, USA, Aug. 14–16, 2024, pp. 4891–4908.
[183] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: a comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 25, no. 1, pp. 538–566, 2023.
[184] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[185] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8229–8249, 2022.
[186] A. K. Nair, E. D. Raj, and J. Sahoo, "A robust analysis of adversarial attacks on federated learning environments," *Comput. Stand. Interfaces*, vol. 86, p. 103723, 2023.
[187] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, et al., "Ibm federated learning: an enterprise framework white paper v0. 1," *arXiv*, 2020, arXiv:2007.10987.
[188] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv*, 2018, arXiv:1808.04866.
[189] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-time, universal, and robust adversarial attacks against speaker recognition systems," in *Proc. ICASSP 2020—2020 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 4–8, 2020, pp. 1738–1742.
[190] Y. Gong, B. Li, C. Poellabauer, and Y. Shi, "Real-time adversarial attacks," *arXiv*, 2019, arXiv:1905.13399.
[191] L. Li, T. Xie, and B. Li, "Sok: Certified robustness for deep neural networks," in *Proc. 2023 IEEE Symp. Secur. Priv. (SP)*, San Francisco, CA, USA, May 21–25, 2023, pp. 1289–1310.
[192] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 9–15, 2019, pp. 1310–1320.
[193] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning," in *Proc. 2018 Netw. Traf. Meas. Anal. Conf.*, Vienna, Austria, Jun. 26–29, 2018, pp. 1–8.
[194] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: Deep neural networks over encrypted data," *arXiv*, 2017, arXiv:1711.05189.
[195] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Comput.*, vol. 24, no. 3, pp. 1999–2012, 2020.
[196] E. Klein, R. Mislovaty, I. Kanter, A. Ruttor, and W. Kinzel, "Synchronization of neural networks by mutual learning and its application to cryptography," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2005, pp. 689–696.