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

Olakunle Ibitoye, Rana Abou-Khamis, Ashraf Matrawy and M. Omair Shafiq

School of Information Technology, Carleton University, Ottawa, Canada

Email: {Kunle.Ibitoye, Rana.Aboukhamis, Ashraf.Matrawy, Omair.Shafiq}@carleton.ca

*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*

## I. INTRODUCTION

There has been an ever-increasing application of machine learning and deep learning techniques in network security. 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 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 illustrated in Fig. **??**.

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.

We structure the remainder of the paper as follows. In Section II, we survey some related work. In section III, we provide some introduction into machine learning basics.

In section IV, we begin with a brief background about adversarial machine learning followed by a description of our adversarial threat model. We also review different adversarial attack methods and algorithms. In section V, 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 VI, we discuss and evaluate adversarial risk in machine learning. In section VII, we review various approaches for defending against adversarial attacks. In section VIII, we provide some discussion and lessons learnt. Finally, in section IX, we add a conclusion for our survey with guidance for future work.

## II. 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 [9]. In network security, the consequences of adversarial attacks are equally significant [10] especially in areas such as intrusion detection [11] and malware detection [12] 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. [13] which reviewed adversarial attacks against deep learning in computer vision. Qui et al [14] 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. [15] reviewed security threats and corresponding defensive techniques of machine learning focusing on the threats in the learning algorithms. Duddu el al. in [16] discussed various research work on adversarial machine learning in cyberwarfare, with some mention of adversarial attacks against malware classifiers. Zhang et al. [17] 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. [18] 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 [19] 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. [20] 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 has also been published including Hao et al. [21] in which various adversarial attacks and defenses in images, graphs and texts were reviewed. In the field of natural language processing, zhang et.al [22] reviewed various publications in which deep adversarial attacks and defenses were proposed. Sun et al. [23] published a survey on adversarial machine learning in graph data. Akhtar et al. [13] computer vision, Duddu et al. [16] cyber warfare.

**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.

## III. MACHINE LEARNING BASICS

Machine learning enables computers learn to solve specific tasks and make predictions based on past observations [24]. Machine learning algorithms vary significantly, and can be grouped by the learning technique, the task similarity in performing functions, or the depth of learning. This is illustrated in Figure 1.

### A. Machine Learning Technique

We classify machine learning algorithms based on the technique in which the model is trained with data. The learning technique of the machine learning algorithm has a direct relationship with the directive autonomy of the model discussed in section VI.
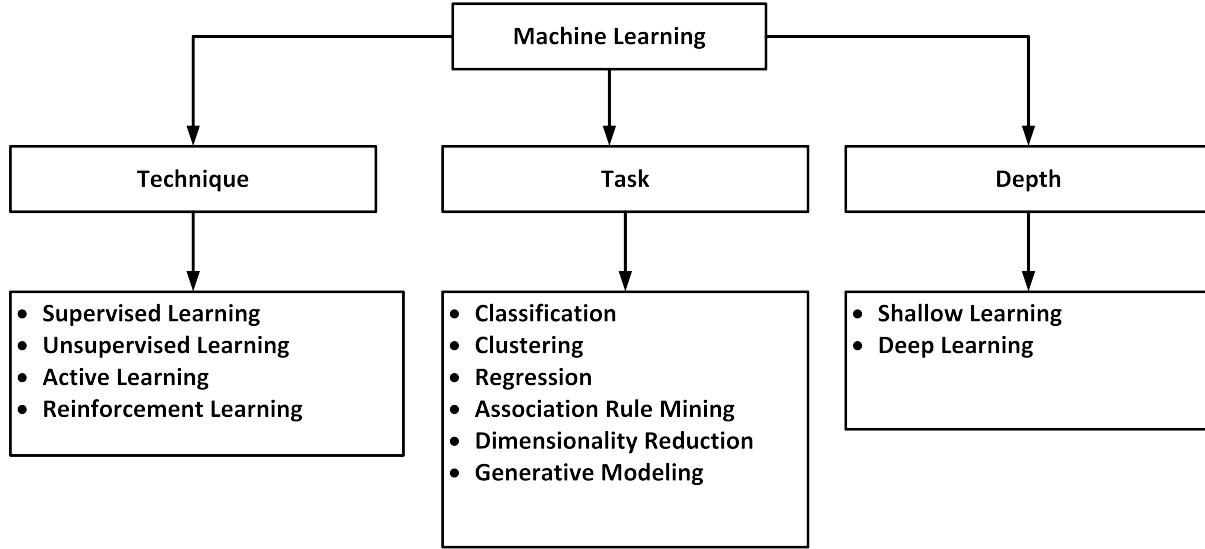
Fig. 1. Three Dimensions of Machine Learning Classification

*1) Supervised Learning:* In supervised machine learning, the model learns from a training dataset that consists of a labeled input and desired output pairs. It generates a mapping function that maps between the input (x) and output (y) by analyzing the training dataset to produce a mapping function [25]. Typical applications of supervised learning are for Regression and Classification tasks.

*2) Unsupervised Learning:* For certain applications where a labelled dataset is not readily available, a different approach to learning is required. Unsupervised learning techniques train a model without providing a labeled input or any output variable to be predicted [26]. Unsupervised learning may be used for clustering some input data based on the information and characteristic of the data. Dimensionality Reduction and Association Rule Learning are typical applications of unsupervised learning.

*3) Semi-Supervised Learning:* In semi-supervised learning, a large amount of unlabeled data with labeled data is used to achieve a better classifier model [27]. Usually, classifiers are trained by using labeled data that consist of input and output pairs and features. Collecting labeled data is often hard, expensive, time-consuming and requires experienced user input [27]. Unlabeled data is easy to collect, but they are limited in terms of usage. samples of tasks that make use of semi-supervised learning include Regression and Classification.

*4) Active Learning:* Active learning allows for selection of the training data actively and with extra flexibility. This reduces the need for a large amount of labeled data by influencing the selection of data required for training [28]. The primary motivation of active learning starts from the cost and time of collecting labeled training dataset.

*5) Reinforcement Learning:* Reinforcement learning exposes and interacts with its environment and learns from the consequences of its action using trial and error. It is trained to make accurate decisions for the future action by capturing the learned knowledge and its experience [26].

*6) Ensemble Learning:* Ensemble learning combines multiple weak classifiers to create a stronger classifier model [29] by taking their individual decisions and their predictions to combine them. Boosting and Bagging are samples of ensemble learning.

*B. Machine Learning Depth*

Schmidhuber et al. [30] classify machine learning into shallow and deep learning which distinguishes the machine learning techniques based on how deep the credit assignment path is.

*1) Shallow Learning:* Shallow learning refers to the approach of standard machine learning models which do not utilize multiple hidden connection or layers. Shallow learning models do not suffer from vanishing gradient and the complexity of computations that come from the growth of connections. However, shallow models are usually limited and unable to capture correlation across the modulates [31].

*2) Deep Learning:* Deep learning involves the use of a multi-layer stack of simple modules [32]. Deep Learning overcomes scalability and complicated problems and is mostly being used for solving major critical scientific related problems on a large scale [13].

*C. Machine Learning Tasks*

Machine learning is used to perform various types of tasks based on the required approach and the type of data available. All machine learning techniques can be divided into six task categories as illustrated in Figure 2.

*1) Classification Tasks in Network Security:* Classification tasks involve categorizing data into different categories from pre-labeled samples [33]. In network security, classification task model is used in detecting known types of fraud [34], and for grouping different users like in social spammers [35]. Also, it is used to categorize programs and files as malware, spyware, and ransomware, and to identify different classes
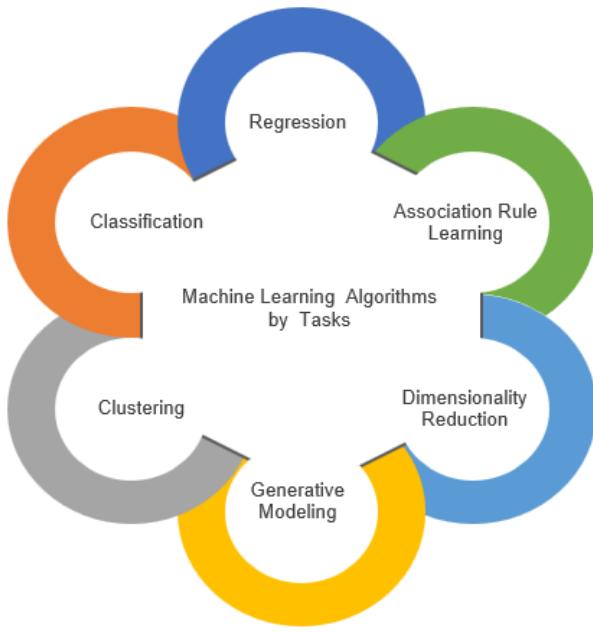
Fig. 2. Machine Learning Tasks

of network attacks. Example of algorithms for classification tasks are Logistic regression, Decision Trees, Random Forests, Artificial neural networks, Support vector machines and Convolutional Neural Networks.

*2) Clustering Task in Network Security:* Clustering tasks are used to group the input data by similarity or patterns into unknown classes [26]. Clustering task is used to compare industry and business processes [36] and detect outliers. Girma et al. used clustering to detect DDOS attacks [37]. Also, the clustering task is used in forensic analysis. samples of clustering algorithms are K-nearest neighbors, K-means, Mixture model, Self-organized Maps (SOM) and Kohonen Networks.

*3) Regression (Prediction) Tasks in Network Security:* Regression tasks involve methods for predicting next series of information from prior data [24]. In network security, a regression model is used to predict relevant parameters from network packet and then draw a comparison between them with the regular parameters [38]. Kolosnjaji et al. [39] used regression models for predicting system calls for executable processes to derive a relationship between the actual processes. Also, regression algorithms are used for anomaly detection in computer networks, user behavior analytics (e.g., Human Interaction Proofs) and predicting anomalies in process behaviour [34] such as credit card fraud transactions. Example of ML algorithms for regression tasks are Linear regression, Polynomial regression, Ridge regression, Support Vector Regression, Decision trees, Random Forest. Deep Learning algorithms for regression tasks include Artificial neural networks, Recurrent neural networks, Neural tuning machines and Differentiable Neural Computer.

*4) Association Rule Learning (Recommendation) Task in Network Security:* Association Rule Learning (ARL) in-

volves the discovery of rules and relations that describe large portions of data and find the link between X and Y where the X is the antecedent and Y is the consequence of rule [26]. Common ARL algorithms include Apriori, Euclat, and Deep belief networks.

*5) Dimensional Reduction (Generalization) Task In Network Security:* Dimensionality reduction encodes a multi-dimensional dataset into a compact lower dimensional representation while preserving as much information as possible in the original dataset. Example algorithms are Principal Component Analysis, Singular Value Decomposition, Linear Discriminant Analysis, Independent Component Analysis.

*6) Generative Modeling Task In Network Security:* Generative modeling tasks involve training a model by learning the data distribution within a training dataset. Subsequently, new data points are generated and associated decisions are made to simulate an entirely new data sample. samples of algorithms for generative modeling tasks include Markov Chains, Variational Auto-encoders, Generative Adversarial Networks (GANs) and Boltzmann Machines.

## IV. 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 [40] 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.

### A. 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) [41]. Signature based IDS detects attacks based on the repository of attacks signatures with no false alarm [42]. However, zero-day attacks can easily bypass signature-based IDS. Anomaly IDS [42] 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.

- Hybrid Approach for Alarm Verification Sima et al. [43] 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 effective-

Fig. 3. Machine Learning Applications in Network Security

ness of their system where the accuracy achieves more than 90% in a stream of 30K alarms per second [43].

- Learning Intrusion Detection Laskov et al. [44] 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 [44].

### B. 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 [39].

- Automatic Analysis of Malware Behavior Rieck et al. [45] 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 [45].
- Automated Multi-level Malware Detection System In [46], authors proposed Advanced Virtual Machine Monitor-based guest-assisted Automated Multi-level 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 [46]. 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. [39] 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 [39].
- A Hybrid Malicious Code Detection Method Li et al. [47] 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 [47].

*C. 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. [48] 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% [48].

- Adaptively Detecting Malicious Queries in Web Attacks Don et al. [49] 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 [49].
- URLNet -Learning a URL Representation with Deep Learning for Malicious URL Detection Le et al. [50] 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 [50].

*D. 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. [51] 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 [51] 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. [52] 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. [35] 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.

### E. 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.[36] 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 [36] 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 [53]. 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 [53]. Chen et al. [34] 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. [54] 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 [54]. 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 [54] 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 [54].

### V. ADVERSARIAL MACHINE LEARNING

Adversarial attacks have been studied for more than a decade now [19]. However, the first notable discovery in adversarial attacks for computer vision was by Szegedy et al. [55] 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 [56][57][58][59].

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) [60] or during the inference phase (evasion attack) [61].

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 [62] [63] . 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

Fig. 4. Adversarial Machine Learning

was proposed in [64] 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. 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

Considering an input x, and a classifier f, the optimization goal for the adversary is to compute such perturbation with a s mall 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 $\delta$ is the perturbation.

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.

### A. Adversarial Threat Model

We examine the threat model in Figure 5 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] [64] 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.

*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 [65]. 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 [66], 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
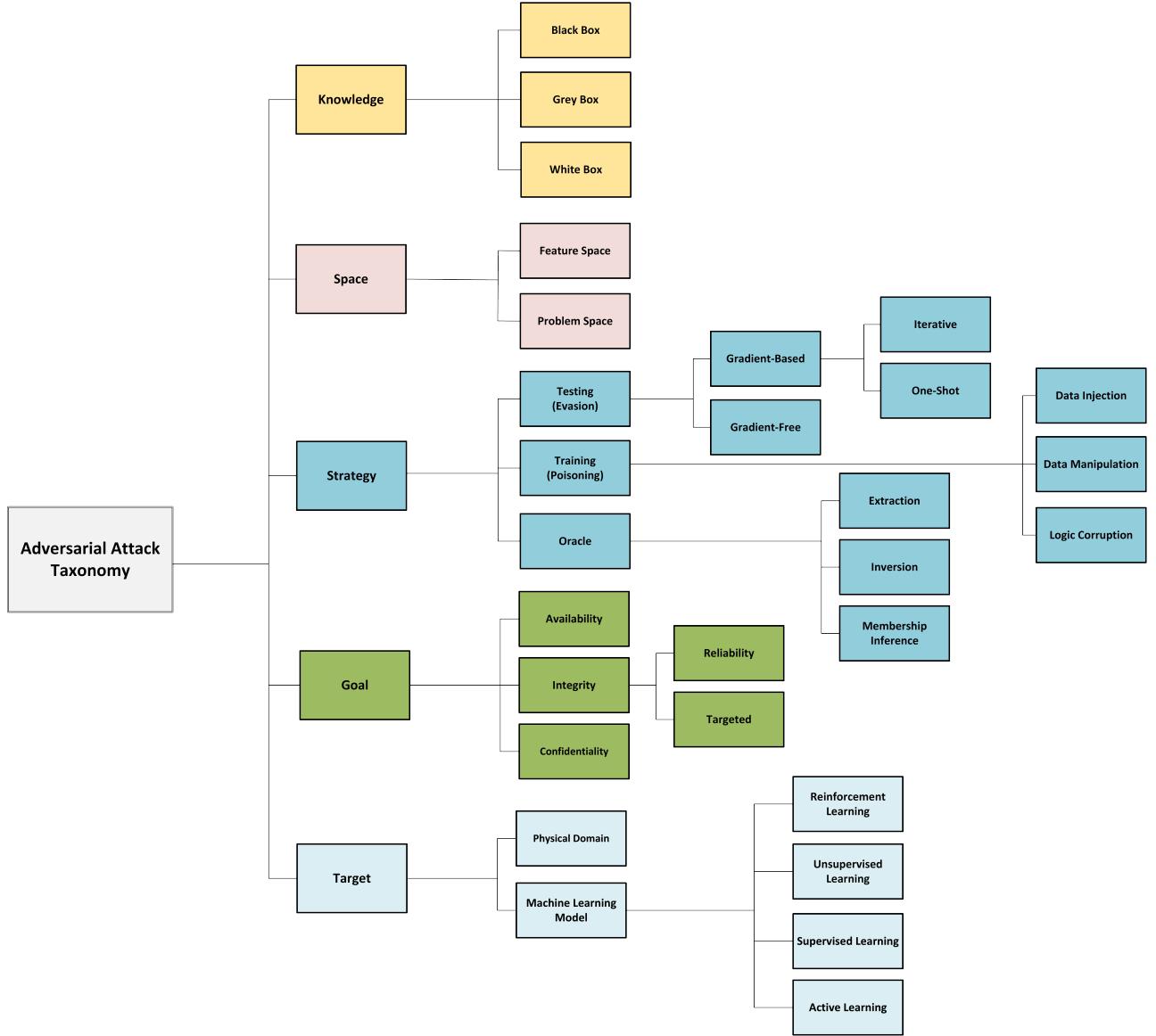
Fig. 5.   Adversarial attack classification

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.

*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 [67]. 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.

*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 6. 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 [68]. 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 [69].

  Gradient-free attacks [65], 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.

- 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 7. Poisoning attacks may be carried out by data injection, data manipulation or logic corruption
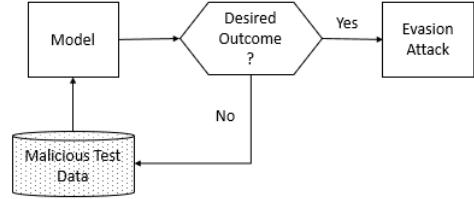


Fig. 6.    Evasion Attack

[68]. 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.
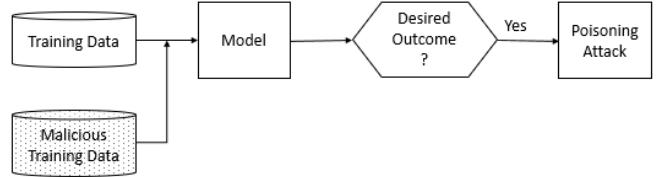


Fig. 7.    Poisoning Attack

- 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 [70]. As a result, the substitute model can then be used for other types of attacks such as evasion attacks [68]. 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 [71]. 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 [72].

*4) Goal:* Traditionally in the field of computer vision, adversarial attacks are regarded in terms of targeted or reliability attacks [13]. 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. [10] have noted that the distinction between reliability and targeted attacks becomes blurred in attacks on binary

10

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.

*5) Target:* We consider the model which is being targeted in our threat model based on the various machine learning techniques which we discussed in section III. In our surveyed work, adversarial attacks are targeted against a specific machine learning technique. Several successful attempts have been made towards transferability of adversarial attacks [73] [74]. 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.

*B. 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 V-A, 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. Our classification method is based on the adversary strategy described in section V-A.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. [55] 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 [75]. The Basic Iterative Method (BIM) proposed in [76] 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. [77] 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. [78] developed Carlini and Wagner Attack, a targeted attack specifically for existing adversarial defense methods. It was discovered that defenses such as defensive distillation [79] were ineffective towards the Carlini and Wagner attack. Madry et al. [6] pro-

**Adversarial Attack Algorithms**

**Evasion**

| | | |
|---|---|---|
| PGD [6] | L-BFGS [37] | FGM [1] |
| Feature Adversaries [75] | BIM [68] | JSM [4] |
| Decision Tree Attack [77] | DeepFool [3] | NewtonFool [69] |
| EAD [37] | Carlini & Wagner [70] | Auto Attack [71] |
| UAP [76] | Shadow Attack [79] | Adversarial Patch [80] |
| QeBB [76] | VAM [70] | Square Attack [71] |
| ZOO [37] | DbBA [79] | Threshold Attack [80] |
| Hop Skip Jump [37] | SimBA [37] | DPatch [37] |
| TUAP [37] | Wasserstein [37] | Spatial Transformation [37] |
| Elastic Net [37] | IFS [37] | |

**Poisoning**

| | |
|---|---|
| SVM Poisoning [81] | Backdoor Attack [82] |
| Feature Collision [83] | |

**Oracle**

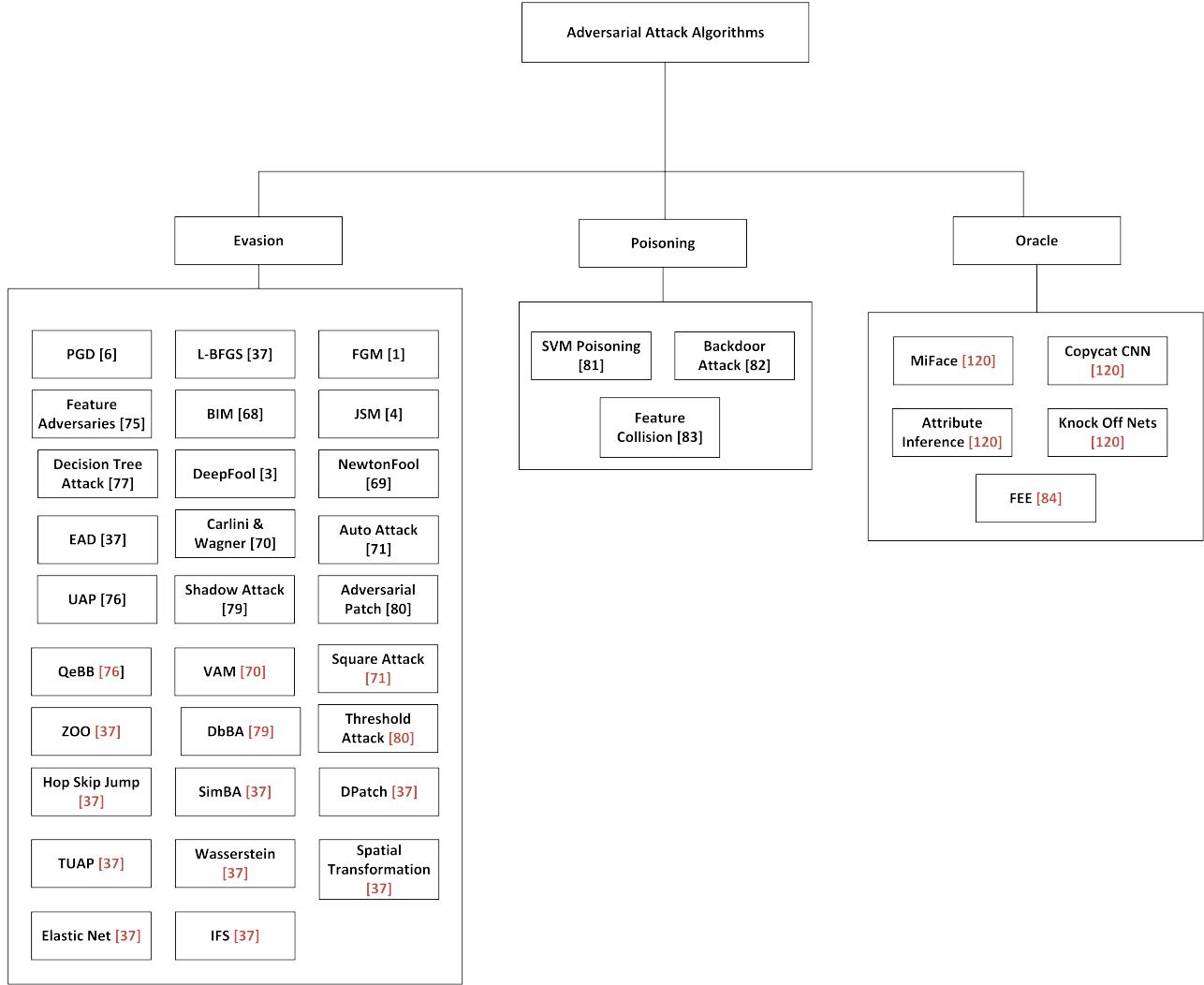| | |
|---|---|
| MiFace [120] | Copycat CNN [120] |
| Attribute Inference [120] | Knock Off Nets [120] |
| FEE [84] | |

Fig. 8.   Adversarial Attack Algorithms

posed 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 $\ell_\infty$-ball, PGD iterate to explore the maximum loss.

Croce et al. [80] 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 [81]. In [80], 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 [82] and Square attack [83] 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. [84] 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 [85] 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. [73] 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. [86] 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 $L_i nfinty$ 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. [89] 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.

*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. [90] 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. [91] 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. [92] 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.

*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" [70]. 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. [72] 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. [72] proposed the attribute inference attack which could be launched either as a white-box or black-box attack.
- Extraction Attacks: Correia-Silva et al. [93] 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. [94] 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 [71] 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.

## VI. 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 [97]. 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 Fig. 9.

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 Fig. 9. 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 Fig. 9. 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.
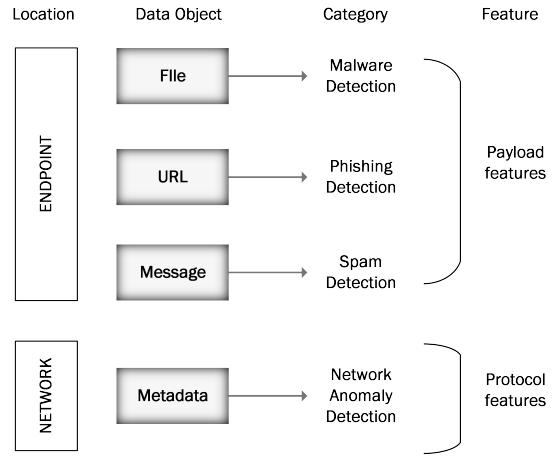


Fig. 9. Adversarial attack classification

### A. 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 [98]. 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.

In contrast to adversarial attacks in the field of image processing or computer vision, malware classification adversarial learning is more challenging. This occurs because even very slight modifications to malware bytes of the binary files can significantly alter the functionality of the malware. In computer vision, the addition of tiny perturbations to an image sample does not alter human perception of the image and same as in speech processing. Text processing and malware detection are similar in this regard since a very

14

slight change in the input such as a word or a byte will alter the meaning of the text or the functionality of the malware. Hence, approaches for generating adversarial samples in the domain of machine learning based malware detection needs to occur in such a way that the malicious functionality of the malware file is not distorted. Several approaches for achieving this adversarial malware attacks have been research and are discussed in the sections below. 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.

*1) Iagodroid:* One of the earliest attacks against machine learning based malware detection systems was the Iagodroid attack [99]. Iagodroid uses a method to induce mislabelling of malware families during the triaging process of malware samples.

*2) Stingray:* Suciu et al [100] 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.

*3) Texture Perturbation Attacks:* Researchers have deployed visualization techniques similar to computer vision and adapted it for malware classification [101]. 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 [12]. The attack model for ATMPA works by allowing the attacker to distort the malware image data during the visualization process.

*4) Android malware attack in Problem space:* [67] et al formalized an approach for problem space adversarial evasion attacks against machine learning based android malware attacks. 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.

*5) EvnAttack:* EvnAttack is an evasion attack model that was proposed in [58] 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.

*6) AdvAttack:* AdvAttack was proposed in [56] 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.

*7) 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.

*8) Black-Box Attacks against RNN Based Malware Detection Algorithms:* Hu et al. [103] 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.

*9) 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.

*10) 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.

*11) 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.

*12) 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.

*B. Adversarial attacks on Spam Detection*

*1) 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 [108]. A variety of good word attacks introduced by Lowd [109] were successfully evading the machine learning models from detecting spam or junk emails.

*2) 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 [110]. However, malicious crowdsourcing detection systems are highly vulnerable to adversarial evasion and poisoning attacks.

*3) Attacks Against ML for Keystroke Dynamics:* Negi et al. [111] created adversarial keystroke samples that misled an otherwise accurate classifier into accepting the artificially generated keystroke samples as belonging to an authentic user.

*4) Attacks against ML for credit card fraud detection:* Zeager et al. [112] 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.

*C. Adversarial attacks against Phishing Detection*

*1) Addressing Adversarial Attacks Against Security Systems Based on Machine Learning:* Apruzzese et al. [113] 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. They also demonstrated that adversarial training was effective in improving the robustness of deep learning based network intrusion detection systems.

*D. Adversarial attacks against Network Anomaly Detection*

*1) IDSGAN:* IDSGAN was proposed by Lin et al. [114] for generating adversarial attacks targeted towards intrusion detection systems. IDSGAN is based on the Wasserstein GAN [115] 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.

*2) TCP Obfuscation Techniques:* Another method for evading machine learning based intrusion detection systems is the use of obfuscation techniques. Homolial et al. [116] proposed the modification of various properties of network connections to obfuscate a TCP communication which successfully evades a wide variety of intrusion detection classifiers.

*3) Deep Adversarial Learning in Intrusion Detection: A Data Augmentation Enhanced Framework:* Zhang el al. [117] 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.

*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. [118]. 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 addition, a GAN based defense was also proposed to improve the robustness of the IDS against adversarial attacks. The training method is adversarial training, which injects adversarial examples into the training data which helps the deep learning model to learn the possible adversarial perturbations. Unlike similar adversarial training methods in which the model becomes robust to only the adversarial samples which it was trained against, making it only as effective as signature based IDS or anomaly detection models, the approach used in this paper is more effective. Since a generative model is included in the deep learning based IDS pipeline, the adversarial training captures the adversarial samples that are generated by the generative

model. Therefore, both known and unknown adversarial perturbations are identified and mitigated against.

*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.

*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 [119]. 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.

*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. [120]. 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, [121], wireless resource allocation [122] [123] and radio signal (modulation) classification [124]. Uses of Machine Learning in IoT include anomaly detection [125], device identification [126] [127], and signal authentication [128].

*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 [129] 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.

*9) Adversarial Deep Learning for Cognitive Radio Security: Jamming Attack and Defense Strategies:* Shi et al. [130] 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. 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.

*10) Performance Evaluation of Physical Attacks against E2E Autoencoder over Rayleigh Fading Channel:* Albaseer et. al [131] 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.

*11) Physical adversarial attacks against end-to-end autoencoder communication systems:* Sadeghi et al. [132] 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. [131], their study demonstrates that adversarial attacks are more destructive than jamming attacks.

*12) Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Detection Models:* Stokes et. al [133] 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.

*13) Targeted Adversarial Examples Against RF Deep Classifiers:* Kokalj-Filipovic et al. [134] 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.

*14) Adversarial classification:* Dalvi et al [135] 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 [109] had considered adversarial attacks against spam detectors albeit not machine learning based.

*15) Slack Attacks:* A byte-based convolutional neural network (MalConv) was introduced by Raff et al. [136]. Unlike image perturbation attacks [55], 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 [59]. 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. [137] 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. [137] 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.

*16) Deep Learning-Based Intrusion Detection With Adversaries:* Wang et al. [11] 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.

*17) Evaluating Deep Learning-Based Network Intrusion Detection System in Adversarial Environment:* Peng et al. [138] 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).

*18) Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks:* Ibitoye et al. [139] studied the adversarial samples effectiveness against deep learning-based Intrusion Detection System (IDS) within the context of an IoT network. The authors provide a com-

prehensive 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).

*19) Online anomaly detection under adversarial impact:* Kloft et al. [140] 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.

*20) Security evaluation of pattern classifiers under attack:* Biggio et al. [141] 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.

## VII. 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 [142]. We present in this paper, an adversarial risk grid map shown in Figure 10 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.

### A. 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 [142], frameworks for adversarial risk based on the concept of obscurity have been proposed [143].

### B. Adversarial Risk Grid Map

A modified notion of adversarial risk was proposed in [144] which suggested that certain classifiers inherently have low adversarial risk. Other works [145] [146] 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.

- *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.

### C. 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.

## VIII. Defending 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. [75] classified the defenses into two broad strategies. Proactive strategies and reactive strategies.
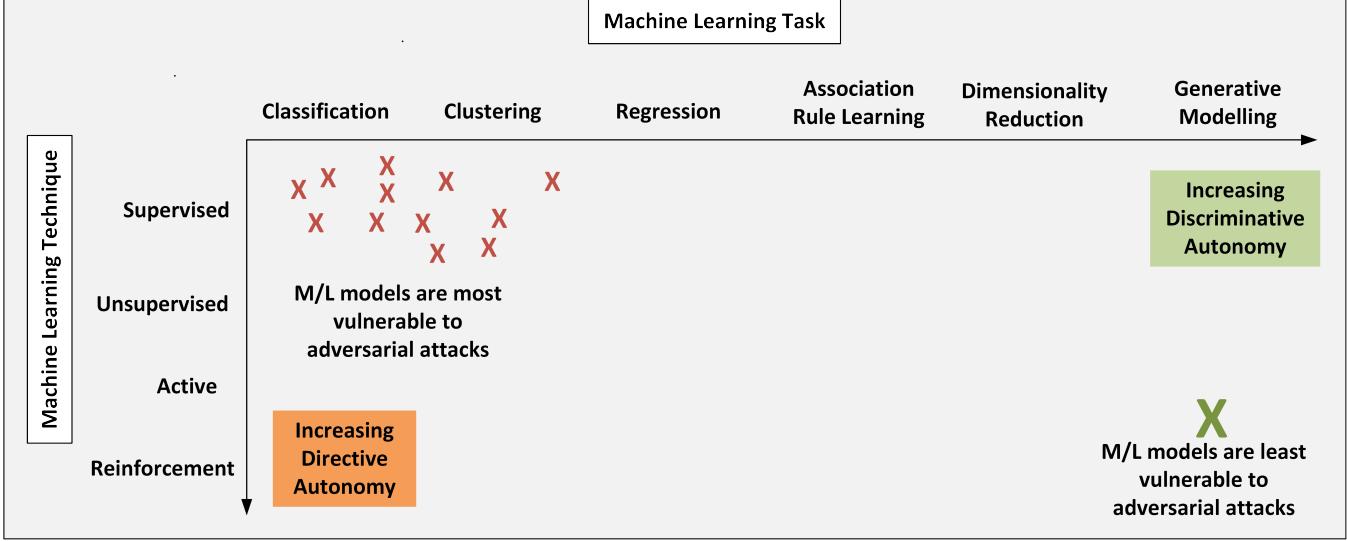
Fig. 10.  Adversarial Risk Grid Map

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 [147] . These optimization algorithms often result in high frequency outputs [148]. This however makes the defense methods against this adversarial samples vulnerable to adversarial samples that are generated within a low frequency subspace.

Notably, the defense against adversarial samples can be classified based on the attackers strategy. For training attacks, the three approaches for defense are data encryption, data sanitization and statistical robustness. For attacks at test time, the three approaches for difference are differential privacy, homomorphic encryption and adversarial training robustness. We illustrate these various approaches in Fig. 11 below.

In this section, we provide the most common attack methods in use today and classify them based on the strategy and approach.

*1) Gradient masking:* Since most of 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 [149] 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[149]. However, [150] indicate that gradient masking react as overfitting in their experiments.

*2) Defensive Distillation:* Distillation technique was originally proposed by Hinton et al. [151] 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 proofed 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 have large network and large dataset. Therefore, they use learning specialist models that each use subset of dataset classes to reduce amount computation [151]. Also, it was adapted by Papernot et al. [79] 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.

*3) Adversarial Training:* Adversarial training [6] is a method which 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. [55] 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 solve 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 result to maximize the model loss. Where, the outer minimization

intents to minimize the loss by finding model parameters that build more robust model with less adversarial loss [6]. Adversarial training improves the classification performance of the machine learning model and makes it more resilient to adversarial crafting.

However, adversarial training has certain limitation 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 [132].

*4) 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 [128] works on the premise that adversarial samples have a higher uncertainty that 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 [152] as well as the use of an auxiliary network of the original network introduced by Metzen et al. in [153].

*5) Feature Reduction:* Other potential defenses for adversarial attacks have been proposed. Simple feature reduction was evaluated by Grosse et al. [154] but was found inadequate in defending against adversarial attacks.

*6) 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 [155] to combine adversarial detecting techniques with one or more other methods for creating a more robust defense against adversarial attacks.

## IX. DISCUSSION AND LESSONS LEARNT

**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 Fig. 10, 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
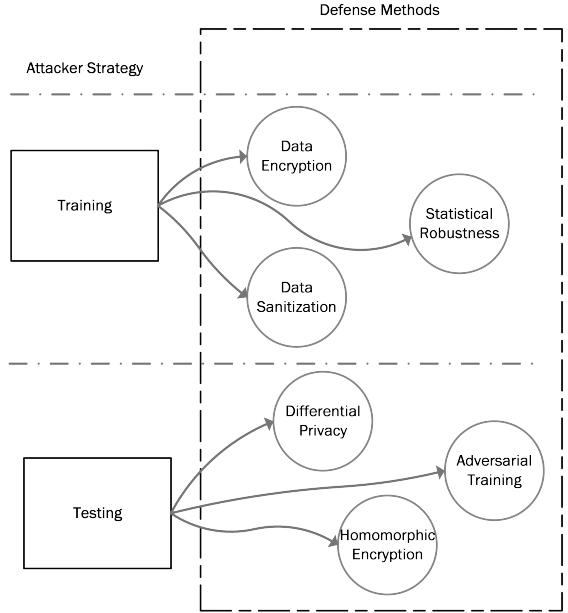


Fig. 11. Adversarial Defense Methods based on Attacker Strategy

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.

**Transferability with regards to machine learning technique**: Transferability of adversarial samples [73] [156] has been shown to be more effective with targeted adversarial samples [74]. 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. [148] 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 [157]. Furthermore, our findings indicate that defenses against adversarial attacks are specific to a particular type of attack and are not necessarily transferable. Recent research [73] have studied the transferability in malware machine learning models in machine learning applications such as malware detection.

**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.

**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 misclassification for the 5 categories we observed i.e malware classification, malware detection, intrusion detection, spam detection, phishing detection. The greatest adversarial effect was observed in the malware classification domain, in which adversarial samples reduced the accuracy of a deep learning based malware classifier from 97% to 5%. The challenge of quantifying the efficiency of adversarial example generation, is an emerging field and several approaches have been proposed in recent literature. In [158] 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.

**How does Adversarial attack in network security differ from computer vision**

1) 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.

2) 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 of this is that it becomes more difficult to defend against adversarial attacks in network security due to the heterogeneity , hence, transferability and universal defenses against . It should be noted that significant strides have been made in computer vision, with regards to developing universal defences but this is still an infant research area in network security. Also, the feature varies greatly based on the network security application. In most cases, the features used in the machine learning classification are also the features that are perturbed in generating the adversarial samples.

## X. CONCLUSION AND FUTURE WORK

We present a first of its kind survey on adversarial attacks on machine learning in network security. The previous survey [13] 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 [157]. 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, adversarial attack has mostly been carried out on data at rest, with very few successful attempts of adversarial attacks on data in transit, or streaming data such as [159] [160]. However, in network security domains such as in the field of intrusion detection, realistic adversarial attacks will be carried out on data in transit. Hence, more research is needed in this area to

understand the potential risks of adversarial attacks against data in transit and the possible defense techniques.

Adversarial attack against federated learning [161] is still an open area of research. Federated learning [162], which is quite different from distributed computation, involves the situation in which each client performs the machine learning computation without sending the data to the cloud. As such, the cloud provider does not have a complete view of the machine learning model with significant gains for privacy and confidentiality.

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 fig 10 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[163] [164] [165]. 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 [166]. 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

### References

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

[2] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 36–42, IEEE, 2018.

[3] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

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

[5] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, 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 preprint arXiv:1706.06083*, 2017.

[7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519, ACM, 2017.

[8] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pp. 16–25, ACM, 2006.

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

[10] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–169, 2018.

[11] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.

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

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

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

[15] 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. 12103–12117, 2018.

[16] V. Duddu, "A survey of adversarial machine learning in cyber warfare," *Defence Science Journal*, vol. 68, no. 4, pp. 356–366, 2018.

[17] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, 2019.

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

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

[20] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 59, 2016.

[21] H. X. Y. M. Hao-Chen, L. D. Deb, H. L. J.-L. T. Anil, and K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

[22] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 3, pp. 1–41, 2020.

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

[24] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear estimation and classification*, pp. 149–171, Springer, 2003.

[25] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.

[26] E. Alpaydin, *Introduction to machine learning*. MIT press, 2009.

[27] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.

[28] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*, pp. 839–846, Citeseer, 2000.

[29] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[30] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[31] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[32] L. Deng, D. Yu, *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[33] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[34] Y. G. Şahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," 2011.

[35] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 435–442, ACM, 2010.

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

[37] A. Girma, M. Garuba, and R. Goel, "Advanced machine language approach to detect ddos attack using dbscan clustering technology with entropy," in *Information Technology-New Generations*, pp. 125–131, Springer, 2018.

[38] M. Zamani and M. Movahedi, "Machine learning techniques for intrusion detection," *arXiv preprint arXiv:1312.2177*, 2013.

[39] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Australasian Joint Conference on Artificial Intelligence*, pp. 137–149, Springer, 2016.

[40] V. Ford and A. Siraj, "Applications of machine learning in cyber security," in *Proceedings of the 27th International Conference on Computer Applications in Industry and Engineering*, 2014.

[41] J. Singh and M. J. Nene, "A survey on machine learning techniques for intrusion detection systems," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 4349–4355, 2013.

[42] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on*, pp. 000277–000282, IEEE, 2017.

[43] 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 *International Conference on Extending Database Technology (EDBT), March 26-29, 2018*, ACM, 2018.

[44] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in *International Conference on Image Analysis and Processing*, pp. 50–57, Springer, 2005.

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

[46] 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 Generation Computer Systems*, vol. 79, pp. 431–446, 2018.

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

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

[49] Y. Dong, Y. Zhang, H. Ma, Q. Wu, Q. Liu, K. Wang, and W. Wang, "An adaptive system for detecting malicious queries in web attacks," *Science China Information Sciences*, vol. 61, no. 3, p. 032114, 2018.

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

[51] 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," 2007.

[52] E. Bursztein, M. Martin, and J. Mitchell, "Text-based captcha strengths and weaknesses," in *Proceedings of the 18th ACM conference on Computer and communications security*, pp. 125–138, ACM, 2011.

[53] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *Networking, sensing and control, 2004 IEEE international conference on*, vol. 2, pp. 749–754, IEEE, 2004.

[54] 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 Cryptology ePrint Archive*, vol. 2018, p. 53, 2018.

[55] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[56] L. Chen and Y. Ye, "Secmd: Make machine learning more secure against adversarial malware attacks," in *Australasian Joint Conference on Artificial Intelligence*, pp. 76–89, Springer, 2017.

[57] L. Chen, S. Hou, Y. Ye, and S. Xu, "Droideye: Fortifying security of learning-based classifier against adversarial android malware attacks," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 782–789, IEEE, 2018.

[58] L. Chen, Y. Ye, and T. Bourlai, "Adversarial machine learning in malware detection: Arms race between evasion attack and defense," in *Intelligence and Security Informatics Conference (EISIC), 2017 European*, pp. 99–106, IEEE, 2017.

[59] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," *arXiv preprint arXiv:1803.04173*, 2018.

[60] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 27–38, 2017.

[61] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402, Springer, 2013.

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

[63] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations*, 2018.

[64] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43–58, ACM, 2011.

[65] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

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

[67] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1332–1349, IEEE, 2020.

[68] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, "A taxonomy and terminology of adversarial machine learning," 2019.

[69] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2040–2042, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

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

[71] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

[72] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, 2015.

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

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

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

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

[77] U. Jang, X. Wu, and S. Jha, "Objective metrics and gradient descent algorithms for adversarial examples in machine learning," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 262–277, 2017.

[78] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, 2017.

[79] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, IEEE, 2016.

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

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

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

[83] F. Croce, M. Andriushchenko, and M. Hein, "Provable robustness of relu networks via maximization of linear regions," in *the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2057–2066, 2019.

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

[85] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

[86] 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 *Thirty-second AAAI conference on artificial intelligence*, 2018.

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

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

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

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

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

[92] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Advances in Neural Information Processing Systems*, pp. 6103–6113, 2018.

[93] 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 *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[94] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4954–4963, 2019.

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

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

[97] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial learning in the cyber security domain," *arXiv preprint arXiv:2007.02407*, 2020.

[98] "Total malware."

[99] A. Calleja, A. Martín, H. D. Menéndez, J. Tapiador, and D. Clark, "Picking on the family: Disrupting android malware triage by forc-

[100] 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 *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 1299–1316, 2018.

[101] K. S. Han, J. H. Lim, B. Kang, and E. G. Im, "Malware analysis using visualized images and entropy graphs," *International Journal of Information Security*, vol. 14, no. 1, pp. 1–14, 2015.

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

[103] W. Hu and Y. Tan, "Black-box attacks against rnn based malware detection algorithms," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[104] A. Al-Dujaili *et al.*, "Adversarial deep learning for robust detection of binary encoded malware," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 76–82, IEEE, 2018.

[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 preprint arXiv:1802.04528*, 2018.

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

[107] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole exe," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[108] 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*, vol. 62, pp. 98–105, Madison, Wisconsin, 1998.

[109] D. Lowd and C. Meek, "Good word attacks on statistical spam filters.," in *CEAS*, vol. 2005, 2005.

[110] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 239–254, 2014.

[111] P. Negi, A. Sharma, and C. Robustness, "Adversarial machine learning against keystroke dynamics," 2017.

[112] M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, "Adversarial learning in credit card fraud detection," in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 112–116, IEEE, 2017.

[113] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *2019 11th International Conference on Cyber Conflict (CyCon)*, vol. 900, pp. 1–18, IEEE, 2019.

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

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

[116] 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 preprint arXiv:1805.02684*, 2018.

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

[118] M. Usama, M. Asim, S. Latif, J. Qadir, *et al.*, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 78–83, IEEE, 2019.

[119] J. Aiken and S. Scott-Hayward, "Investigating adversarial attacks against network intrusion detection systems in sdns," in *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–7, IEEE, 2019.

[120] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Iot network security from the perspective of adversarial deep learning," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, IEEE, 2019.

ing misclassification," *Expert Systems with Applications*, vol. 95, pp. 113–126, 2018.

[121] F. Liang, C. Shen, and F. Wu, "An iterative bp-cnn architecture for channel decoding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 144–159, 2018.

[122] 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 *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6, IEEE, 2017.

[123] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, pp. 213–226, Springer, 2016.

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

[125] J. Canedo and A. Skjellum, "Using machine learning to secure iot systems," in *2016 14th annual conference on privacy, security and trust (PST)*, pp. 219–222, IEEE, 2016.

[126] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "Profiliot: a machine learning approach for iot device identification based on network traffic analysis," in *Proceedings of the symposium on applied computing*, pp. 506–509, 2017.

[127] 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 *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2177–2184, IEEE, 2017.

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

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

[130] 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 *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2018.

[131] A. Albaseer, B. S. Ciftler, and M. M. Abdallah, "Performance evaluation of physical attacks against e2e autoencoder over rayleigh fading channel," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 177–182, IEEE, 2020.

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

[133] 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 *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pp. 1–8, IEEE, 2018.

[134] S. Kokalj-Filipovic, R. Miller, and J. Morman, "Targeted adversarial examples against rf deep classifiers," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, pp. 6–11, 2019.

[135] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108, 2004.

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

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

[138] Y. Peng, J. Su, X. Shi, and B. Zhao, "Evaluating deep learning based network intrusion detection system in adversarial environment," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 61–66, IEEE, 2019.

[139] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.

[140] M. Kloft and P. Laskov, "Online anomaly detection under adversarial impact," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 405–412, 2010.

[141] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 4, pp. 984–996, 2013.

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

[143] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.

[144] A. S. Suggala, A. Prasad, V. Nagarajan, and P. Ravikumar, "Revisiting adversarial risk," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2331–2339, 2019.

[145] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," in *Advances in Neural Information Processing Systems*, pp. 1178–1187, 2018.

[146] 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 preprint arXiv:1805.12152*, vol. 2, no. 3, 2018.

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

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

[149] A. Nayebi and S. Ganguli, "Biologically inspired protection of deep networks from adversarial attacks," *arXiv preprint arXiv:1703.09202*, 2017.

[150] Y. Yanagita and M. Yamamura, "Gradient masking is a type of overfitting," *International Journal of Machine Learning and Computing*, vol. 8, no. 3, 2018.

[151] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[152] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 135–147, ACM, 2017.

[153] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[154] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[155] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv preprint arXiv:1710.10766*, 2017.

[156] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[157] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Advances in Neural Information Processing Systems*, pp. 5014–5026, 2018.

[158] H. Li, S. Zhou, W. Yuan, J. Li, and H. Leung, "Adversarial-example attacks toward android malware detection system," *IEEE Systems Journal*, vol. 14, no. 1, pp. 653–656, 2019.

[159] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1738–1742, IEEE, 2020.

[160] Y. Gong, B. Li, C. Poellabauer, and Y. Shi, "Real-time adversarial attacks," *arXiv preprint arXiv:1905.13399*, 2019.

[161] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, 2020.

[162] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[163] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning," in *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–8, IEEE, 2018.

[164] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: Deep neural networks over encrypted data," *arXiv preprint arXiv:1711.05189*, 2017.

[165] 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 Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[166] 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*, pp. 689–696, 2005.