applied sciences

MDPI

Review

Better Safe Than Never: A Survey on Adversarial Machine Learning Applications towards IoT Environment

Sarah Alkadi , Saad Al-Ahmadi and Mohamed Maher Ben Ismail *

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11362, Saudi Arabia; sara.alqadi@gmail.com (S.A.); salahmadi@ksu.edu.sa (S.A.-A.)
* Correspondence: mbenismail@ksu.edu.sa

Abstract: Internet of Things (IoT) technologies serve as a backbone of cutting-edge intelligent systems. Machine Learning (ML) paradigms have been adopted within IoT environments to exploit their capabilities to mine complex patterns. Despite the reported promising results, ML-based solutions exhibit several security vulnerabilities and threats. Specifically, Adversarial Machine Learning (AML) attacks can drastically impact the performance of ML models. It also represents a promising research field that typically promotes novel techniques to generate and/or defend against Adversarial Examples (AE) attacks. In this work, a comprehensive survey on AML attack and defense techniques is conducted for the years 2018–2022. The article investigates the employment of AML techniques to enhance intrusion detection performance within the IoT context. Additionally, it depicts relevant challenges that researchers aim to overcome to implement proper IoT-based security solutions. Thus, this survey aims to contribute to the literature by investigating the application of AML concepts within the IoT context. An extensive review of the current research trends of AML within IoT networks is presented. A conclusion is reached where several findings are reported including a shortage of defense mechanisms investigations, a lack of tailored IoT-based solutions, and the applicability of the existing mechanisms in both attack and defense scenarios.

Keywords: Internet of Things (IoT); Cybersecurity; intrusion detection; adversarial machine learning (AML)

check for updates

Citation: Alkadi, S.; Al-Ahmadi, S.; Ismail, M.M.B. Better Safe Than Never: A Survey on Adversarial Machine Learning Applications towards IoT Environment. *Appl. Sci.* **2023**, *13*, 6001. https://doi.org/10.3390/app13106001

Academic Editor: Luis Javier Garcia Villalba

Received: 30 March 2023
Revised: 30 April 2023
Accepted: 11 May 2023
Published: 13 May 2023

1. Introduction

Cybersecurity is growingly considered a major concern for different computer applications. It needs to be noticed for all types of network traffic to ensure that any potential security issues are noticed and detected such as any kind of intrusions and attacks. Within IoT, the constraints of resources can affect the efficiency of the well-known security solutions making them luring targets for cyber-attacks. This reflects the necessity of tailored solutions to address the related-security issues in IoT networks.

Recently, Machine Learning techniques have been increasingly adopted within several research fields. In particular, they have been integrated into IoT frameworks in order to enhance security and reinforce privacy. ML techniques have been gaining the upper hand because they relaxed the human intervention constraint through sophisticated algorithms that support decision-making tasks in a timely manner. In fact, ML-based models can analyze the traffic passed through devices and detect abnormal behaviors given the resource-constraint characteristic and the three main layers of IoT systems. ML techniques can be categorized into: (i) Supervised learning, (ii) Unsupervised learning, and (iii) Reinforcement learning approaches. In addition, they can also be grouped based on their architecture into (i) Traditional shallow learning and (ii) Deep learning techniques [1,2]. One should note that the direct application of traditional ML techniques proved to be inefficient in handling data in an IoT environment [3,4]. This can be inferred from several reasons related to ML algorithms in terms of complexity, scalability, real-time processing, data dimensionality, and data distribution. Firstly, ML algorithms introduce some complex challenges because of memory, computational complexity, and diversity of data types. Secondly, they are to some extent unable to provide scalable solutions, specifically for IoT devices due to energy constraints. Thirdly, ML algorithms are not designed to process large streams of data in real time. In other words, they typically assume training the entire data collection which is not fit for the IoT environment. Fourthly, their performance is considerably affected by the curse of dimensionality usually associated with real-world data. Lastly, the use of data out of the underlying data distribution is used for ML model training and testing in which attackers can craft adversarial examples and cause performance degradation [3,4].

Although there is active research work to manage the aforementioned limitations, there is limited work to study the effect of AML on ML-based IoT solutions. These solutions' limitations are usually characterized by classification and detection issues where ML models are vulnerable to adversarial samples. The adversary can inject adversarial samples to affect the model decision boundary and therefore misclassify the analyzed data [3,5].

Researchers have reported the potential vulnerability of ML models to adversarial attacks [1,3,4]. This has promoted researcher efforts related to Adversarial Machine Learning (AML) challenges. In fact, AML mainly concerns the intersection of computer security and machine learning fields. Specifically, it works on analyzing the attacks that aim to degrade the performance of ML-based models. It also investigates the process of generating and detecting the crafted adversarial examples and how eventually incorporate possible defensive mechanisms. The area has been extensively associated with applications dealing with images as primary data modality. On the other hand, it is still growing within the fields of network traffic analysis and IoT [6,7].

There are several elements that reflect the crucial need to conduct the proposed survey. Those elements stem from the sacristy of studying the effect of AML attacks and defenses in the IoT context. As such, this survey focuses on investigating the current advances and trends of AML applied to the IoT-based IDS domain. It considers the unique characteristics of IoT and their challenges when adopting ML-based intrusion detection solutions. It identifies the recent and important methods for both crafting and defending adversarial examples. Accordingly, the importance of this survey comes from its objectives of presenting a comprehensive review of the AML attacks and defenses against the IoT environment in the context of the ML-based intrusion detection domain.

Contribution & Structure

This article investigates the application of AML concepts within the IoT context. It surveys the current trends of AML within the IoT environment considering the most reputable science databases such as Springer [8], IEEE Xplore [9], arXiv [10], ScienceDirect [11], and Research Gate [12]. The framework adopted for this survey is given in Figure 1. Network Intrusion Detection Systems (NIDS) require real-time analysis of sensitive data which reflects the robustness and security needs. Only fifteen papers address those topics. This indicates the sacristy of research studies toward the intersection between those two fields. The analysis of these works reveals the effectiveness of AML crafting techniques in enhancing ML-based NIDS models. Moreover, the defense mechanisms have not been explored sufficiently with a considerable lack of benchmark datasets. The structure of this manuscript is illustrated in Figure 2. As seen, it includes the following: Section 2 introduces the main specifications and security challenges found in IoT networks.

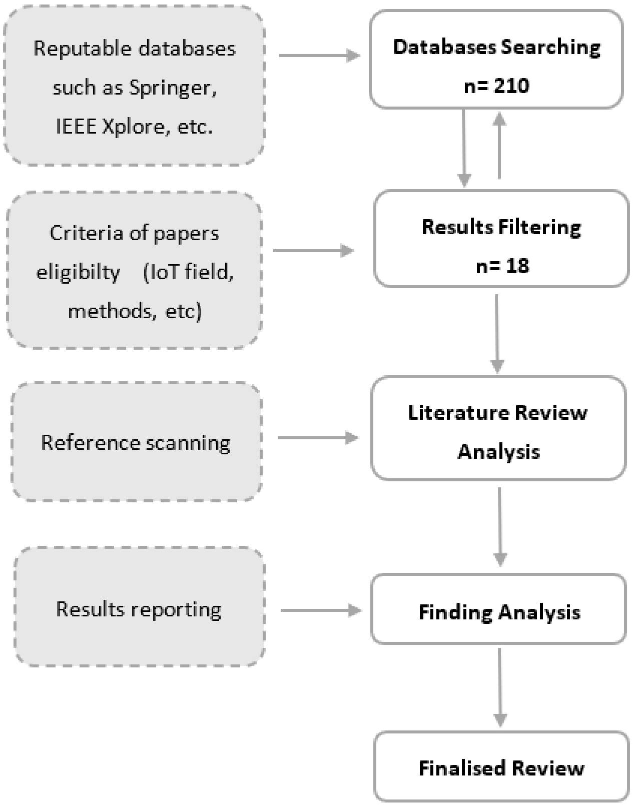
Reputable databases such as Springer, IEEE Xplore, etc.
Databases Searching n= 210
Criteria of papers eligibilty (IoT field, methods, etc)
Results Filtering n= 18
Reference scanning
Literature Review Analysis
Results reporting
Finding Analysis
Finalised Review

Figure 1. The research framework for the proposed survey.

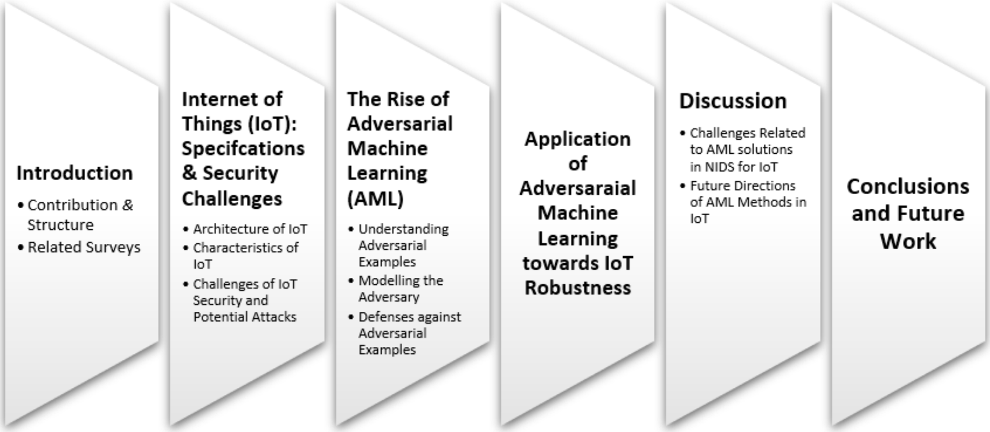
Introduction
• Contribution & Structure
• Related Surveys
Internet of Things (IoT): Specifcations & Security Challenges
• Architecture of IoT
• Characteristics of IoT
• Challenges of IoT Security and Potential Attacks
The Rise of Adversarial Machine Learning (AML)
• Understanding Adversarial Examples
• Modelling the Adversary
• Defenses against Adversarial Examples
Application of Adversaraial Machine Learning towards IoT Robustness
Discussion
• Challenges Related to AML solutions in NIDS for IoT
• Future Directions of AML Methods in IoT
Conclusions and Future Work

Figure 2. The structure of the survey.

The adversarial machine learning area, including various aspects of causes and characteristics, crafting methods, threat models, defense mechanisms, and evaluation metrics, is investigated in Section 3. In Section 4, we survey the state-of-the-art research that applies adversarial machine learning techniques to enhance IoT robustness. We then discuss the surveyed papers and present the main findings of this research in Section 5. Finally, the conclusions and future work are presented in Section 6.

2. Internet of Things (IoT): Specifications & Security Challenges

2.1. Architecture of IoT

The features of IoT platforms have attracted great attention from research and industrial communities for advancing people's daily life. Several architectures of IoT have been proposed as seen in Figure 3. The three-layer model [13,14] which consists of three layers where groups of networks standards, technologies, and services are placed [13,15,16] was initially proposed. The latter three layers are detailed below:

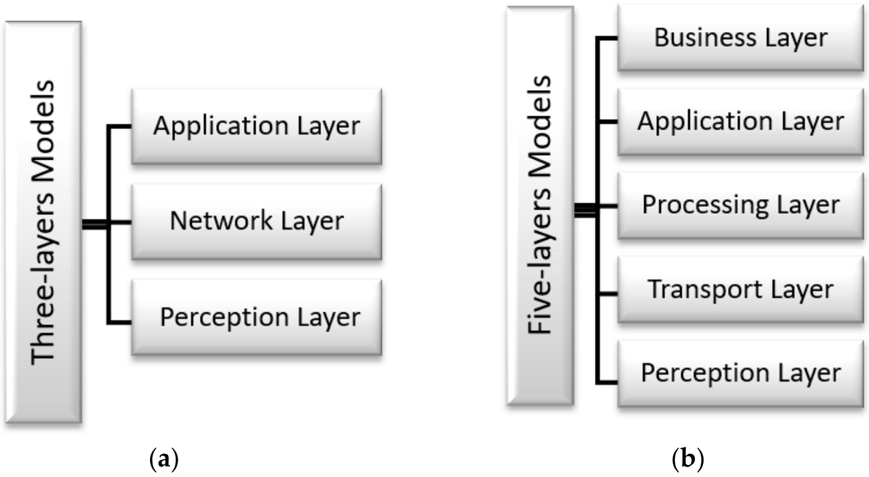

Figure 3. Typical IoT architecture. (**a**) shows three-layer models of IoT architecture and (**b**) shows five-layer models of IoT architecture.

2.1.1. Perception Layer

It represents the first layer of IoT architecture that is used for indicating the objects at the physical level. The objects work within the adjacent environment by sensing, collecting, and processing information. It is defined as smart objects with the following common characteristics:

- Communication: objects have connectivity to the internet and to other objects to collect data, update status and collaboratively provide proper services.
- Identification: objects have unique identification and can be located based on their physical location.

Further characteristics that can be also added to IoT-application are outlined below:

- Addressability: objects can be configured either directly or remotely.
- Processing capabilities: objects have embedded processing capabilities to handle shared information by the sensors and direct the actuators.
- User interface: objects have an appropriate interface for easing the user's experience.

One should note that multiple protocols are being currently adopted. Namely, RFID, Bluetooth, IEEE 802.11ah, IEEE 802.15.4e, and Z-Wave are widely used [17–19].

2.1.2. Network Layer

This layer is located between the perception and application layers as a mediator to transmit the data and information. This layer works on transmitting and processing the network traffic and data alongside connecting IoT infrastructure components including smart objects, network devices, and servers. Multiple protocols are adopted such as Wi-Fi, IPv6, GSM, and others for the purpose of linking devices with smart services. Moreover, specific protocols are created and used such as the 6LoWPAN protocol due to the low computational power of IoT devices. The criteria for selecting protocols for specific applications can be summarized as follows: the network's capacity, the node's computational power, and the required transmission speed.

The wireless protocols are more adopted since the wireless sensor network has unique features that ease its installation and expansion in different environments. In the case of wired networks, it offers better transmission rates with more reliable connections which is recommended to be used in crucial environments of IoT [17–19].

2.1.3. Web/Application Layer

This layer is the third layer in IoT architecture where services are provided to the end users via proper software that hides the heterogeneity of the underlying layers. All the data are going through several pre-processing steps including storing, aggregation, and analysis. Thus, the data are used by IoT software in various aspects including but not limited to transportation, health, education, business, logistics, home, and many others. The adopted technologies vary from cloud computing which works on data provided by previous layers of resources remotely to edge computing in which the performance of IoT networks is enhanced through distributing the workflow between end nodes. The data are also managed in this layer considering their different formats [17–19].

Although the three-layer model represents the main characteristics of IoT, it lacks important aspects for tackling IoT research issues. As such, the five-layer model is proposed that incorporates two more layers namely, the processing and business layers alongside perception, transport, and application layers. The perception and application layers apply their same functionalities in the three-layer model. For the other different layers, it starts with the transport layer which is located between the perception and processing layers to handle the transmission of the sensor data in-between by networks such as wireless, LAN, and Bluetooth. It is followed by the processing layer where data analytics techniques are applied to substantial amounts of data received from the previous layer including storing, analyzing, and processing capabilities. These capabilities require employing several technologies such as big data analytics modules. The final layer is the business layer that represents the management console where all the components of IoT such as applications, business, and users' privacy are controlled.

2.2. Characteristics of IoT

IoT has multiple features and characteristics that make it distinguished and can be summarized as follows [3,14]:

- Heterogeneity: IoT networks contain several types of devices that work together to form a reliable communication channel. This means different technologies, protocols, paradigms, and capabilities are used based on constraints related to the computational power of the hardware. Such technologies include wireless sensor networks (WSN), radio-frequency identification (RFID), near-field communication (NFC), and others.
- Large-Scale Architecture: A massive number of IoT network devices are connecting at a large scale level which leads to constraints on communication capabilities. Multiple challenges are introduced with regard to this matter in terms of design, storage, speed, efficiency, accessibility, and security of IoT networks. It requires standardized technologies to enhance performance and ensure proper scalability.
- Power and Cost Constraints: Due to the huge increasing number of IoT-connected devices, low-power, and low-cost solutions are used to adapt the complexity of these networks and smooth their workflow.
- Interconnectivity: Connections between IoT devices are used to conduct global and local information at various times and from any place. The type of connectivity can be determined based on the IoT-provided services. Local connections take place in services such as autonomous vehicles while the global ones can be seen in smart home services where access requires management of critical infrastructure.
- Close Proximity: IoT networks have used close proximity where dedicated short-range communication instead of using network-centric communications as in the traditional Internet. This minimizes the use of central authority through key enabling technologies of IoT such as Device-to-Device communication (D2D) and Machine-to-Machine communication (M2M).
- Reliability and Latency: IoT networks have supported the workflow of critical services using Ultra-Reliable and Low Latency communication. This can help these services such as robotic surgery, intelligent transportation system, and others by ensuring strict criteria in terms of delay and reliability of IoT network performance.

- Autonomic Computing: IoT networks are considered an autonomic computing system that includes self-configuration, self-optimization, self-healing, and self-protection properties. Their properties contribute by allowing automatic configuration, automatic performance boosting, automatic error detection, and automatic defense mechanism, respectively. These properties support the operation of IoT systems in emergency and disaster situations [20].
- Intelligence: Smart services are incorporated with IoT networks where a decision is made in a timely manner. This can be achieved by performing analysis and processing on the massive amount of IoT-generated data followed by taking proper actions without human intervention.

Additionally, there are multiple areas where IoT applications are adopted such as smart environment, smart agriculture, smart transport, smart health, smart energy, defense manufacturing, and industrial engineering [21].

2.3. Challenges of IoT Security and Potential Attacks

The original design of the Internet was not considering the emergence of modern technologies which in turn make the deployment of IoT hard on top of the existing networks and security solutions. Moreover, the underlying infrastructure of the Internet has several limitations in terms of scalability, complexity, configuration, and resource constraint. A lot of applications are presented by IoT networks for a vast number of end-users making security and privacy more challenging topics in such an environment. The limitation of the computational power of IoT devices complicates the process of providing efficient security mechanisms. Moreover, the variety of IoT devices increases the attack surface in which they are being targeted by different attacks [22,23]. According to the aforementioned IoT architecture layers, three possible attack surfaces can be summarized as follows [24]:

- Perception Surface. In this direct surface, physical devices are found where attacks are conducted on units such as sensors, actuators, microcontrollers, RFID readers, and others. Identification, communication, and collection of information are performed by these devices making them targeted by several physical and logical attacks including vandalism, Denial of Service (DoS), eavesdropping, jamming, and others.
- Network Surface. Wired and wireless sensor networks are used to connect IoT devices which reflects the necessity of integrating them at a large scale. Due to that large-scale topology, IoT networks' surface is exposed to attacks while the data is transmitted using low-efficient security protocols [25]. Several attack scenarios can be seen such as scanning open ports to access the victim's networks and steal sensitive information. The attack types include man in the middle, spoofing, DoS, traffic analysis, jamming, and others.
- Application/Web Surface. Web mobile software-based applications are used increasingly to control and share the services provided by IoT devices via either clouds or servers. Several mobile platforms ease the process of deploying relative applications due to the use of open architecture such as the Android operating system. However, this introduces a new vector for exploiting threats and launching attacks on IoT devices. Such attacks include bluejacking, eavesdropping, blue-snarfing, DoS, and others. Additionally, cloud computing presents an additional attack vector where end-users can get data breaches, DoS, flooding attacks, and others on the cloud surfaces.

3. The Rise of Adversarial Machine Learning (AML)

Recently, researchers have reported the potential vulnerability of ML models to adversarial attacks [1,3,4]. This has promoted researchers' efforts toward AML challenges. In fact, AML is concerned with the intersection of computer security and machine learning fields. It analyzes the attacks that aim to degrade the performance of ML-based models. It also investigates the process of generating and detecting the crafted adversarial examples and how eventually incorporate possible defensive mechanisms. The area has been extensively associated with applications dealing with images as primary data modality. On the other hand, it is still growing within the fields of network traffic analysis and IoT [6,7]. This section summarizes the rise of AML from four main perspectives: understanding AML, modeling the adversary, defending against adversarial examples, and related performance evaluation.

3.1. Understanding Adversarial Examples

3.1.1. Adversarial Examples Causes and Characteristics

The criticality of adversarial examples has driven many questions about the causes behind them and how they are constructed. The analysis of such problems supports the mitigation efforts provided by researchers to manage this vulnerability. One of the causes can be the inability of the model to generalize and predict accurately the pattern of unseen data. Moreover, Goodfellow et al. [26] investigated the effect of adding perturbations to a regularized model for enhancing prediction performance. However, the reported results did not confirm the expected improvement. Other researchers [27] investigated the non-linearity of ML models in increasing the chance of constructing adversarial examples. They claimed that both linear and non-linear models can be considered for constructing adversarial examples by injecting inputs with small perturbations. According to Goodfellow et al. [26], the linear behavior of the model, in which each individual input feature is normalized, can also yield adversarial examples. Moreover, perturbing one dimension of each input will not affect the classification accuracy as effectively as perturbing all the dimensions of the inputs [28]. When dealing with adversarial examples, there are three main characteristics to be considered as follows [28]:

- Transferability. Adversarial examples can be constructed and used across several architectures and parameters of ML models which perform the same tasks. This characteristic shows the capability of those examples to be constructed by a known substitute model and then used to attack relevant unknown target models. Transferability can be categorized into two types [29]:
- Cross-data transferability: This happens when the training of both substitute and target models uses similar machine-learning techniques but with different data.
- Cross-technique transferability: This happens when the training of both substitute and target models uses the same data but with different machine-learning techniques.
- Regularization Effect. Adversarial examples can be used to enhance model robustness using adversarial training. Adversarial training solutions is adopted as defense mechanisms by several researchers. However, constructing large adversarial examples is costly in terms of computational power compared to other regularization mechanisms such as dropout [30].
- Adversarial Instability. Adversarial examples can lose the adversarial characteristics when physical effects are applied including rotation, translation, rescaling, and lighting [31]. This leads to the classification of these examples correctly which motivates attackers to enhance the robustness of adversarial examples construction methods.

However, some limitations can be faced when dealing with adversarial examples. This is inferred from the restriction in the perturbation numbers added where it is preferred to keep it at a low scale. Moreover, there might also be more optimization constraints on crafting the perturbations itself such as original content preserving, non-distinguishable perturbed input sample, and payload-constrained input [32]. This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3.1.2. Adversarial Examples Magnitude Measurement

For crafting adversarial examples, gradient-based methods are widely adopted for adding perturbation using specific distance metrics [28,33,34]:

- The L0 norm of perturbations—measures the number of mismatched (non-zero) elements between original and adversarial samples in the vector where the features' perturbed number is minimized.

- The L1 norm of perturbations—measures the total number of absolute values of the differences between original and adversarial samples in which the features' perturbed number is minimized.
- The L2 norm of perturbations—measures the Euclidean distance between original and adversarial samples in which the Euclidean distance between those data points is minimized.
- L∞ norm of perturbations—measures the maximum difference between the original and adversarial samples in which the maximum amount of perturbation is applied on any feature.

3.1.3. Adversarial Examples Crafting Methods

The crafting methods of adversarial examples have been widely investigated and a framework might be proposed for further clarification. It is worth noting that the efficiency of crafting adversarial examples comes from minimizing the total perturbation as much as possible to avoid being easily detected. Accordingly, there are two possible steps that are repeated iteratively where datapoint X is replaced with $X + \delta X$ until the adversarial goal is achieved and the perturbation δX is applied. An explanation of all equations notations is presented in Table 1. The following is the main steps of crafting methods of adversarial examples [28]:

a. Direction Sensitivity Estimation.

In this step, the sensitivity of applying changes to inputs features is measured by analyzing the data distributions around specific datapoint where the model decision boundary is possibly affected and changed. Accordingly, there are several techniques for performing direction sensitivity estimation such as the following:

- **Limited-memory Broy-den, Fletcher, Goldforb, Shanno (L-BFGS).** This method has been proposed by Szegedy et al. [6] for crafting adversarial examples through a minimization problem. In such a scenario, the adversary constructs with L2-norm an image X' similar to the original image X where X' can be labeled as a different class. This is considered a complex problem to be solved due to the use of nonlinear and non-convex functions. They tried to search for an adversarial sample by finding the minimum loss function additions to L2-norm according to the following formula [28]:

$$\min c. \parallel X - X^{'} \parallel_2 + loss_{F,I}(X^{'}) \tag{1}$$

  where c is a hyper-parameter that is randomly initialized by linear search; $\parallel X - X^{'} \parallel_2$ is L2-norm and $loss_{F,I}(*)$ is the loss function. Thus, the problem is transformed into a convex optimization process but with complicated and expensive calculations [28,35].
- **Fast Gradient Sign Method (FGSM).** This method has been proposed by Goodfellow et al. [26] for crafting adversarial examples where the cost function is calculated with regard to the gradient direction. FGSM is different from LBFGS since it uses the L1-norm and does not perform iterative processes which makes it an excellent choice when it comes to computational cost and time. In such a scenario, misclassification can occur by adding perturbations according to the following formula [28]:

$$X^{adv} = X + \epsilon sign(\nabla_{\mathcal{X}} J(X, \mathcal{Y}_{true})) \tag{2}$$

  where X^{adv} is the adversarial example of X, X is the original sample and ϵ is a hyper-parameter that is randomly initialized to control the amplitude of the disturbance. Additionally, *sign* (_) is a sign function, and J (_) is the cost function with respect to the original sample with the correct label $\mathcal{Y}_{true}$ and $\nabla_{\mathcal{X}} J$ is the gradient of X. However, this method is subjected to label leaking where other researchers suggest replacing the correct label $\mathcal{Y}_{true}$ with the predicted label [28,35,36].
- **Iterative Gradient Sign Method (IGSM).** This method has been proposed by Kurakin et al. [36] for crafting adversarial examples by optimizing the FGSM method. Per-

turbations are iteratively applied into several smaller steps followed by clipping the results which guarantees that these perturbations are close to the original samples. It is worth noting that the non-linearity of IGSM is in the gradient direction where multiple iterations are required. This reflects the simplicity of this method compared to L-BFGS and its higher success rate of the resulting adversarial samples compared to FGSM. For each iteration, the following formula is used where $Clip_{\mathcal{X},\epsilon}(*)$ denotes [$X - \epsilon, X + \epsilon$] [28]:

$$X_0^{adv} = X, X_{N+1}^{adv} = Clip_{\mathcal{X},\epsilon}\{X_N^{adv} + \alpha sign(\nabla_{\mathcal{X}} J(X_N^{adv}, \mathcal{Y}_{true}))\} \tag{3}$$

Moreover, IGSM has two distinct types of adversarial goals: (1) minimizing the confidence of the original prediction and its belongingness to the original class, or (2) maximizing the confidence of the prediction and its belongingness to the class with the lowest probability instead of the correct class [28,35].

- **Iterative Least-Likely Class Method (ILCM).** This method has been proposed by Kurakin et al. [37] for crafting adversarial examples by perturbing the target class and replacing it with the least-likely probability class for the dataset disturbance. It leads to a degradation in the classifier performance with significant errors such as misclassifying a dog as a car. The ILCM differs from FGSM and L-BFGS by identifying the exact wrong class for the adversarial examples. Moreover, it is suitable to be used when handling datasets with a considerable number of distinct classes such as ImageNet. In such a scenario, perturbations can be added according to the following formula [28]:

$$X_0^{adv} = X, X_{N+1}^{adv} = Clip_{\mathcal{X},\epsilon}\{X_N^{adv} + \alpha sign(\nabla_{\mathcal{X}} J(X_N^{adv}, \mathcal{Y}_{LL}))\} \tag{4}$$

  where the least-likely probability class is represented by $\mathcal{Y}_{LL}$. There is another option where the least-likely class is replaced with a random class as the target class which is thereby called an iteration random class method [28,35].
- **Jacobian Based Saliency Map (JSMA).** This method has been proposed by Papernot et al. [38] for crafting adversarial examples using the model's Jacobian matrix. It works by using the gradients of relative output and input components to construct a saliency map and build the gradients based on the impact of each pixel. The L0 distance norm is utilized where a limited number of the image pixels are modified, and they represent the most important pixels based on the saliency map. Therefore, gradients are significantly important in perturbing the pixel and making the prediction of the image towards the target classes. It can be performed as follows [28]:

  I. Firstly: Calculate the forward derivative $\nabla F(X)$ according to the following formula:

$$\nabla F(X) = \frac{\partial F(x)}{\partial X} = \left[\frac{\partial F_j(x)}{\partial X_i}\right]_{i \in 1 \ldots M, j \in 1 \ldots N} \tag{5}$$

  II. Secondly: Construct the saliency map S based on the calculated forward derivative.

  III. Thirdly: Select the pixel with the highest importance using the saliency map in an iterative manner until either classifying the output as the target class or maximum perturbation is achieved.

It is worth noting that JSMA is used for targeting misclassification attacks with several strengths such as a high success rate, and a high transfer rate, however, it is its main disadvantage related to its high computational cost [28,35].

Table 1. Notations of equations.

Symbol	Definition
min	Minimum distance
C	Random hyperparameter
X	Original image
X', X^{adv}	Adversarial image
$\|\|X - X'\|\|_2$	L_2-norm
$Loss_{F,l}(X')$	The loss function of (X')
ε	Random hyperparameter
Sign (*)	Sign function
J(*)	Cost function
$\mathcal{Y}_{true}$	Correct label of X
$\nabla_X J(*)$	The gradient of X
$Clip_{X,\varepsilon}(*)$	Denotes $[X - \varepsilon, X + \varepsilon]$
$\mathcal{Y}_{LL}$	Least likely (the lowest probability) target class
$\nabla F(x)$	Forward derivative

b. Perturbation Selection.

In this step, the knowledge of sensitivity is used by the adversary to select the most suitable perturbation for exploiting the model. This includes two methods as below:

Perturb all the input dimensions. Some researchers investigated the manipulation of all input dimensions where direct sensitivity estimation methods are used. In the [26] experiment, FGSM is utilized to evaluate the gradient sign direction for each input dimension and thereby minimizes the Euclidean distance between the original inputs and the related adversarial samples. However, applying such a method can be detected easily since the number of perturbations is large.

Perturb the selected input dimension. Some researchers investigated the perturbation of a selected number of input dimensions with the use of the saliency map [38]. This method contributes to limiting the number of perturbations effectively but at the price of higher computation cost.

3.2. Modelling the Adversary

The crucial use of ML models has encouraged the specifications of threat models which in turn highlights possible adversarial attack scenarios and conditions. This contributes to enhancing defense mechanisms properly by tailoring them towards specific attacks followed by measuring their performance. Huang et al. [7] introduced the AML concept where adversarial attacks are presented through a taxonomy modeling the adversarial threats according to the following aspects: goals, knowledge, and capabilities.

3.2.1. Adversarial Capabilities

Adversarial capabilities represent the potential impact of an adversary when attacking the ML models which can be grouped into two categories: Influence, and Specificity [7,29,39].

a. Influence:

This category focuses on the adversary's influence on certain classification elements such as changing the dataset or the algorithms when running attacks on the target model. Such attacks include causative, evasion, or exploratory attacks where there is an influence over either the training dataset or the testing dataset, or both. Consideration of training and testing phases is used to clarify more about the adversary's influence according to his/her capabilities in those phases, as follows [29,39]:

- Training Phase Influence: In this phase, attacks take place by influencing or corrupting the model performance in which the datasets alteration is performed, and can be summarized as follows:

   1. Data Injection: The adversary can affect the target model by injecting adversarial samples and inserting them into the training dataset. This can happen with some control over the training dataset but not over the learning algorithm.
   2. Label manipulation: The adversary can modify the training labels only and gain the most vulnerable label to degrade the model performance. The label perturbations can happen with some control over the training dataset and can be applied in a random manner to the distribution of training data. An experiment indicates that a random perturbation of the training labels can degrade the performance of shallow ML models significantly [40].
   3. Data Manipulation: The adversary can poison the training dataset before it has been used for training the target model. The adversary can modify both the labels and input features of the training data and affect the decision boundary. The training data can be accessed but without the need to access the learning algorithm.
   4. Logic Manipulation: The adversary can manipulate the learning algorithm and affect its workflow logic which thereby makes the ML model under his/her control.

- Testing Phase Influence: In this phase, attacks take place to force the target model to produce incorrect outputs without influencing it. These types of attacks use other techniques to extract useful information rather than influencing the training phases, and can be summarized as follows:
   1. Model Evasion: The adversary can evade the target model by crafting adversarial samples during the testing phase.
   2. Model Exploratory: The adversary can gain various levels of knowledge about the target model in terms of the learning algorithm and training dataset distribution pattern, as follows:
      - i. Data Distribution Access: The adversary can access the training dataset distribution of the target models. The substitute local model is built to imitate the target model in classifying a set of distribution samples. This helps in generating adversarial samples where they are used on the target model for misclassification purposes.
      - ii. Model Oracle: The adversary can only query the target model by inputting a set of samples and checking the related output labels. This access is carried out as an oracle and followed by creating a substitute local model to be used on the obtained results from the query. Then the adversary uses the adversarial samples from the substitute model to affect the target model.
      - iii. Input–Output Collection: The adversary can collect from the target model the input—output pairs to analyze the possible patterns. This is carried out without accessing the training dataset distribution.

b. Specificity:

This category focuses on attack specificity in which the determination of attack effects is clarified. It considers the attacks with multiple vectors alongside a specific vector against the target model. Attacks within this category can be further classified as follows [4,29,39]:

- Targeted: The adversary defines specific targets when performing attacks causing model misclassification into certain classes.
- Indiscriminate: The adversary has no defined targets where performing attacks causes general misclassification without specifications.

3.2.2. Adversarial Knowledge

There are various levels of knowledge about the target model where an adversary can perform the attacks. Possible knowledge elements include training data, learning algorithms, feature space, cost function, and tuned parameters. Accordingly, the adversary knowledge about the target model can fall into three types of categories as shown in Figure 4 [39]:

- Complete Knowledge: It is called White-Box Attack where the adversary has access to the whole learning process including data collection, feature extraction, feature selection, learning algorithm, and model-tuned parameters. In such a scenario, the target model is open source and access to the training dataset may be available or not to the adversary.
- Partial Knowledge: It is called Grey-Box Attack where an adversary does not have access to the training dataset and is equipped with partial knowledge about the learning process in terms of learning algorithms and the feature space. However, the adversary is not aware of either the training dataset or the tuned parameters.
- Zero Knowledge: It is called Black-Box Attack where an adversary does not have any knowledge about the majority of learning process elements including the training dataset, learning algorithm, and feature space. In such a scenario, the adversary queries the target model in which feedback on crafted query adversarial samples is used to enhance other substitute models.

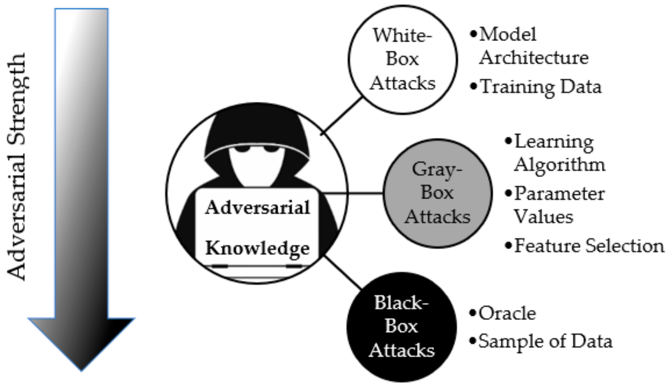

Figure 4. Adversarial Knowledge.

The adversary can evolve from a black box to a white box through an iterative learning process with the use of an inference mechanism to reach the required level of knowledge [4,39].

3.2.3. Adversarial Goals

The adversarial impact on the target ML model is used to clarify the main objectives behind the adversarial attacks as seen in Figure 5. Accordingly, the adversarial goals can be categorized based on the incorrectness of the model, as follows [29]:

- Confidence Reduction: The adversary reduces the confidence of the target model classification process. This can be seen in an example of an image recognition task where a “stop” sign is recognized with a lower confidence value with regard to the correct class belongingness.
- Misclassification: The adversary modifies the prediction of an input example and is misclassified on the decision boundary to a different class. This can be seen in an example of an image recognition task where a “stop” sign is recognized in another class that is different from the “stop” sign class.
- Targeted Misclassification: The adversary works on crafting adversarial examples and modifying the input point to be misclassified by the target model into another specific class. This can be seen in an example of an image recognition task where the “stop” sign is recognized into another specific class like the “go” sign.
- Source/target Misclassification: The adversary works on crafting adversarial examples and modifying specific input points to be misclassified by the target model into another specific class. This can be seen in an example of an image recognition task where the “stop” sign is recognized into another specific class like the “go” sign.

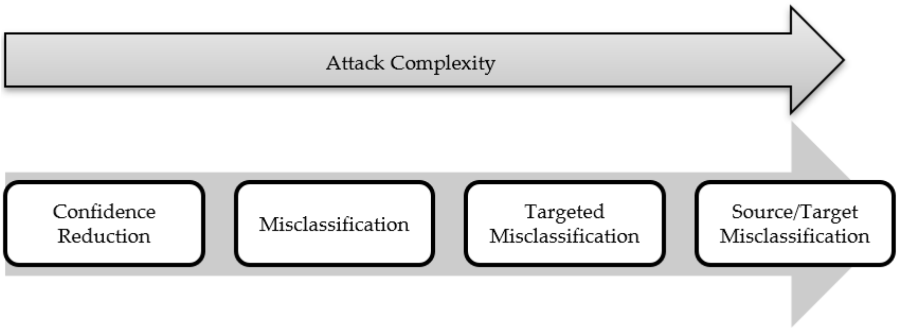

Figure 5. Adversarial Goals.

3.3. Defenses against Adversarial Examples

Since there are many methods used for crafting adversarial examples, it is essential to ensure the proper robustness of ML solutions against those vulnerabilities. According to the literature, defenses towards adversarial examples can be classified using distinct categories [32]. However, for the sake of clear mapping, the adopted classification is categorized into three main types as shown in Figure 6.

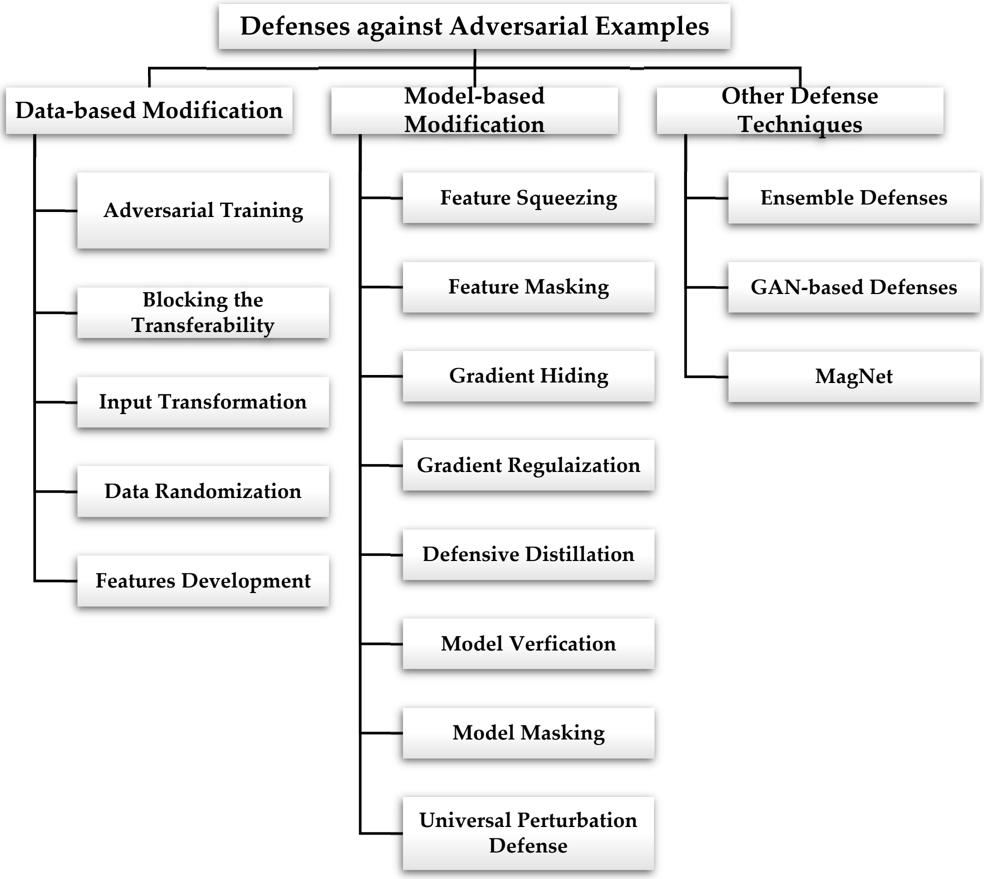

Figure 6. Taxonomy of the AML defense mechanisms is subdivided into three categories: (1) Data-based Modification, (2) Model-based Modification, and (3) Other Defense Techniques [26,41–78].

3.3.1. Data-Based Modification

This category covers the techniques of modifying the data and its related features during either the training phase or the testing phase based on the attacker's capabilities. Such techniques include:

a. Adversarial Training:

Adversarial training has been widely adopted in many research studies to enhance the robustness of ML solutions against adversarial attacks and show their defects [26,41]. The main notion behind this technique is reducing the potential misclassification results when data perturbation is fed to the ML solutions. It works on adding adversarial examples to the training data alongside generating new adversarial examples during the training epochs. Through these epochs, the characteristics of adversarial examples are controlled by the loss function where the hyper-parameters are tuned accordingly. By equalizing the number of both original examples and adversarial examples during training, the models can give better adversarial training results.

Adversarial training is used also to handle the regularization problems and thereby avoid overfitting. Since adversarial training is correlated to the training phase where white-box attacks take place, it is not robust to black-box attacks where new adversarial examples are presented [28,32]. Therefore, a study investigated the use of the ensemble adversarial training concept where different training data of pre-trained models are combined and then allow the model to generalize well on unseen inputs [42].

b. Blocking the Transferability:

Since transferability is a unique characteristic of adversarial examples, defenses in this context prevent the adversarial examples' transferability and thereby prevent black-box attacks. As mentioned previously, transferability can happen to models with different architectures or trained on different training datasets. A labeling method has been proposed to prevent the transferability between models [44]. It relies on adding a NULL label to the dataset to mitigate the adversarial examples effects where the model can detect them more efficiently. For this purpose, three steps are performed which start with training the target model as an initial step. It is followed by finding the NULL probabilities within the examined dataset and ends with adversarial training. A higher probability is assigned to the NULL label when there is more perturbation, and the probabilities of other labels are also increased for the original labeling. Thus, the method eases the detection of adversarial examples by annotating the perturbation as a NULL label instead of classifying it into one of the ranges of original labels. It also does not affect the model accuracy for the classification process of original datasets [35].

c. Input Transformation:

Some research studies have found the data transformation technique useful for enhancing the model's robustness against adversarial attacks such as FGSM [26], DeepFool [79], and universal disturbance attacks [45–47]. Such transformation includes data compression, variance minimization, bit-depth reduction, and input reconstruction which are utilized to prevent adversarial perturbations. In terms of compression, JPEG and Display Compression Technology (DCT) compression methods are used in mitigating attacks by performing compression over different image data formats such as JPEG. After training the model on those inputs, the overall accuracy is enhanced, and the attack disturbance effect can be controlled. However, the compression techniques are not well effective against more powerful and advanced attacks like Carlini & Wagner attacks [80,81]. Moreover, the increase of compression amount affects the original accuracy of the models while the decrease of its amount is not sufficient in eliminating the adversarial examples impact.

In terms of input reconstruction, a cleaning process is applied to the adversarial examples to transform them into legitimate examples in a reverse way. After the transformation takes place, the adversarial effect on the models' classification will be removed. An example of such work is presented using a deep contractive autoencoder technique, where a denoising autoencoder is used for cleaning adversarial examples [50]. However, the transformation techniques can be also used by the adversary to make the adversarial examples further stronger in the face of defense mechanisms [35,82].

d. Data Randomization:

Research in this category deals with applying different modification operations on adversarial examples such as random resizing and padding. This means that random sequences are added to the adversarial examples which reduces their effectiveness significantly. Some researchers use the two randomization operations including random resizing and random padding in the test phase. Accordingly, both non-iterative and iterative adversarial attacks can be reduced in an effective manner [48]. Other researchers utilize a separate module for conversing data where several operations of randomization are performed such as Gaussian randomization during the training phase which strengthens the model's capability [49].

e. Adversarial Robust Features Development:

Some studies focus on utilizing feature space in defending adversarial attempts against the classification process. They investigate developing adversarial robust features by studying the underlying data in terms of natural spectral geometric aspect and its relationship with the metric of interest. Their results reflect the effectiveness of the proposed approach and guarantee that any function of the dataset can have a lower bound of robustness while ensuring variation in outputs [51].

3.3.2. Model-Based Modification

This category covers the techniques of modifying the model through the methods of parameter tuning, feature selection, and so on. Such techniques include the following:

a. Feature Squeezing:

The typical features space is generally large with several least essential elements that can facilitate exploiting potential vulnerabilities of ML solutions. These vulnerabilities lead to a large interference where the adversaries can craft adversarial examples and benefit from the model's high sensitivity. Feature squeezing is achieved by reducing the feature and minimizing the complexity of data representation. The first feature squeezing method works on reducing the color-bit depth by having encoded color with fewer values. The second one applies a smoothing filter with the use of local and non-local techniques where mapping several inputs to a single value is performed. This enhances the model performance by reducing its complexity and makes it more robust but at the cost of classification accuracy. Note that some weaknesses of this technique have been recently reported [43].

b. Feature Masking:

Deep learning models can incorporate several layers when performing classification tasks. There are some sensitive features within the model inputs that need to be hidden from the adversary. Researchers found that adding a masking layer before processing the classified model can avoid potential exploitation. The masking layer works on both the original and the adversarial examples of images and calculates the most sensitive features which have the highest weight. Thus, the technique changes the corresponding weights of this additional layer to zero which leads to more privacy [52].

c. Gradient Hiding:

Some ML techniques such as decision tree, random forest, and K-Nearest Neighbor (K-NN), yield non-differentiable models which make gradient-based attacks ineffective. This inspires researchers to study the effectiveness of hiding the gradient information of models so they cannot be used to craft adversarial examples. This technique is utilized specifically for mitigating gradient-based attacks by causing numerical instabilities and restraining the adversary attempt in gradient estimation [53]. However, both black-box and white-box scenarios can be applied successfully and defeat this defense mechanism. It happens by training a surrogate classifier which is used for gradient estimation and adversarial examples generation due to the transferability characteristic of those examples [42].

d. Gradient Regularization:

Another defense solution related to gradient is proposed by the researchers in [57,58]. In particular, they used a regularization of input gradient which can be defined as a robustification method for training differentiable Deep Neural Network (DNN) models that integrate a penalty term with the gradient of loss function. In this scenario, the output variation is due to a change in the input in which a small perturbation does not affect the model output. The authors showed also that the interpretability of adversarial perturbations is increased through the adopted regularization. However, it increases the complexity in terms of computational power by a factor of two.

e. Defensive Distillation:

The basic idea of distillation was proposed for knowledge transfer from large to small networks [83]. This idea is adopted also as a defense mechanism by using the probability distribution vector produced by the first model as input to the second model. The model with large and intensive computations is simulated by a small model without changing the neural network architecture and degrading the accuracy of model performance. This helps in smoothing the training process and enhancing the model's ability of generalization to become more resilient against adversarial examples. The steps associated with the defensive distillation experiment can be summarized as follows:

- First Step: Datasets are labeled using the probability vectors produced by the first DNN. The newly produced labels are soft labels which are different from hard labels.
- Second Step: the second DNN model is trained using either the soft labels or both hard and soft labels.

Due to the knowledge transfer between models, the second model is simplified in terms of size, computational power, and training overhead keeping at the same time the needed robustness [54]. However, the defensive distillation method is not effective against some stacks such as Carlini and Wagner attack [80,81].

f. Model Verification:

ML models need a verification step to validate any processed input. As such, the model input is assessed to ensure its adherence to the model's properties and criteria which in turn supports accurate detection of new unseen adversarial examples. One research study demonstrates that this verification is an NP-complete problem and uses the Satisfiability Modulo Theory (SMT) solver to address it. For robustness, it employed Rectified Linear Unit (ReLU) activation function where neural networks are used with its whole architecture without any simplification. This means not considering only limited input regions and avoiding verification of only a problem approximation [55]. Moreover, another research work [56] investigates the local adversarial robustness of DNNs by means of discretization. Their proposed system indicates the consistency of the output label within a specific neighborhood and is then applied through all the network layers.

g. Model Masking:

The researchers in [76] investigated other defense mechanisms by presenting a noise-augmented classifier to perform masking and ensure robust classification. Accordingly, they employed a very small noise within the logit output of the DNN model. The authors showed that their incorrect logits mislead the attack. It was effective in mitigating the low-distortion attacks and preserving the accuracy of the model.

h. Universal Perturbation defense method:

The universal perturbation defense method is used in defeating adversarial examples where the ML model architecture includes a perturbation rectifying network (PRN). This network is located before the input layer as a preprocessing layer. Both clean images and images with perturbations are used for training the network. Another perturbations detector is additionally trained to denoise the inputs and extract features used to differentiate between the PRN inputs and outputs [63]. Although the method gives satisfactory results in identifying adversarial samples, the detector can be evaded using some attacks [84].

3.3.3. Other Defense Techniques

This category covers the techniques of using other supportive models in addition to the main model for robustness reasons. Such techniques include the following:

a. Ensemble Defenses

This technique combines multiple defense strategies to overcome adversarial attacks. The combination can be made either in a parallel or sequential manner for extra protection [42]. In such context, PixelDefend [59] was introduced through the combination of two defensive mechanisms including adversarial detection and input reconstruction [59]. However, the technique exhibits a considerable weakness against attacks. In particular, the transferability property of adversarial examples impacts the effectiveness of PixelDefend [85].

b. GAN–based Defenses

The idea of a Generative Adversarial Network (GAN) was first coined by Goodfellow et al. [86]. Two neural networks including generator and discriminator were coupled in a competitive manner in a defensive context against both white-box and black-box attacks. A research study suggests the use of GAN to relocate the input image within the generator range through the reduction of reconstruction error before being fed to the classifier. This leads to reducing the possibility of adversarial examples by having the benign data points closer to the generator range [60]. Another study uses a trained GAN in the cleaning process of adversarial examples within the original dataset. Using different attacks and datasets, the adversarial examples are scored lower than the original data points by the GAN's discriminator [61]. However, it is notable that the ability of interpretation and generation plays a crucial role in the success of GAN. In addition, GAN requires a sufficient level of training to ensure proper performance.

c. MagNet

MagNet [62] can be introduced as a framework that relies on two components to defend against adversarial examples. Namely, these two components are a network of detectors and a network of reformers. The first one is used to detect adversarial examples from the original ones. On the other hand, the second component projects the adversarial examples with small perturbations and transforms them into the original ones using an automatic encoder. This technique shows effectiveness against black-box and gray-box attacks while it shows a performance degradation in white-box attacks. Accordingly, using more several encoders with a random selection might mitigate the adversary's ability in the scenario of white-box attacks.

3.4. Evaluation Metrics

In the following, we outline existing evaluation measures typically used by researchers from different perspectives to assess the performance of Adversarial Machine Learning techniques.

3.4.1. Statistical Measures

The first perspective is evaluating the classifiers' performance using statistical measures widely used in the field of cyber security. This includes the confusion matrix with its related metrics including accuracy, precision, recall, F-measure, ROC (receiver operating characteristic) curve, and AUC (the area under the ROC curve). In terms of the confusion matrix, it contains several values for classification results, either they are correctly or incorrectly classified based on predefined classes. Most pre-mentioned metrics are inferred from the confusion matrix values with different combinations. The model's accuracy is mainly concerned with its robustness where models with less vulnerability to adversarial examples are preferred. The robustness of a model can be characterized based on two essential elements:

- High accuracy results are reached when the model is used on training and test datasets.
- Input's classification is consistently predicted the same for a given example.

In the context of adversarial examples, the models' robustness is affected by the perturbation size in which the model with high robustness means it requires a higher possibility of minimum perturbation to cause misclassification.

As such, these metrics cover the required information about the model and ease the process of comparison between related research works [28].

3.4.2. Security Evaluation Curves

The second perspective is evaluating the security of classifiers using security evaluation curves. Those curves can be used to investigate the classifiers' performances when there are multiple attacks with different attackers' levels of strength and knowledge. As such, those graph-based metrics are quite useful for comparing the different defense mechanisms and reflecting their performance level [39].

3.4.3. Adversarial Examples-Related Measures

The third perspective is evaluating the security of classifiers using adversarial-related measures. Those measures include:

- Success Rate: It is associated with the process of generating adversarial examples where the increment of success rate relates to a decrease in the perturbation size. This can be seen when a comparison is made between the generative methods of adversarial examples where the iterative gradient sign method (IGSM) and the Jacobian-based Saliency Map Attack (JSMA) method have a higher success rate than the fast gradient sign method (FGSM). The first two methods generate adversarial examples with lower or specific perturbations while the latter one performs large perturbations with the chance of label leaking. Nevertheless, having adversarial examples with a 100% success rate is quite difficult [28].
- Transfer Rate: It is associated with the transferability characteristic of adversarial examples where those examples can be transferred across different models. As such, the transfer rate is used for measuring which is the ratio of transferred adversarial examples number to the total adversarial example number generated by the main model. Transferability can be classified into targeted or non-targeted transferability where it is measured by matching rate and accuracy rate, respectively. It depends on two factors where the first one is the model parameters that contain its architecture, capacity, and test accuracy. A better transfer rate when it comes to the first factor can be achieved with similar architecture, small capacity, and high accuracy. The second factor is the adversarial perturbation magnitude where the higher perturbation to the original examples leads to a higher transfer rate [28].

4. Applications of AML towards Internet of Things (IoT) Robustness

In this section, the intersection between the Internet of Things (IoT) and Adversarial Machine Learning (AML) is investigated. Different research works have conducted experiments relevant to AML applications within IoT scenarios. For instance, authors in [64] proposed an iterative pipeline of defense systems against adversarial examples. It mainly encloses three elements: (i) A detector, (ii) An attack engine, and (iii) A defense mechanism. The detector is a DL-based model used for visualization-based botnet detection. It contains a feature extractor built using two consecutive convolutional layers and a classifier with a fully connected layer. On the other hand, the considered attack engine combines gradient-based adversarial attacks and GAN-based adversarial attacks. This includes FGSM [26], DeepFool [79], projected gradient descent [64], and a custom generative adversarial network named Pix2Pix conditional GAN [64]. Those components proceed harmonically to simulate the process of adversarial examples and original data transformation which are all used for crafting adversarial examples. For the defense mechanism, adversarial training is employed to strengthen the system by updating the weights of the network layers. The experiment results revealed the success of the system after a limited number of iterations. The adversarial examples number of PGD method drops from 824 to 226 only. In [65], the authors proposed an ensemble defense system named, Def-IDS, to combat adversarial attacks within Network Intrusion Detection Systems (NIDS) domain. Two modules were associated to build the system. Namely, a multi-class generative adversarial network (MGAN) that aims at generating similar examples of the multi-class intrusions instances through a single GAN model, and a multi-source adversarial retraining (MAT) model intended to retrain the detector and thereby smooth the decision boundary for a more robust classifier. It uses adversarial examples of several crafting methods including FGSM [26], DeepFool [79], JSMA [38], and BIM attacks. The second module contributes also to enhancing the detection of adversarial examples alongside ensuring the adversarial examples' transferability against multiple attacks. The resulting system was evaluated using the CSECIC-IDS2018 dataset [87] and DNN model and yielded promising results. As such, by applying the enhanced dataset to the training process, the intrusion detector is getting more robust against both known and unknown adversarial attacks.

The researchers in [66] outlined an MBAGP-CNN system that incorporates vulnerability verification and defense techniques using deep learning with a mixed batch adversarial generation process. Crafting adversarial text-based CAPTCHAs is performed using several attack methods such as FGSM [26], Iterative Fast Gradient Sign Method (I-FGSM) [36], and the Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [66] algorithms. Specifically, the system works on breaking the transferability attack by feeding both the original and the adversarial CAPTCHA images to the CNN model to perform the retraining process. The authors demonstrated the enhanced generalization capabilities of the model where it can defend against adversarial text-based CAPTCHAs compared with the image processing filters and the transfer learning models. Competitive results have been achieved, and the test accuracy of the defense model was only affected by 5% for three gradient-based adversarial attacks. The study also investigated the effect of tuning the learning rate in accelerating the convergence rate. However, even after performing a vulnerability assessment, the risk of automatic malicious attacks still exists.

In [88], the effect of various levels of data poisoning attacks on four ML models was investigated. Namely, Gradient Boosted Machines, Random Forests (RF), Naive Bayes (NB), and Feedforward Neural Network (FNN) models were used in the experiments that were intended to evaluate their robustness and compare the degradation results using different poisoning rates ranging from 5% to 30%. ToN_IoT [89] and UNSW NB-15 [90] datasets which contain a combination of benign and malicious instances were used in the experiments. The results revealed that the model performance is affected negatively in a correlation with the data poisoning increment level in terms of metrics such as accuracy and detection rates. Especially, the performance degraded significantly when associating the considered models with a 30% data poisoning rate. However, the study focuses only on the impact of poisoning attacks without considering any defense mechanism intended to reinforce the model's robustness. Additionally, a disparate feature set is used by the adopted classifiers which complicates the adversary attack attempt by increasing the time complexity.

A hierarchical ensemble learning method was introduced in [67] to defend against adversarial examples for network security classifiers. Particularly, a set of features based on the destination ports visited by a host and TCP resets behavior are used. These features set aims at distinguishing between benign and malicious traffic and detecting network scanning. In a botnet detection scenario, the target classifier accuracy dropped to 47% after performing an adversarial attack. Nevertheless, the proposed defense system enhances accuracy again to reach 100%. However, enhancing the robustness comes at the cost of computational complexity for the defender.

The researchers in [68] investigated the generation of adversarial examples by modifying the models' features. The difference between benign and malicious traffic is derived using the features that correspond to higher relevance weight. For this purpose, Information Gain Ratio is used for selecting the most key features followed by applying a perturbation to those selected features. Several ML models including Bayesian Network, Support Vector Machine (SVM), Decision Tree (DT), and Random Forest were deployed in this research as NIDS. They were evaluated against Denial of Service (DoS) attacks using an IoT network dataset. All models' performances proved to be affected and decreased by up to 47%. On the other hand, the use of adversarial training improved the model's robustness against adversarial attacks.

In [91], the authors presented Kitsune, which is a specific deep learning-based NIDS for an IoT environment. It contains four components: (i) Packet parser, (ii) Feature extractor, (iii) feature mapper, and (iv) Anomaly detector. Furthermore, the Mirai [92] dataset was used to evaluate the performance of Kitsune against four popular attacks. Namely, FGSM [26], JSMA [38], C&W [80,81], and ENM [91] were considered in white-box attack scenarios. The authors conducted experiments for measuring the success of both integrity and availability attacks, and the standard LP distance metrics [28,33,34] were used for comparison. The success rate of the integrity attacks for all the aforementioned algorithms reached 100%. In terms of availability attacks, C&W [80,81] and ENM [91] yielded success rates of 100% while FGSM [26] and JSMA [38] resulted in worse performance with 4% and 0% success rates, respectively. The authors summarized that adjusting traffic features in a limited range with only 1.38 of input features can allow adversaries to craft effective adversarial examples that are capable of bypassing NIDS.

Different crafting methods of adversarial examples such as FGSM [26], PGD, and BIM were used in [69] to evaluate the robustness of Feedforward Neural Networks (FNN) and Self-Normalizing Neural Networks (SNN). The work investigated the classification performance of these models within intrusion attacks scenario. The BoT-IoT [93] dataset was used to study the robustness of the outlined models. The experiment results revealed that SNN outperforms FNN when dealing with adversarial examples. The study suggested the role of self-normalizing features in the superior performance of SNN. Therefore, the effectiveness of feature normalization in enhancing models' robustness was also investigated.

The authors in [94] investigated the robustness of smart speakers' systems towards adversarial examples threats. Several ML models including Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (kNN), AdaBoost, and Neural Networks (NN) were utilized. A new dataset enclosing network traffic of the smart home environment was collected within 9 days with different microphone settings. Different adversarial techniques were applied to the original traffic features including the constant padding and additive white Gaussian noise (AWGN) techniques. The classification accuracy of the models was measured based on different scenarios taking into consideration the used protocols and network constraints. In [70], the effect of adversarial examples against six classifiers was explored. Particularly, KitNET [92], Multi-Layer Perceptron (MLP), Logistics Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), and Isolation Forest (IF) were used in the different experiment scenarios. Specifically, the authors used grey-box and black-box adversarial attacks, as two attack scenarios, to evaluate the model's robustness. One should mention that the attack method relies on Particle Swarm Optimization (PSO) [95] used for Traffic Mutation to mutate the given traffic using non-payload-based features. The authors investigated three different defensive schemes, including Adversarial training, features selection, and adversarial feature reduction, to strengthen their method. The experimental results proved the effectiveness of the method with more than 97% evasion rate and more than 50% evasion rate reduction for attack and defense schemes, respectively.

The researchers in [96] performed an adversarial attack against Kitsune, a deep learning-based NIDS through a black-box scenario. The feature selection has been performed using saliency maps in which the most critical features were identified. They used a combination of saliency map and FGSM [26] methods to craft the adversarial examples and attack an IDS for IoT networks named, Kitsune. Two attack scenarios from the Mirai dataset [92] have been considered to evade detection which included an IoT botnet attack and a video streaming application. In the first scenario, the attacker tries to modify and alter malicious traffic while in the latter one, he tries to manipulate benign traffic to be misclassified as malicious one. Although the modification was minimal with only 0.005% of the malicious packet bytes, the experiment results yielded an attack success rate exceeding 95%.

The authors in [71] investigated evading the NIDS detection through traffic time change technique. They introduced a timing-based adversarial network traffic reshaping attack named, TANTRA. The attack can use the timestamp attributes to reshape the malicious traffic without affecting the packet's content. Several DL-based NIDS methods were selected including KitNET [92], Autoencoder, and Isolation Forest which were further assessed using the Kitsune [92] and CIC-IDS2017 [97] datasets. The attack method proved its efficiency and gained a high success rate of 99.99%. Additionally, the authors proposed a defense mechanism by applying adversarial training on the models' benign and reshaped traffic. In [98], the utilization of generative adversarial networks (GAN) in detecting the adversarial examples and enhancing the NIDS robustness was studied. The researchers suggested maintaining attack features based on the generative adversarial networks (MAC-GAN) which is a framework for anomaly detection attacks. This framework contained two main components. The first one relies on manual analysis of the attack characteristics while GAN represents the second component that is intended to evade the detection models. The features of the network traffic can be classified into either perturbable or non-perturbable. The first category is the only one chosen for modification by GAN's generator. The CI-CIDS2017 and Kitsune datasets were used in the evaluation of the framework that includes multiple NIDS classifiers. These classifiers include KitNET, Isolation Forest, Gaussian Mixture Modelling (GMM), Support Vector Machine (SVM), Stacked Autoencoder (SAE), and Restricted Boltzmann Machine (RBM). The experimental results showed a degradation in the classifiers' performance. For instance, KitNET [92] True Positive Rate (TPR) dropped from 99.8% to 0% after the attacks. The researchers in [99] used multiple adversarial examples crafting methods including JSMA [38], FGSM [26], and C&W [80,81] attacks. The attacks were performed to evaluate multiple classifiers' performances such as MLP, RF, DT, and SVM using the BoT-IoT [93] and UNSW NB-15 [X] datasets. The experiment results confirmed the strength of the aforementioned adversarial attacks against machine learning classifiers. SVM was the most affected classifier with a decrease of accuracy of around 50% on both datasets. On the other hand, RF showed great robustness among other classifiers with a decrease in accuracy by 21%. In terms of attacks, C&W [80,81] was the most effective attack when associated with UNSW-NB15 [90] dataset. On the other hand, FGSM [26] performed well on the Bot-IoT dataset. However, JSMA [38] yielded less effect on both datasets with respect to performance metrics including accuracy, F1-measure, and recall.

The effect of FGSM [26] attack against DL-based NIDS such as Gated Recurrent Unit, LSTM, and CNN was explored in [72]. CICIDS2018 [87] dataset was used for three scenarios of the training phase. Namely, original examples training, adversarial examples training, and a combination of both prementioned training were investigated in this research. The experiment results revealed the importance of adversarial training defense techniques in enhancing the models' resilience. Particularly, LSTM model robustness has been improved in terms of accuracy. However, the decision boundary was affected after the adversarial training. Recently, the authors in [73] designed a Graph Neural Network (GNN) based solution to generate API graph embedding and identify malware from benign applications. For defensive purposes, the Generative Adversarial Network (GAN)-based algorithm, called VGAEMalGAN, was intended to attack the graph-based GNN Android malware model. Specifically, this contributes to hardening the detection model against adversarial inference attacks by retraining the model with the GAN-generated samples after being labeled as malware. In [74], the researchers studied the effect of a Membership Inference Attack (MIA) against a wireless classifier over the air. Such an attack infers some confidential information such as waveform, channel, and device characteristics. A proactive defense mechanism is proposed by developing a shadow MIA model to fool the adversarial attempt. The mechanism has successfully reduced the MIA effect on the classifier's accuracy and enhanced the privacy of the wireless signal classifier.

The authors in [75] performed an adversarial machine learning-based partial-model attack using a data poisoning technique. For IoT models, the attack was conducted within the data collection and aggregation stage of IoT systems assuming the adversary's knowledge of the decision output of the IoT fusion center. The success rate of the attack reached 83% reflecting the high vulnerability of IoT systems against such attacks. Multiple defense mechanisms were proposed without actual implementation. This includes the development of a robust anomaly detection mechanism in the IoT fusion center, improvement of privacy protection for all IoT infrastructure layers, and adversarial training. Another ML-based application designated to industrial IoT and its vulnerabilities to adversarial attacks were studied in [77]. The authors adopted crafting methods of adversarial samples based on the Jacobian-based saliency map technique against several classifiers. For the malware attack scenario, two defense mechanisms were proposed. They include specific selection criteria of adversarial samples for retraining the classifiers. The first mechanism employed the distance from the malware cluster centered on adversarial sample selection while the other used a probability measure derived from kernel-based learning (KBL) [100]. The two sample selection methods increased the detection accuracy by 6% compared to the random selection method. In [78], the researchers explored over-the-air spectrum poisoning attacks against an IoT communication system. They pointed out the energy efficiency and detection complexity of attacks. The considered attack targets the spectrum sensing period and modifies the transmitter's input data. The adopted defense strategy confirmed the robustness of the transmitter by intentionally making wrong decisions in specific time slots and thereby misleading the adversary. The time slots are selected based on the low classification scores which means these slots are not close to the decision boundary.

In [101], the authors investigated malware detection in black-box attack scenarios through the employment of a GAN-based algorithm namely (MalGAN). The proposed algorithm utilized two detectors to generate adversarial examples attacks and thereby bypassing the target detection models in non-IoT environments. For the experiment, six classifiers were used including random forest (RF), logistic regression (LR), decision trees (DT), support vector machines (SVM), multi-layer perceptron (MLP), and a voting-based ensemble of these classifiers (VOTE). The detection rate was drastically reduced to approximately zero showing the MalGAN method's superiority. The attacker is able to control adversarial examples by changing their probability distribution. For defending the attacks, adversarial training is used where they showed insufficient solutions.

The authors in [102] investigated several classifiers' resilience against multiple classifiers such as Support Vector Machines (SVM), Random Forest (RF), Convolutional Neural Networks (CNN), and Auxiliary-Classifier Generative Adversarial Networks (AC-GAN). They are evaluated in darknet traffic and application detection context to control such illegal activities. The CIC-Darknet2020 dataset is used in the performance evaluation where RF performed the best. They used AC-GAN as a sample generator for poisoning attacks to craft adversarial examples based on probability analysis. The reduction of performance in some selected classes reached zero which reflected a successful attack target. Adversarial training is employed as a defense mechanism to enhance models' detection rates. A brief of the surveyed papers is presented in Table 2.

Table 2. Summary of the Surveyed Papers.

Reference	Adversarial Technique	Defense Technique	Classifier	Dataset	Threat Model	Evaluation Metrics	Results and Findings
Taheri et al. [64]	• FGSM • DeepFool • PGD • GAN	• Adversarial Training	• CNN	• CTU-13	White-box Attacks	• Accuracy • F-measure	• The study presented a victim model which achieved a 99% rate of both accuracy and F-measure. • The study examined the robustness of the victim model where the misclassification rate for FGSM attack dropped from 673 to 237 samples.
Wang et al. [65]	• FGSM • DeepFool, • JSMA • BIM	• Adversarial Training • GAN	• DNN	• CSECIC-IDS2018	Black-box Attacks	• Precision • Recall • F-measure • Accuracy	• The study showed that the proposed system smoothed the decision boundary of the detector to detect multi-source adversarial examples. • The study showed that the proposed system does not affect the detection accuracy of original inputs.
Dankwa and Yang [66]	• FGSM • I-FGSM • MI-FGSM	• Adversarial Training	• CNN	• Private	Not defined	• Accuracy	• The study investigated the robustness of the MBAGP-CNN model using K-fold cross-validation. • The accuracy of the proposed defense model against FGSM, I-FGSM, and MI-FGSM attacks ranges between 84.30%, 83.44%, and 82.20%, respectively.
Dunn et al. [88]	• Poisoning Attacks (not identified)	• None	• Gradient Boosting Machine • RF • NB • FNN	• ToN_IoT • UNSW NB-15	White-box Attacks	• Accuracy • Precision • False Positive Rate (FPR) • Detection Rate	• The study showed that the model's accuracy was proportional to the level of poisoning by up to 20% using both datasets • The study concludes the need for further investigation of models' robustness in a defensive context.
De Lucia and Cotton [67]	• Feature Importance	• Ensemble Defenses	• Not defined	• Private	White-box Attacks	• Accuracy	• The study presented a defense in depth method through composing several classifiers for the security of wire and wireless networks • The study lacks details about the attack strategies, classifier, and used dataset.
Anthi et al. [68]	• Feature Importance	• Adversarial Training	• RF • J48 DT • Bayesian Network • SVM	• Smart Home IoT testbed	White-box Attacks	• Precision • Recall • F-measure	• The study explored the usage of the feature importance method for perturbation and crafting adversarial examples • The study confirmed the effectiveness of adversarial training in enhancing classification performance.

Table 2. *Cont.*

Reference	Adversarial Technique	Defense Technique	Classifier	Dataset	Threat Model	Evaluation Metrics	Results and Findings
Clements et al. [91]	• FGSM • JSMA • C&W • ENM	• None	• Kitsune	• Mirai	White-box Attacks	• Accuracy • False Positive Rate (FPR) • Success Rate • ·	• The study explored different attack strategies for enhancing the classifier's robustness. • The study proposed further investigation of bridging the gap between the network traffic and the adversarial examples of the DL-based model.
Ibitoye et al. [69]	• FGSM • PGD • BIM	• Feature Normalization	• DNN • SNN	• BoT-IoT	White-box Attacks	• Accuracy • Precision • Recall • F-measure	• The study provided two different DL-based NIDS models that are compared in terms of several evaluation metrics. • The study pointed out the impact of feature normalization on both robustness and detection rates.
Ranieri et al. [94]	• Constant Padding • Savitzky–Golay filters • Additive White Gaussian Noise (AWGN)	• None	• AB • DT • kNN • RF • NN	• Private	Black-box Attacks	• Accuracy	• The study investigated various privacy threats that can exploit smart speakers such as Google Home Mini smart speaker • The study revealed the lack of sufficient adversarial learning countermeasures for a smart speaker system using different attack strategies.
Han et al. [70]	• PSO-based Traffic Mutation Algorithm	• Adversarial Training • Feature Selection • Adversarial Feature Reduction	• KitNET • MLP • LR • DT • SVM • IF	• Kitsune • CIC-IDS2017	- Grey-box Attacks - Black-box Attacks	• Precision • Recall • F-measure • Detection Evasion Rate (DER) • Malicious traffic Evasion Rate (MER) • Malicious Probability Decline Rate (PDR) • Malicious features Mimicry Rate (MMR)	• The study showed the automatic mutation attack of original traffic while preserving an affordable execution overhead • The proposed method included both attack and defense mechanisms to investigate the robustness of various NIDSs using different ML/DL models and non-payload-based features.
Qiu et al. [96]	• FGSM • Saliency Map	• None	• KitNET	• Mirai	Black-box Attacks	• Success Rate • Accuracy • False Positive Rate (FPR)	• The study presented two scenarios of adversarial technique attacks toward DL-based NIDS in a black-box environment. • The study showed the significant degradation of NIDS detection accuracy with only minimal manipulation for the traffic.

Table 2. *Cont.*

Reference	Adversarial Technique	Defense Technique	Classifier	Dataset	Threat Model	Evaluation Metrics	Results and Findings
Sharon et al. [71]	• Timing-Based Adversarial Network Traffic Reshaping Attack (TANTRA)	• Adversarial Training	• KitNET • Autoencoder • IF	• CICIDS2017 • Kitsune	Black-box Attacks	• Detection Rate • False Positive Rate (FPR)	• The study presented a timing-based adversarial network traffic reshaping attack named, TANTRA that showed a promising result. • The study investigated the classifier robustness by performing training on both benign traffic and reshaped malicious traffic.
Zhong et al. [98]	• WGAN	• None	• KitNET • IF • RBM • SAE • SVM • GMM	• CICIDS2017 • Kitsune	Black-box Attacks	• True Positive Rate (TPR)	• The study utilized the perturbable features of traffic to bypass ML-based detectors. • The study verified the strength of GAN in evading state-of-the-art ML-based detectors. • .
Pacheco and Sun [99]	• FGSM • JSMA • C&W	• None	• RF • DT • SVM • MLP	• UNSW-NB15 • Bot-IoT	White-box Attacks	• Accuracy • Area Under the Curve (AUC) • Recall • F-measure	• The study showed the effectiveness of several attacks in decreasing the classifiers' detection performance. • The study utilized contemporary datasets that reflected the modern network environment to analyze the strength of several attack methods.
Fu et al. [72]	• FGSM	• Adversarial Training	• CNN • LSTM • GRU	• CICIDS2018	White-box Attacks	• Accuracy	• The study explored different DL-based NIDS while they are tested under FGSM attack. • The study investigated the effect of defense mechanisms on the overall performance of classifiers' accuracy.
Yumlembam et al. [73]	• GAN	• Adversarial Training • GAN	• GNN	• CICMaldroid • Drebin	White-box Attacks	• Accuracy • Precision • Recall • F-measure	• The study explored the effectiveness of graph-based classification using GNN to generate API graph embedding. • The proposed detection system achieved an accuracy of nearly 98.43% after retraining using GAN-based- adversarial malware.
Shi et al. [74]	• Member Inference Attack (MIA)	• Adversarial Training	• DNN	• Private	Black-box Attacks	• Accuracy	• The study showed an increase in member sample accuracy while a decrease in non-member samples. • The study developed a defense scheme by adding carefully crafted perturbations in the classification process which affect the MIA work.

Table 2. *Cont.*

Reference	Adversarial Technique	Defense Technique	Classifier	Dataset	Threat Model	Evaluation Metrics	Results and Findings
Luo et al. [75]	• Poisoning Attack	• Adversarial Training • Privacy Improvement	• CNN • SVM	• Private	White-box Attacks	• Success Ratio	• The study revealed the vulnerabilities of IoT systems towards adversarial machine learning-based partial-model attack • The attack disrupted the decision-making in the process of data fusion of IoT with limited control of only 8 IoT devices out of 20 devices.
Khoda et al. [77]	• JSMA	• Adversarial Training	• SVM • RF • BN • DNN	• Drebin	White-box Attacks	• Accuracy • Precision • Recall • F-measure	• The study focused on enhancing the defense mechanisms of malware detection models against adversarial samples. • The defense mechanisms are based on adversarial training where two sample selection methods are used and outperformed the random ones.
Sagduyu et al. [78]	• Poisoning Attack	• Adversarial Training • Features Development	• DNN	• Private	White-box Attacks	• Success Rate • Misdetection Rate • False Alarm	• The study investigated the AML attack's impact on over-the-air spectrum sensing during both the test and training phases. • The study showed the effectiveness of the proposed defense mechanism in increasing the errors in the adversary's decisions and preventing performance degradation of the transmitter.
Weiwei et al. [101]	• MalGAN	• Adversarial Training	• RF • LR • DT • SVM • MLP • VOTE	• Private	Black-box Attacks	• TPR	• The study explored the use of GAN in malware and black-box scenarios to enhance the detection rate of IDS • The study indicated the strength of the proposed algorithm (MalGAN) in decreasing the detection rate with the use of adversarial retraining as a defense mechanism.
Rust-Nguyen et al. [102]	• Poisoning Attack	• Adversarial Training	• SVM • CNN • RF • AC-GAN	• CIC-Darknet2020 dataset	White-box Attacks	• Accuracy	• The study evaluated darknet traffic detection with consideration of adversarial attacks' impact. • The study reflected the successful attempt of AML attacks against the best-performing classifiers while the defense method is not detailed.

5. Discussion & Research Directions

Adversarial Machine Learning has been applied to evaluate the robustness of ML-based NIDS by proposing context-related methods for attack, defense, or both. Despite the noticeable progress in exploiting AML within NIDS applications, the surveyed works exhibit a considerable scarcity of recent comprehensive surveys relevant to AML applications within IoT frameworks, specifically.

In the case of attack methods, several adversarial crafting techniques have been proposed. One can notice that FGSM [26] was the most used one followed by JSMA [38] and C&W [80,81] attacks. However, FGSM [26] might be an impractical option for the NIDS field since it works on perturbing each possible feature for crafting adversarial examples. Those features exhibit high dependency, strong correlation, and hard constraints. This makes the process impractical and even harder to deploy. Additionally, the process of perturbation is challenging since the adversary needs complete control to perform such an operation. In terms of attack strategies, most of the works assume a black box setting that is suitable for simulating real-world attacks where the attacker can have access to the system's outputs with limited knowledge about the inputs. Additionally, the validity and properness of adversarial examples generated by generic crafting techniques are not assessed sufficiently. This yields potential inconsistency in the generated adversarial traffic. The features of network traffic are structured using specific data types and value ranges. Namely, they can have binary, categorical, or continuous values compared to other domains such as image recognition where such constraints are relaxed. Moreover, many ML algorithms are designed to manage numerical values which require the conversion of categorical values into multiple binary features. This conversion results in serious flaws in applying the perturbations by the famous AML crafting methods such as FGSM [103]. It is worth noting the criteria for selecting the proper crafting techniques by considering several important aspects such as computational power, features space, and perturbations magnitude. Consequently, the validity of adversarial examples can be measured by ensuring limited perturbations, defined values range, retained features semantic relations, and preserved network traffic information [104].

Adversarial crafting methods have been performed against a wide range of ML-based classifiers including both shallow and deep learning methods such as NB, SVM, DT, RF, MLP, CNN, and DNN. Notably, the reported performance of these classifiers degraded drastically. In particular, CNN, DT, and NB yielded the worst results. Additionally, one can notice the lack of publicly available datasets for AML applications in the IoT domain. In fact, most of the state-of-the-art works used Kitsune [92] and CICIDS (2017/2018) [87,97] datasets. As such, there is an urgent need to have benchmarking datasets that reflect the current IoT network structures and features in order to evaluate the performance of NIDSs under adversarial machine learning attacks.

Even though different attack strategies have been studied, only a few research consider defense mechanisms for enhancing the robustness of ML-based classifiers. Moreover, no customized mechanisms have been proposed to defend IoT against AML attacks compared to image recognition applications. Moreover, some studies did not incorporate any defense solutions while others used a limited number of generic techniques. Actually, adversarial training is the most used defense technique compared to feature reduction, ensemble learning, and GAN-based defense. NIDS requires a real-time operation to allow proper processing of the network's traffic. However, several challenges such as process overhead, computational power, and memory requirements are faced by defense mechanisms in the context of NIDS in IoT environments. For example, feature reduction and adversarial training might affect the decision boundary and yield a misclassification of the original examples as negative cases. Additionally, ensemble learning leads to high complexity in terms of the computational power of training and deploying the model. Thus, it is quite challenging to balance between the good detection of original examples and the robust performance towards adversarial ones.

Recent research works have paid attention to Reinforcement Learning (RL) as a promising research area to tackle the IoT challenges [3]. RL is used with agents that can interact with the surrounding environment for rewards as optimization targets. Such an agent is linked with the IoT environment and exposed to several AML challenges including convergence and real-time problems. Several significant research gaps are found that need to be considered as extensions of IoT potential applications. Consideration needs to be given to both the computational and storage overhead of RL [3,105]. Another trending solution is the use of federated learning which allows the deployment of ML applications on the large scale of IoT. It can provide decentralized, collaborative, and privacy-preservation solutions while handling IoT specifications. However, it is also subjected to AML attacks that can deceive the FL-based model and thereby requires more sophisticated defense methods for further investigation [106].

6. Conclusion & Future Works

Machine learning approaches within IoT networks have been gaining an increasing popularity since there are large constraints found within IoT environments. Thus, ML-based NIDSs are widely adopted within IoT systems to enhance the detection rates of any potential attacks and threats. However, those systems are subjected to adversarial examples threats that degrade the overall performance significantly. This reflects the importance of studying the AML characteristics and features to boost the robustness of the ML-based detectors. This can help in mitigating catastrophic impacts on the IoT network in the case of security issues. Thus, this study presents an in-depth overview of the latest research progress on the intersection between IoT and AML fields. In such a context, a wide variety of adversarial attack generation and defense methods have been explored to analyze the current research works. The surveyed studies contribute to forming the maturity levels of current research on AML and IoT in terms of three main aspects: attack methods, defense methods, and evaluation metrics. Moreover, a discussion of the most important findings and results are presented in terms of strength, weakness, and future directions. To sum up, this paper extensively investigates, analyses, and discusses the current literature works to find out gaps, shortcomings, and limitations. A conclusion is reached where there are some specifications of IoT networks that require tailored solutions to address the related issues. For future enhancement, federated learning (FL) represents an emerging paradigm [14] that allows the deployment of ML applications on the large scale of IoT and its highly constrained networks while maintaining their privacy and security. Such a paradigm enhances the design of global detection models which are composed of sub-models from other IoT devices in a distributed manner. This distributed learning architecture can be explored further in future work to address potential security attacks and possible robustness issues.

Author Contributions: Conceptualization, S.A. and S.A.-A.; methodology, S.A.; software, S.A.; validation, S.A. and M.M.B.I.; formal analysis, S.A.; investigation, S.A.; resources, S.A., S.A.-A. and M.M.B.I.; data curation, S.A.; writing—original draft preparation, S.A.; writing—review and editing, S.A., S.A.-A. and M.M.B.I.; visualization, S.A.; supervision, S.A.-A. and M.M.B.I.; project administration, M.M.B.I.; funding acquisition, S.A.-A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Deanship of Scientific Research at King Saud University for funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at King Saud University for funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR).

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

References

1. Alsheikh, M.A.; Lin, S.; Niyato, D.; Tan, H.-P. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1996–2018. [CrossRef]
2. Butun, I.; Morgera, S.D.; Sankar, R. A Survey of Intrusion Detection Systems in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 266–282. [CrossRef]
3. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine learning in IoT security: Current Solutions and Future Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. [CrossRef]
4. Duddu, V. A Survey of Adversarial Machine Learning in Cyber Warfare. *Def. Sci. J.* **2018**, *68*, 356. [CrossRef]
5. Qiu, J.; Wu, Q.; Ding, G.; Xu, Y.; Feng, S. A survey of machine learning for big data processing. *EURASIP J. Adv. Signal Process.* **2016**, 67. [CrossRef]
6. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. *arXiv* **2013**, arXiv:1312.6199.
7. Huang, L.; Joseph, A.D.; Nelson, B.; Rubinstein, B.I.P.; Tygar, J.D. Adversarial Machine Learning. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, Chicago, IL, USA, 21 October 2011; pp. 43–58.
8. Springer. Available online: https://www.springer.com/gp (accessed on 1 July 2022).
9. IEEE Xplore. Available online: https://ieeexplore.ieee.org/Xplore/home.jsp (accessed on 1 July 2022).
10. arXiv. Available online: https://arxiv.org/ (accessed on 1 July 2022).
11. Science Direct. Available online: https://www.sciencedirect.com/ (accessed on 1 June 2022).
12. Research Gate. Available online: https://www.researchgate.net/ (accessed on 1 June 2022).
13. Ala, A.F.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376.
14. Nguyen, T.D.; Rieger, P.; Miettinen, M.; Sadeghi, A.-R. Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System. In Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS) 2020, San Diego, CA, USA, 23–26 February 2020.
15. Saadeh, M.; Sleit, A.; Sabri, K.E.; Almobaideen, W. Hierarchical architecture and protocol for mobile object authentication in the context of IoT smart cities. *J. Netw. Comput. Appl.* **2018**, *121*, 119. [CrossRef]
16. Gazis, V. A Survey of Standards for Machine-to-Machine and the Internet of Things. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 482–511. [CrossRef]
17. Yang, Z.; Yue, Y.; Yang, Y.; Peng, Y.; Wang, X.; Liu, W. Study and Application on the Architecture and Key Technologies for IOT. In Proceedings of the IEEE International Conference Multimedia Technology (ICMT), Hangzhou, China, 26–28 July 2011; pp. 747–751.
18. Wu, M.; Lu, T.-J.; Ling, F.-Y.; Sun, J.; Du, H.-Y. Research on the Architecture of Internet of Things. In Proceedings of the IEEE 3rd International Conference Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, 20–22 August 2010; Volume 5, pp. V5-484–V5-487.
19. Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information* **2021**, *12*, 87. [CrossRef]
20. Sari, R.F.; Rosyidi, L.; Susilo, B.; Asvial, M. A Comprehensive Review on Network Protocol Design for Autonomic Internet of Things. *Information* **2021**, *12*, 292. [CrossRef]
21. Bout, E.; Loscri, V.; Gallais, A. How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2021**, *24*, 248–279. [CrossRef]
22. Makhdoom, I.; Abolhasan, M.; Lipman, J.; Liu, R.P.; Ni, W. Anatomy of Threats to the Internet of Things. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1636–1675. [CrossRef]
23. Farris, I.; Taleb, T.; Khettab, Y.; Song, J.S. A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 812–837. [CrossRef]
24. Tahsien, S.M.; Karimipour, H.; Spachos, P. Machine learning based solutions for security of Internet of Things (IoT): A survey. *J. Netw. Comput. Appl.* **2020**, *161*, 102630. [CrossRef]
25. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 2481–2501. [CrossRef]
26. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
27. Philipp, G.; Carbonell, J.G. The Nonlinearity Coefficient—Predicting Overfitting in Deep Neural Networks. *arXiv* **2018**, arXiv:1806.00179.

28. Zhang, J.; Li, C. Adversarial Examples: Opportunities and Challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 2578–2593. [CrossRef]
29. Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; Mukhopadhyay, D. A Survey on Adversarial Attacks and Defences. *CAAI Trans. Intell. Technol.* **2021**, *6*, 25–45. [CrossRef]
30. Ko, B.; Kim, H.; Oh, K.; Choi, H. Controlled Dropout: A Different Approach to Using Dropout on Deep Neural Network. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Republic of Korea, 13–16 February 2017; pp. 358–362.
31. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing Robust Adversarial Examples. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 3 July 2018; pp. 284–293.
32. Qayyum, A.; Usama, M.; Qadir, J.; Al-Fuqaha, A. Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and the Way Forward. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 998–1026.
33. Zhao, P.; Liu, S.; Wang, Y.; Lin, X. An ADMM-based Universal Framework for Adversarial Attacks on Deep Neural Networks. In Proceedings of the 26th ACM international conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 1065–1073.
34. Martins, N.; Cruz, J.M.; Cruz, T.; Abreu, P.H. Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review. *IEEE Access* **2020**, *8*, 35403–35419. [CrossRef]
35. Qiu, S.; Liu, Q.; Zhou, S.; Wu, C. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Appl. Sci.* **2019**, *9*, 909. [CrossRef]
36. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Machine Learning at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
37. Kurakin, A.; Goodfellow, I. Adversarial Examples in the Physical World. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
38. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Berkay Celik, Z.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, Germany, 21–24 March 2016; pp. 372–387.
39. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.* **2018**, *84*, 317–331. [CrossRef]
40. Biggio, B.; Nelson, B.; Laskov, P. Support Vector Machines under Adversarial Label Noise. In Proceedings of the Asian Conference on Machine Learning, Taoyuan, Taiwan, 13–15 November 2011; pp. 97–112.
41. Huang, R.; Xu, B.; Schuurmans, D.; Szepesv'ari, C. Learning with A Strong Adversary. *arXiv* **2015**, arXiv:1511.03034.
42. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble Adversarial Training: Attacks and Defenses. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
43. Xu, W.; Evans, D.; Qi, Y. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In Proceedings of the 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, CA, USA, 18–21 February 2018.
44. Hosseini, H.; Chen, Y.; Kannan, S.; Zhang, B.; Poovendran, R. Blocking Transferability of Adversarial Examples in Black-box Learning Systems. *arXiv* **2017**, arXiv:1703.04318.
45. Dziugaite, G.; Ghahramani, Z.; Roy, D.M. A Study of the Effect of JPG Compression on Adversarial Images. *arXiv* **2016**, arXiv:1608.00853.
46. Das, N.; Shanbhogue, M.; Chen, S.; Hohman, F.; Chen, L.; Kounavis, M.E.; Chau, D.H. Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning with JPEG Compression. *arXiv* **2017**, arXiv:1705.02900.
47. Guo, C.; Rana, M.; Cisse, M.; van der Maaten, L. Countering Adversarial Images using Input Transformations. In Proceedings of the International Conference on Learning Representation (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
48. Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; Yuille, A. Adversarial Examples for Semantic Segmentation and Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1369–1378.
49. Wang, Q.; Guo, W.; Zhang, K.; Ororbia II, A.G.; Xing, X.; Liu, X.; Lee Giles, C. Learning Adversary-resistant Deep Neural Networks. *arXiv* **2016**, arXiv:1612.01401.
50. Gu, S.; Rigazio, L. Towards Deep Neural Network Architectures Robust to Adversarial Examples. *arXiv* **2014**, arXiv:1412.5068.
51. Garg, S.; Sharan, V.; Zhang, B.; Valiant, G. A Spectral View of Adversarially Robust Features. *Adv. Neural Inf. Process. Syst.* **2018**, 10159–10169.
52. Gao, J.; Wang, B.; Lin, Z.; Xu, W.; Qi, Y. Deepcloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples. *arXiv* **2017**, arXiv:1702.06763.
53. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Berkay Celik, Z.; Swami, A. Practical Black-box attacks against Machine Learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 506–519.

54. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as A Defense to Adversarial Perturbations against Deep Neural Networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 26 May 2016; pp. 582–597.
55. Katz, G.; Barrett, C.; Dill, D.L.; Julian, K.; Kochenderfer, M.J. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In Proceedings of the International Conference on Computer Aided Verification, Heidelberg, Germany, 22–28 July 2017; pp. 97–117.
56. Huang, X.; Kwiatkowska, M.; Wang, S.; Wu, M. Safety Verification of Deep Neural Networks. *arXiv* **2016**, arXiv:1610.06940.
57. Ross, A.S.; Doshi-Velez, F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
58. Lyu, C.; Huang, K.; Liang, H. A Unified Gradient Regularization Family for Adversarial Examples. In Proceedings of the 2015 IEEE International Conference on Data Mining, Atlantic, NJ, USA, 14–17 November 2015; pp. 301–309.
59. Song, Y.; Kim, T.; Nowozin, S.; Ermon, S.; Kushman, N. Pixeldefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 3 May–30 April 2018.
60. Samangouei, P.; Kabkab, M.; Chellappa, R. Defense-GAN: Protecting Classifiers against Adversarial Attacks using Generative Models. *arXiv* **2018**, arXiv:1805.06605.
61. Santhanam, G.K.; Grnarova, P. Defending against Adversarial Attacks by Leveraging an Entire GAN. *arXiv* **2018**, arXiv:1805.10652.
62. Meng, D..; Chen, H. Magnet: A Two-pronged Defense against Adversarial Examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 135–147.
63. Akhtar, N.; Liu, J.; Mian, A. Defense against Universal Adversarial Perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake, UT, USA, 18–23 June 2018.
64. Taheri, S.; Khormali, A.; Salem, M.; Yuan, J. Developing a Robust Defensive System against Adversarial Examples Using Generative Adversarial Networks. *Big Data Cogn. Comput.* **2020**, *4*, 11. [CrossRef]
65. Wang, J.; Pan, J.; AlQerm, I.; Liu, Y. Def-IDS: An Ensemble Defense Mechanism against Adversarial Attacks for Deep Learning-based Network Intrusion Detection. In Proceedings of the 2021 IEEE International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021; pp. 1–9.
66. Dankwa, S.; Yang, L. Securing IoT Devices: A Robust and Efficient Deep Learning with a Mixed Batch Adversarial Generation Process for CAPTCHA Security Verification. *Electronics* **2021**, *10*, 1798. [CrossRef]
67. De Lucia, M.J.; Cotton, C. A Network Security Classifier Defense: Against Adversarial Machine Learning Attacks. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Linz, Austria, 13 July 2020; pp. 67–73.
68. Anthi, E.; Williams, L.; Javed, A.; Burnap, P. Hardening Machine Learning Denial of Service (DoS) Defences against Adversarial Attacks in IoT Smart Home Networks. *Comput. Secur.* **2021**, *108*, 102352. [CrossRef]
69. Ibitoye, O.; Shafiq, O.; Matrawy, A. Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Big Island, HI, USA, 9–13 December 2019; pp. 1–6.
70. Han, D.; Wang, Z.; Zhong, Y.; Chen, W.; Yang, J.; Lu, S.; Shi, X.; Yin, X. Evaluating and Improving Adversarial Robustness of Machine Learning-Based Network Intrusion Detectors. *IEEE J. Sel. Areas Commun.* **2021**, *99*, 2632–2647. [CrossRef]
71. Sharon, Y.; Berend, D.; Liu, Y.; Shabtai, A.; Elovici, Y. TANTRA: Timing-Based Adversarial Network Traffic Reshaping Attack. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3225–3237. [CrossRef]
72. Fu, X.; Zhou, N.; Jiao, L.; Li, H.; Zhang, J. The robust deep learning–based schemes for intrusion detection in Internet of Things environments. *Ann. Telecommun.* **2021**, *76*, 273–285. [CrossRef]
73. Yumlembam, R.; Issac, B.; Jacob, S.M.; Yang, L. IoT-based Android Malware Detection using Graph Neural Network with Adversarial Defense. *IEEE Internet Things J.* **2022**, *10*, 8432–8444. [CrossRef]
74. Shi, Y.; Sagduyu, Y.E. Membership Inference Attack and Defense for Wireless Signal Classifiers with Deep Learning. *arXiv* **2022**, arXiv:2107.12173. [CrossRef]
75. Luo, Z.; Zhao, S.; Lu, Z.; Sagduyu, Y.E.; Xu, J. Adversarial Machine Learning based Partial-model Attack in IoT. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning (WiseML '20). Association for Computing Machinery, New York, NY, USA, 13–18 July 2020.
76. Nguyen, L.; Wang, S.; Sinha, A. A Learning and Masking Approach to Secure Learning. In *Decision and Game Theory for Security*; GameSec 2018; Lecture Notes in Computer Science; Bushnell, L., Poovendran, R., Başar, T., Eds.; Springer: Cham, Switzerland, 2018; pp. 453–464.
77. Khoda, M.E.; Imam, T.; Kamruzzaman, J.; Gondal, I.; Rahman, A. Robust Malware Defense in Industrial IoT Applications using Machine Learning with Selective Adversarial Samples. *IEEE Trans. Ind. Appl.* **2019**, *56*, 4415–4424. [CrossRef]
78. Sagduyu, Y.; Shi, Y.; Erpek, T. Adversarial Deep Learning for Over-the-air Spectrum Poisoning Attacks. *IEEE Trans. Mob. Comput.* **2019**, *10*, 1. [CrossRef]

79. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
80. Carlini, N.; Wagner, D. Defensive Distillation is Not Robust to Adversarial Examples. *arXiv* **2016**, arXiv:1607.04311.
81. Carlini, N.; Wagner, D. Adversarial Examples are Not Easily Detected: Bypassing Ten Detection Methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017; pp. 3–14.
82. Xue, M.; Yuan, C.; Wu, H.; Zhang, Y.; Liu, W. Machine Learning Security: Threats, Countermeasures, and Evaluations. *IEEE Access* **2020**, *8*, 74720–74742. [CrossRef]
83. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2014**, arXiv:1503.02531.
84. Sadi, M.; Talukder, B.M.S.; Mishty, K.; Rahman, M.T. Attacking Deep Learning AI Hardware with Universal Adversarial Perturbation. *arXiv* **2021**, arXiv:2111.09488.
85. He, W.; Wei, J.; Chen, X.; Carlini, N.; Song, D. Adversarial Example Defense: Ensembles of Weak Defenses are Not Strong. In Proceedings of the 11th USENIX Workshop on Offensive Technologies (WOOT)'17), Vancouver, BC, Canada, 14–15 August 2017.
86. Goodfellow, I. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv* **2014**, arXiv:1701.00160.
87. CSE-CIC-IDS2018 Dataset. Available online: https://www.unb.ca/cic/datasets/ids-2018.html. (accessed on 1 October 2022).
88. Dunn, C.; Moustafa, N.; Turnbull, B. Robustness Evaluations of Sustainable Machine Learning Models against Data Poisoning Attacks in the Internet of Things. *Sustainability* **2020**, *12*, 6434. [CrossRef]
89. ToN_IoT Datasets. 2019. Available online: https://search.datacite.org/works/10.21227/feszdm97# (accessed on 1 December 2022).
90. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Dataset for Network Intrusion Detection Systems. In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
91. Clements, J.; Yang, Y.; Sharma, A.A.; Hu, H.; Lao, Y. Rallying Adversarial Techniques against Deep Learning for Network Security. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–8.
92. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *arXiv* **2018**, arXiv:1802.09089.
93. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *arXiv* **2018**, arXiv:1811.00701. [CrossRef]
94. Ranieri, A.; Caputo, D.; Verderame, L.; Merlo, A.; Caviglione, L. Deep Adversarial Learning on Google Home devices. *J. Internet Serv. Inf. Secur.* **2021**, *11*, 33–43.
95. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
96. Qiu, H.; Dong, T.; Zhang, T.; Lu, J.; Memmi, G.; Qiu, M. Adversarial Attacks against Network Intrusion Detection in IoT Systems. *IEEE Internet Things J.* **2020**, *99*, 10327–10335. [CrossRef]
97. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 2018 International Conference on Information Systems Security and Privacy(ICISSP), Funchal, Portugal, 22–24 June 2018; pp. 108–116.
98. Zhong, Y.; Zhu, Y.; Wang, Z.; Yin, X.; Shi, X.; Li, K. An adversarial Learning Model for Intrusion Detection in Real Complex Network Environments. In Proceedings of the Wireless Algorithms, Systems, and Applications: 15th International Conference, WASA 2020, Qingdao, China, 13–15 September 2020; Lecture Notes in Computer Science. Yu, D., Dressler, F., Yu, J., Eds.; Springer: Cham, Switzerland, 2020; pp. 794–806.
99. Pacheco, Y.; Sun, W. Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets. In Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021), Vienna, Austria, 11–13 February 2021; pp. 160–171.
100. Müller, K.; Mika, S.; Tsuda, K.; Schölkopf, K. An Introduction to Kernel-based Learning Algorithms. In *Handbook of Neural Network Signal Processing*; CRC Press: Boca Raton, FL, USA, 2018.
101. Weiwei, H.; Tan, Y. Generating Adversarial Malware Examples for Black-box Attacks based on GAN. In Proceedings of the Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, 21–24 November 2022; Springer Nature: Singapore, 2023. Part II. pp. 409–423.
102. Rust-Nguyen, N.; Sharma, S.; Stamp, M. Darknet Traffic Classification and Adversarial Attacks Using Machine Learning. *Comput. Secur.* **2023**, *127*, 103098. [CrossRef]
103. Merzouk, M.A.; Cuppens, F.; Boulahia-Cuppens, N.; Yaich, R. A Deeper Analysis of Adversarial Examples in Intrusion Detection. In Proceedings of the Risks and Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, France, 4–6 November 2020; Lecture Notes in Computer Science. Garcia-Alfaro, J., Leneutre, J., Cuppens, N., Yaich, R., Eds.; Springer: Cham, Switzerland, 2020; Volume 12528, pp. 67–84.
104. Alatwi, H.; Morisset, C. Adversarial Machine Learning in Network Intrusion Detection Domain: A Systematic Review. *arXiv* **2021**, arXiv:2112.03315.

105. Standen, M.; Kim, J.; Szabo, C. SoK: Adversarial Machine Learning Attacks and Defenses in Multi-Agent Reinforcement Learning. *arXiv* **2023**, arXiv:2301.04299.
106. Nair, A.K.; Raj, E.D.; Sahoo, J. A Robust Analysis of Adversarial Attacks on Federated Learning Environments. *Comput. Stand. Interfaces* **2023**, *86*, 103723. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.