future internet

MDPI

Review

A Holistic Review of Machine Learning Adversarial Attacks in IoT Networks

Hassan Khazane [1], Mohammed Ridouani [1], Fatima Salahdine [2,*] and Naima Kaabouch [3,*]

[1] RITM Laboratory, CED Engineering Sciences, ENSEM, Hassan II University, Casablanca 20000, Morocco; hassan.khazane-etu@etu.univh2c.ma (H.K.); mohammed.ridouani@etu.univh2c.ma (M.R.)
[2] Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223, USA
[3] School of Electrical and Computer Science, University of North Dakota, Grand Forks, ND 58202, USA
* Correspondence: fsalahdi@uncc.edu (F.S.); naima.kaabouch@und.edu (N.K.)

Abstract: With the rapid advancements and notable achievements across various application domains, Machine Learning (ML) has become a vital element within the Internet of Things (IoT) ecosystem. Among these use cases is IoT security, where numerous systems are deployed to identify or thwart attacks, including intrusion detection systems (IDSs), malware detection systems (MDSs), and device identification systems (DISs). Machine Learning-based (ML-based) IoT security systems can fulfill several security objectives, including detecting attacks, authenticating users before they gain access to the system, and categorizing suspicious activities. Nevertheless, ML faces numerous challenges, such as those resulting from the emergence of adversarial attacks crafted to mislead classifiers. This paper provides a comprehensive review of the body of knowledge about adversarial attacks and defense mechanisms, with a particular focus on three prominent IoT security systems: IDSs, MDSs, and DISs. The paper starts by establishing a taxonomy of adversarial attacks within the context of IoT. Then, various methodologies employed in the generation of adversarial attacks are described and classified within a two-dimensional framework. Additionally, we describe existing countermeasures for enhancing IoT security against adversarial attacks. Finally, we explore the most recent literature on the vulnerability of three ML-based IoT security systems to adversarial attacks.

Keywords: adversarial attacks; adversarial examples; machine learning; deep learning; Internet of Things; intrusion detection system; malware detection system; device identification system

Citation: Khazane, H.; Ridouani, M.; Salahdine, F.; Kaabouch, N. A Holistic Review of Machine Learning Adversarial Attacks in IoT Networks. *Future Internet* **2024**, *16*, 32. https://doi.org/10.3390/fi16010032

Academic Editors: Georgios Kambourakis and Gianluigi Ferrari

Received: 14 November 2023
Revised: 12 January 2024
Accepted: 15 January 2024
Published: 19 January 2024

1. Introduction

According to Statista [1], there will be about 30.9 billion interconnected IoT devices, while non-IoT connections including smartphones, laptops, and computers are estimated to be just over 10 billion units by 2025 globally. This large proliferation of IoT devices has enabled a diverse array of applications across multiple domains [2], from healthcare and smart homes to manufacturing and logistics, enabling a seamless transfer of data between devices and services. However, this growth has also led to new security challenges [3], as these devices are often resource-constrained, operate in heterogeneous environments, and are deployed in physically insecure locations.

To detect and mitigate cyberattacks, Intrusion Detection Systems (IDSs) [4], Malware Detection Systems (MDSs) [5], and Device Identification Systems (DISs) [6] are often employed to monitor IoT network traffic and detect malicious activities [7–9]. ML [10,11] techniques, including Deep Learning (DL) [12,13], have shown promise in enhancing the effectiveness of these systems, by leveraging the ability of ML algorithms to learn from data and identify patterns that indicate anomalous behavior.

Nonetheless, the application of ML techniques within IDSs, MDSs, and DISs introduces new vulnerabilities. Attackers can potentially manipulate or bypass these systems by exploiting the inherent nature of ML models, which involves learning and recognizing

patterns. Adversarial machine learning attacks are a particular concern. Those attacks on ML-based security systems involve injecting malicious input data called Adversarial Examples to cause misclassification or bias or modify the ML model to produce incorrect results. As illustrated in Figure 1, adversarial samples are designed by intentionally introducing a small perturbation to the initial inputs, to mislead the ML model into generating an incorrect prediction [14,15].

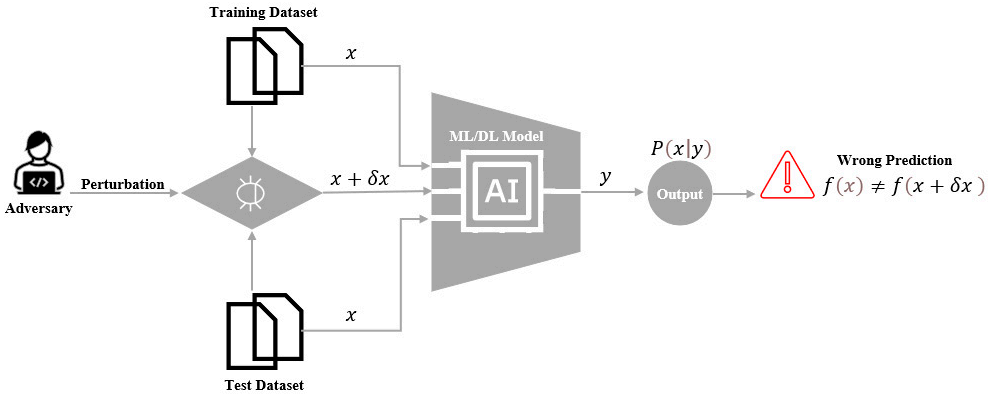

Figure 1. Generic process of adversarial attack.

Numerous surveys have been published that explore how adversarial attacks affect the performance of ML-based systems in diverse domains, including, but not limited to, computer vision [16–19], natural language processing [20,21], and speech recognition [22]. The majority of existing surveys are related to adversarial attacks against ML in the domain of computer vision [16–18] and traditional network security [23,24]. However, these attacks have received less attention in the field of IoT network security. Figure 2a illustrates the growing focus of the research community on adversarial attacks. In contrast, Figure 2b highlights the low number of published research in the context of IoT ML-based security.

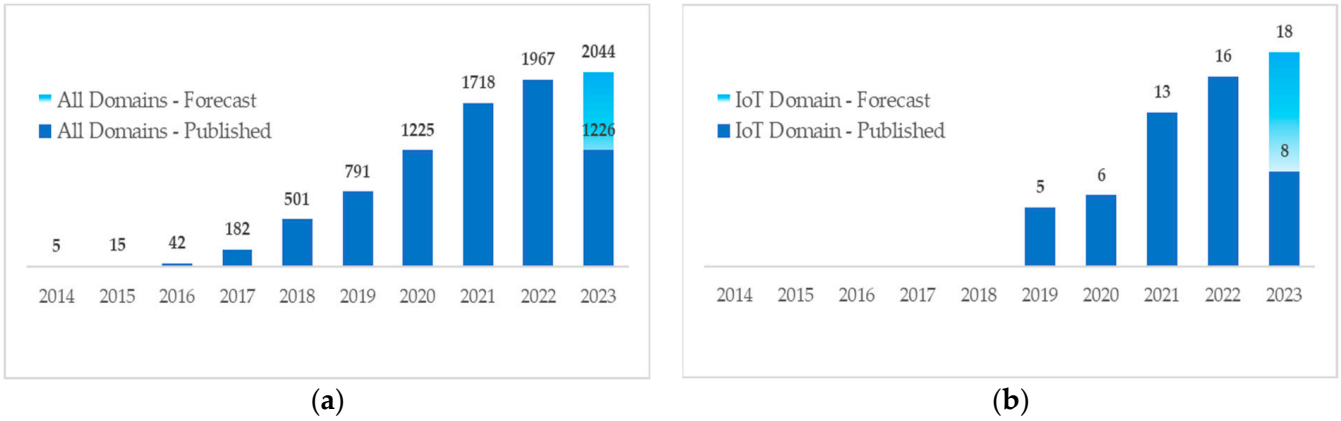

Figure 2. Total number of papers related to Adversarial Attacks published in recent years: (**a**) In all domains; (**b**) In the IoT domain only. The row data source is from [25] and it is completed based on our research findings in the IoT domain from 2019 to July 2023. The forecast was projected through quadratic curve modeling.

In the field of traditional network security, the authors of [24] presented a survey of the current research landscape regarding the ML vulnerability to adversarial attacks. The survey reviewed different varieties of adversarial attacks encompassing evasion attacks and poisoning attacks and discussed their impact on various traditional network security ML-based models such as IDSs and MDSs. The study also outlined various defensive mechanisms that have been suggested to minimize the effects of adversarial attacks. How-

ever, the survey's main focus was on traditional network security, while the security of IoT networks was very briefly discussed in a very short paragraph with a unique reference in the IoT context literature. Jmila, H et al. [23] provided a comparative study of ML-based IDS vulnerability to adversarial attacks and paid more attention to the so-called shallow models (non-deep learning models). The authors assessed the resilience of seven shallow ML-based and one Deep Neural Network (DNN), against a variety of adversarial attacks commonly employed in state-of-the-art datasets using NSL-KDD [26] and UNSW-NB15 [27]. The survey paid minimal attention to adversarial attacks in the field of IoT security, offering only four references without any accompanying discussion. Alatwi et al. [28] discussed adversarial black-box attacks against IDS and provided a survey of recent research on traditional network security and Software-defined Networking (SDN). Within its scope, the survey focused solely on reviewing research studies that employed adversarial generation attacks using different variants of Generative Adversarial Networks (GAN). Meanwhile, it overlooked the most widely used adversarial attack methods and defense strategies. Furthermore, limiting this survey to the black-box attacks was of interest, as it closely aligns with the most realistic circumstances for the adversary. However, studying the white-box attacks could be more interesting and beneficial for IDS's manufacturers who have complete access to their system and seek to assess its resilience against adversarial attacks, as well as in the scenario of insider attacks [29,30], where the attackers can have access to sensitive resources and system information, the protection against white-box attacks can be more challenging.

In the IoT network context, only a handful of published surveys have discussed adversarial attacks against ML-based security systems. For instance, in the survey in [30], the authors' primary focus was to review and categorize the existing body of information on adversarial attacks and defense techniques in IoT scholarly articles, with a unique emphasis on insider adversarial attacks. The authors presented a taxonomy of adversarial attacks, from an internal perspective, targeting ML-based systems in an IoT context. Additionally, they offered real-world application examples to illustrate this concept. The article also discussed defensive measures that can be used to resist these kinds of attacks in IoT. However, the external (black-box) adversarial attacks, which represent a realistic scenario, are not discussed, hence the Model Extraction attacks were not covered in the survey as the insider adversary usually has full knowledge of the ML model. In [31], the authors surveyed existing IDSs used for securing IoT-based smart environments such as Network Intrusion Detection Systems (NIDS) and Hybrid Intrusion Detection Systems (HIDS). They provided benefits and drawbacks of diverse anomaly-based intrusion detection methods, such as signal processing model, protocol model, payload model, rule-based model, machine learning, and others, where machine learning techniques require a brief overview without discussing the vulnerability of those ML-based systems to adversarial attacks. The study in [32] presented a thorough examination of ML-based attacks on IoT networks, offering a classification of these attacks based on the employed ML algorithm. The authors sought to explore a range of cyberattacks that integrated machine learning algorithms. However, adversarial attacks received only a brief discussion as one category of ML-based attacks, with mention of three adversarial attacks: the Jacobian-based Saliency Map Attack (JSMA), DeepFool, and the Carlini and Wagner (C&W) attack, as well as defense methods but they lack in-depth discussion. In [33], Li et al. surveyed adversarial threats that exist within the context of Cyber-Physical Systems (CPS). CPS is a subset of IoT, where the connection between cyberspace and physical space is provided by actuators and sensors. As a result, the work presented in [33] was limited to sensor-based threats only, which are a subset of network-based and side-channel attacks in the attack taxonomy of IoT networks. He et al. [34] explored the disparity in adversarial learning within the fields of Network Intrusion Detection Systems (NIDS) and Computer Vision. They accomplished this by reviewing the literature on adversarial attacks and defenses against IDS, with a special focus on IDS in traditional networks. The authors limited their study to evasion attacks only, considering that NIDS are typically created in secure environments, in which case the

external attackers lack access to the training data set. Furthermore, the authors provided a taxonomy related to NIDS and not to adversarial attacks themselves.

In light of the information presented above and summarized in Table 1, there is a notable scarcity of published surveys specifically addressing adversarial attacks against ML-based security systems in IoT networks. The limited number of existing surveys tend to have a narrow focus on the issue, with some solely concentrating on ML-based IDSs, while disregarding the wider scope, which encompasses ML-based MDSs and ML-based DISs. Also, some have been focusing primarily on insider threats while neglecting external ones. Additionally, certain surveys exclusively examine black-box attacks, overlooking white-box attacks.

To bridge these gaps, this survey offers a comprehensive review of the current research landscape regarding adversarial attacks on IoT networks, with a special emphasis on exploring the vulnerabilities of ML-based IDSs, MDSs, and DISs. The survey also describes and classifies various adversarial attack generation methods and adversarial defense methods.

To the best of our knowledge, this survey will be the first attempt of its kind to comprehensively discuss the holistic view of adversarial attacks against ML-based IDSs, MDSs, and DISs in the context of IoT, making a significant contribution to the field. This paper's contributions are outlined as follows:

1. Revising and redefining the adversarial attack taxonomy for ML-based IDS, MDS, and DIS in the IoT context.
2. Proposing a novel two-dimensional-based classification of adversarial attack generation methods.
3. Proposing a novel two-dimensional-based classification of adversarial defense mechanisms.
4. Providing intriguing insights and technical specifics on state-of-the-art adversarial attack methods and defense mechanisms.
5. Conducting a holistic review of the recent literature on adversarial attacks within three prominent IoT security systems: IDSs, MDSs, and DISs.

The rest of this paper is organized as follows: Section 2 gives background about IoT network architecture and its privacy and security perspective. Section 3 redefines the threat model taxonomy in the IoT network context. Section 4 gives an overview of the most popular adversarial attack generation methods. Section 5 elaborates on the existing adversarial defense methods. Section 6 discusses the recent studies related to adversarial attacks against ML-based security systems in IoT networks. Section 7 ends the paper with challenges and directions for future works, and Section 8 concludes the paper.

Table 1. Summary comparison of related surveys.

Ref.	Year	Network	Major Contribution(s)	Limitation(s)	Attacker's Knowledge		Security Systems			Adversarial Attack Taxonomy	Adversarial Attack Methods	Adversarial Defense Methods
					White-Box	Black-Box	IDS	MDS	DIS			
[23]	2022	Traditional	Robustness evaluation of seven shallow ML-based IDS against adversarial attacks.	IoT network security is just mentioned in four references with no discussion. Only three adversarial defense techniques were mentioned.	✓	✓	✓	×	×	✓	✓	×
[24]	2019	Traditional	Evaluation of different adversarial attacks to ML models applied in computer and traditional network security. Classification of adversarial attacks based on security applications. Risk identification using adversarial risk grid map.	Mainly focused on traditional network security while IoT network security was very briefly discussed in a very short paragraph.	✓	✓	✓	✓	×	✓	✓	✓
[28]	2021	Traditional	Summarize recent research on black-box adversarial attacks against NIDS.	Focused on black-box attacks only. Most popular adversarial attack methods and defense methods were not discussed	×	✓	✓	×	×	×	×	×
[30]	2022	IoT	Taxonomy of adversarial attacks from insider (internal) perspective. Real-life applications of adversarial insider threats.	Focused on insider (white-box) adversarial attacks only. Model Extraction attacks were not covered as the survey is limited to insider adversarial threats where the adversary has full knowledge of the ML model	✓	×	×	✓	×	✓	✓	✓
[31]	2018	IoT	Reviewed the existing IDSs used for securing IoT-based smart environments such as Network Intrusion Detection Systems (NIDS) and Hybrid Intrusion Detection Systems (HIDS).	The vulnerability of ML-based IDSs to adversarial attacks was not covered.	×	×	✓	×	×	×	×	×
[32]	2022	IoT	Overview of existing ML-based attacks in IoT network. Classification of ML-based attacks based on the type of the used ML algorithm.	Adversarial attacks were briefly discussed as one type of various ML-based attacks in IoT networks. The authors mentioned some adversarial attacks and defense methods with no discussion.	✓	✓	✓	×	×	×	×	×

Table 1. *Cont.*

Ref.	Year	Network	Major Contribution(s)	Limitation(s)	Attacker's Knowledge		Security Systems			Adversarial Attack Taxonomy	Adversarial Attack Methods	Adversarial Defense Methods
					White-Box	Black-Box	IDS	MDS	DIS			
[33]	2020	CPS	Surveyed adversarial threats within the context of Cyber-Physical Systems (CPS).	Considered only adversarial attacks that exploit sensors in IoT and CPS devices. Limited to sensor-based threats only	✓	✓	×	×	×	×	×	✓
[35]	2022	Traditional	Adversarial attacks on malware detection systems Adversarial malware evasion threat modeling.	They were focused on the computer and cybersecurity domain, while the IoT network security domain was overlooked.	✓	✓	×	✓	×	✓	✓	×
[36]	2023	Traditional	Highlighting various types of adversarial attacks against IDS in the context of traditional networks.	IoT network security context was not included. Model Extraction attacks were not covered.	✓	✓	✓	×	×	✓	✓	✓
[34]	2023	Traditional	Explored the disparity in adversarial learning within the fields of Network Intrusion Detection Systems (NIDS) and Computer Vision specifically focusing on DL-based NIDS in traditional network.	Mainly focused on traditional network security while IoT network security was very little discussed. Poisoning and model extraction attacks are not covered.	✓	✓	✓	×	×	×	✓	✓
Our Work	2023	IoT	Holistic review of ML adversarial attacks in three prominent IoT security systems: IDSs, MDSs, and DISs. Re-defining taxonomy of threat methods in IoT context. 2D classification of both adversarial attacks and defense methods.		✓	✓	✓	✓	✓	✓	✓	✓

2. Background

2.1. Security and Privacy Overview

In the last twenty years, the potential applications of IoT have been steadily multiplying across various sectors paving the way for new business prospects [2,37,38]. Yet, the emergence of IoT has simultaneously presented manufacturers and consumers with new challenges [2,3,39]. One of the principal challenges lies in safeguarding the security and privacy of both the IoT objects and the data they produce. Ensuring the security of IoT networks is a complicated and arduous task due to the inherent intricacies within the IoT network characterized by the interconnection of multiple heterogeneous devices from different locations and exchanging information with each other through various network technologies. As a result, IoT systems are notably vulnerable to privacy and security threats.

Before delving into those security threats in the IoT landscape, it is pivotal to explore its security and privacy features. Overlooking these security measures can introduce vulnerabilities into the framework. Through a thorough review of the literature on IoT security [40–43], these features have been pinpointed. Figure 3 encapsulates the key security and privacy features of the IoT infrastructure.

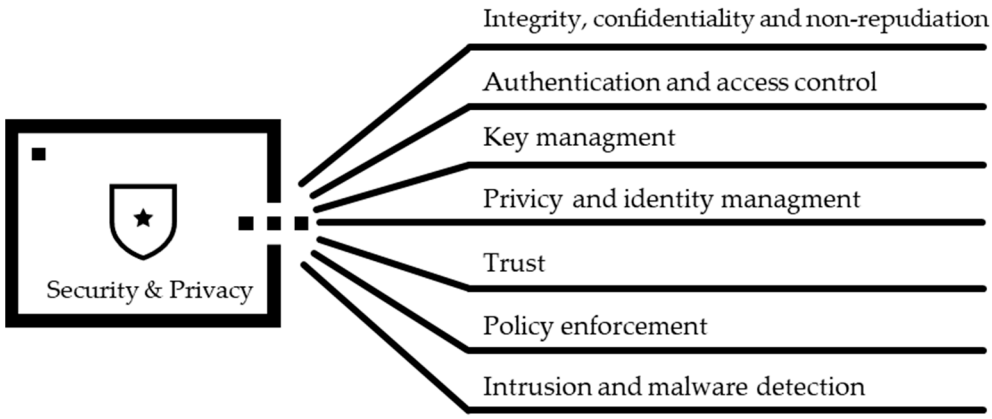

Figure 3. Key security and privacy features of IoT network.

Traditional security methods, which employ a predefined set of strategies and rules, have exhibited several drawbacks when implementing specific features. They often overlook new varieties of attacks and are restricted to pinpointing certain types of threats. Hence, the emergence of advanced security solutions such as solutions powered by artificial intelligence. The utilization of ML algorithms has the potential to offer security solutions for IoT networks, ultimately improving their reliability and accessibility. ML-based security models can process large amounts of data in real time and continuously learn from generated training and test data, which increases their accuracy as well as enables them to proactively anticipate new attacks by drawing insights from previous incidents. Our survey will limit the study to contemporary research on the vulnerability of three ML-based IoT security systems: Intrusion Detection System (IDS), Malware Detection System (MDS), and Device Identification System (DIS).

2.2. Internet of Things Overview

The IoT is one of the cutting-edge technologies in Industry 4.0, where the term "Things" refers to smart devices or objects interconnected through wireless networks [44,45]. These "Things" range from everyday household objects to advanced industrial instruments capable of sensing, gathering, transmitting, and analyzing data. Such capabilities facilitate smart decision-making and services enhancing both human life quality and industrial production.

At present, there is no agreed-upon structure for IoT architecture. The fundamental framework of IoT comprises three layers: the perception layer, the network layer, and the application layer [46]. Yet, based on the requirements for data processing and making

intelligent decisions, a support or middleware layer, positioned between the network and application layers, was later deemed to be essential [47]. Different technologies are utilized within each of these layers, introducing various challenges and security concerns [2,48]. Figure 4 shows the four-layered IoT architecture showing various devices, technologies, and applications along with possible security threats at each layer.

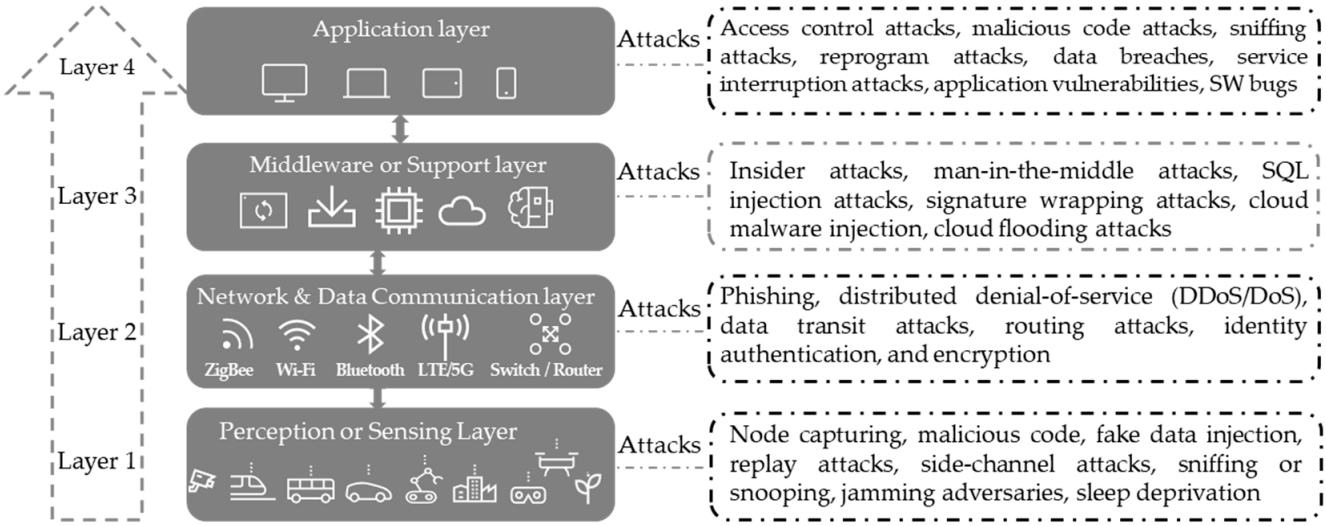

Figure 4. Four-layered IoT architecture and corresponding security issues.

- Perception layer: The bottom layer of any IoT framework involves "things" or endpoint objects that serve as the bridge between the physical and the digital worlds. The perception or sensing layer refers to the physical layer, encompassing sensors and actuators capable of gathering information from the real environment and transmitting it through wireless or wired connections. This layer can be vulnerable to security threats such as insertion of fake data, node capturing, malicious code, side-channel attacks, jamming attacks, sniffing or snooping, replay attacks, and sleep deprivation attacks.
- Network layer: It is known as the second layer connecting the perception layer and middleware layer. It is also called the communication layer because it acts as a communication bridge, enabling the transfer of data acquired in the perception layer to other interconnected devices or a processing unit, conversely. This transmission utilizes various network technologies like LTE, 5G, Wi-Fi, infrared, etc. The data transfer is executed securely, ensuring the confidentiality of the obtained information. Nonetheless, persistent security vulnerabilities can manifest as data transit attacks, phishing, identity authentication, and encryption attacks, and distributed denial-of-service (DDoS/DoS) attacks.
- Middleware layer: It is also commonly known as the support layer or processing layer. It is the brain of the IoT ecosystem, and its primary functions are data processing, storage, and intelligent decision-making. The middleware layer is the best candidate to implement advanced IoT security mechanisms, such as ML-based security systems, thanks to its high computation capacity. Therefore, it is also a target of adversarial attacks and other various attacks such as SQL injection attacks, cloud malware injection, insider attacks, signature wrapping attacks, man-in-the-middle attacks, and cloud flooding attacks.
- Application layer: It is the uppermost layer within the IoT architecture. It serves as the user interface to monitor IoT devices and observe data through various application services and tools, such as dashboards and mobile applications, as well as applying various control activities by the end user. There are various use cases for IoT applications such as smart homes and cities, smart logistics and transportation, and smart agriculture and manufacturing. This layer is also subject to various security threats

such as sniffing attacks, service interruption attacks, malicious code attacks, reprogramming attacks, access control attacks, data breaches, application vulnerabilities, and software bugs.

3. Adversarial Attack Taxonomy

Threat modeling is a classification process used in information security and risk management to identify potential threats, vulnerabilities, and associated risks. This classification approach is used in many research fields such as traditional network security [23,24], intelligent networks [49], and IoT networks [30]. A threat taxonomy groups threats into hierarchical classes based on common characteristics. This helps determine the best approach for detecting and mitigating the threat. A variety of attacks require diverse approaches depending on the nature of the attack and the specificities of the system being targeted.

In the study [23], the authors classified adversarial attacks in network security in two dimensions only, knowledge and goal. This classification is very short, simplified, and does not reflect other characteristics of adversarial attacks. The taxonomy proposed in [24] is an extensive classification where in addition to the common classes, the authors added two more classes, space and target. The space class includes feature space and problem space sub-classes where feature space attack aims to modify or alter the features without generating new instance, while problem space attack attends to modify the actual instance itself to create an entirely new sample. This classification is not applicable in the context of IoT networks in which the feature mapping is not invertible or not differentiable due to inherent constraints of IoT network traffic. Furthermore, IoT traffic features can be binary, categorical, or continuous. Moreover, the values of these features are closely correlated, with some being constant and others being unalterable. Hence this classification is applicable to unconstrained domains like computer vision, where the main feature is the image's pixels. Moreover, the target class given by this study [24] in which they classified the threat between the physical domain target and ML model target is against the inherent nature of adversarial attacks to fool ML Models.

Inspired by the adversarial attacks taxonomy framework proposed in [30,49], we re-defined the adversarial attacks taxonomy based on four main classifications; the attacker's knowledge, the attack goal, the attacker's capability, and the attacker's strategy as summarized in Figure 5. Our taxonomy is tailored towards including other adversarial attack characteristics and IoT security system specificities that were not in the scope of the studies [30,47]. The study in [30] was limited to insider attacks and white-box attacks, where the adversary has full knowledge of ML models and data. Hence, the characteristics of a black-box attack were not considered. In contrast, the study in [47] was limited to poisoning attacks only, where the adversary adds malicious data during the training phase. Hence, adversarial attacks during the testing and deployment phases were not considered.

Hence, our proposed taxonomy framework is a tailored approach to classify adversarial attacks according to their common characteristics and consider the specificities of ML-based IoT security systems. This will help researchers and practitioners to better understand the potential risks, identify relevant vulnerabilities, and set feasible security objectives.

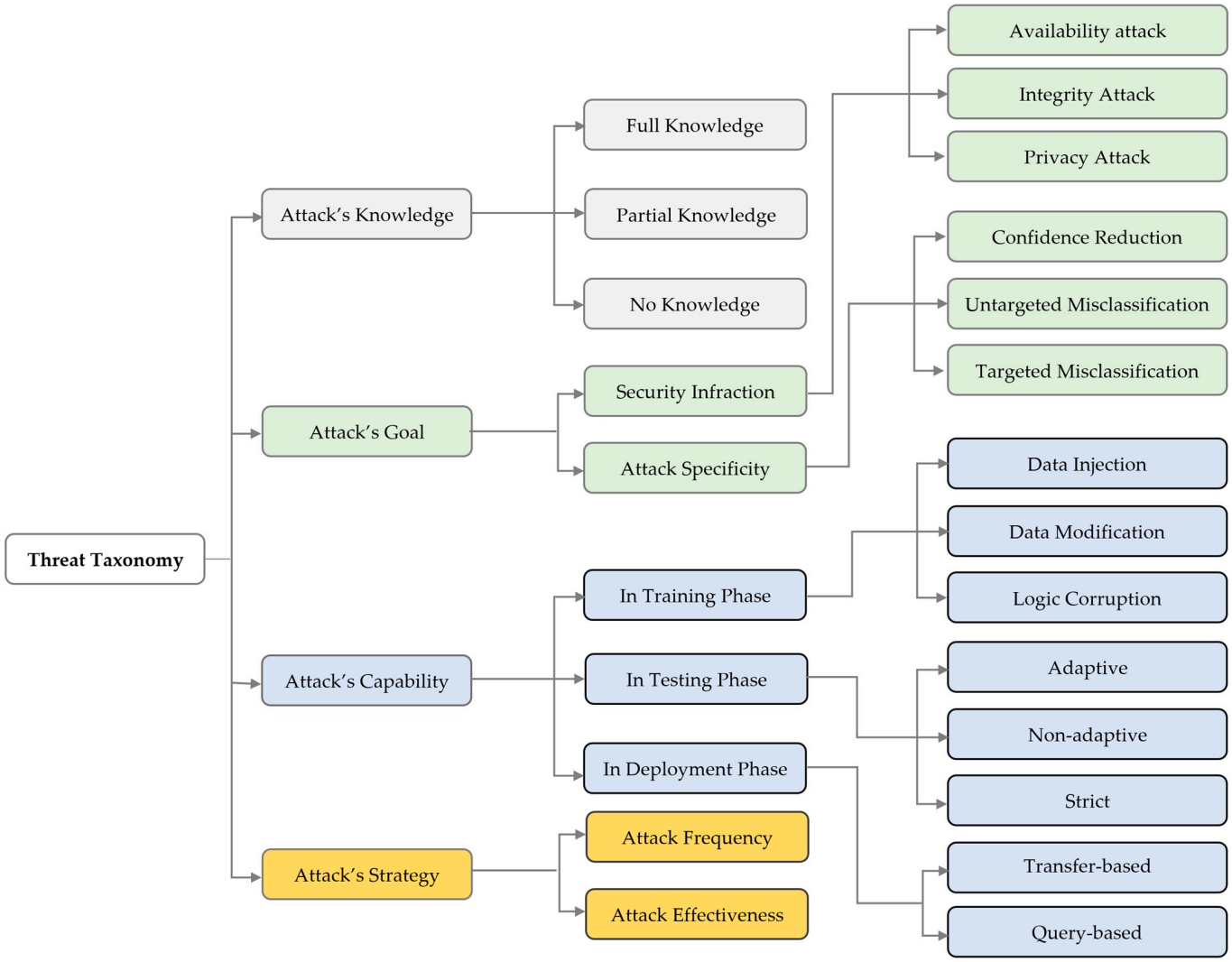

Figure 5. Adversarial attack taxonomy.

3.1. Attacker's Knowledge

One of the dimensions of threat model classification is the level of information and knowledge accessible to adversaries concerning the ML model. Attack knowledge can be classified according to the following levels:

- Full knowledge: This refers to white-box attacks, where the attacker possesses complete awareness of the target ML system's information. This means that the adversary possesses complete and unrestricted access to the training dataset, ML model architecture, and its hyper-parameters as well as the feature learning. This is generally not feasible in most real adversarial attacks. However, the purpose of studying them is to assess the vulnerability of the target ML system to all possible cases and scenarios.
- Partial knowledge: Referring to gray-box attacks, where the attacker possesses partial information of the target ML system's inner workings. This means that the adversary may have limited access to the feature representations, training dataset, and learning algorithm's parameters. Using partial information, the attacker can create a practical strategy to deceive the ML model.
- No knowledge: This corresponds to black-box attacks, where the attacker is entirely unaware of the architecture and parameters of the target model. The adversary relies solely on his capability to query the target ML system by inputting the chosen data and monitoring corresponding results. These attacks are considered the most practical

because they operate under the assumption that the attacker can only leverage system interfaces that are readily accessible for typical use.

3.2. Attacker's Goal

The attacker's objective is to influence the outcomes of the ML system either by misleading the system or by introducing perturbations to the input. The attacker's goal can be then outlined as follows:

- Security Infraction: Refers to security violations and can be classified into three main dimensions.
- Availability Attack: The attacker intends to minimize the model's performance at testing or deployment phases, thereby making it unreliable and useless. Availability attacks can be executed through data poisoning when the attacker gains control over a portion of the training dataset, or through model extraction when the attacker predicts some relevant parameters of the target model.
- Integrity Attack: Focuses on undermining the integrity of an ML model's output, leading to erroneous predictions made by the model. The attacker can induce an integrity breach by executing an evasion attack during the testing or deployment phases or a poisoning attack during the training phase.
- Privacy Attack: The attacker's objective could involve gaining information about the system data, leading to data privacy attacks, or about the ML model, resulting in model privacy attacks.
- Attack Specificity: Based on their impact on the model output integrity, the attack specificity can be divided into three distinct categories:
- Confidence Reduction: The adversary intends to decrease the prediction certainty of the target model.
- Untargeted Misclassification: The adversary endeavors to change the predicted classification of an input instance to any class other than the original one.
- Targeted Misclassification: The adversary seeks to generate inputs that compel the classification model's output to become a particular desired target class or endeavors to make the classification output for a specific input correspond to a specific target class.

3.3. Attacker's Capability

Illustrates the impact of the adversary on the target ML system's operation. The efficiency of an adversarial attack is determined by the capability and strategy to manipulate the classes and features of the training data or test data gathered from various IoT networks. It is influenced by factors such as the quantity of malicious data introduced or altered and the specific portion of the training or testing data that the attacker targets. The categorization of attacks on ML models varies according to the stages within the ML model pipeline: training phase, testing phase, and deployment phase.

- Training phase: In this phase, attacks on the ML model are more frequent than often realized. The attacker aims to mislead or disrupt the model's outcomes by directly modifying the training dataset. Those kinds of attacks are known as "poisoning" or "contaminating", and they require that an adversary has a degree of control over training data. The attacker's tactics during the training phase are shaped by their adversarial capabilities which can be classified into three distinct categories.
- Data Injection: The attacker lacks access to the learning model's parameters and training dataset, yet possesses the capability to append new data to the training dataset, thereby inserting adversarial samples to fool or degrade the ML model's performance.
- Data Modification: The adversary cannot access the learning algorithms but can manipulate the training data, contaminating it before it is used to train the target model.
- Logic Corruption: The adversary can tamper with the learning algorithm of the target ML model. In other words, the learning algorithm is susceptible to interference from the opponent.

- Testing phase: In testing, adversarial attacks do not alter the training data or directly interfere with the model. Instead, they seek to make the model produce incorrect results by maliciously modifying input data. In addition to the level of information at the adversary's disposal and, the attacker's knowledge, the efficacy of these attacks depends on three main capabilities: adaptive attack, non-adaptive attack, and strict attack.
- Adaptive Attack: The adversary is crafting an adaptive malicious input that exploits the weak points of the ML model to mistakenly classify the malicious samples as benign. The adaptiveness can be achieved either by meticulously designing a sequence of input queries and observing their outputs in a black-box scenario or through accessing the ML model information and altering adversarial example methods that maximize the error rate in case of a white-box scenario.
- Non-adaptive attack: The adversary's access is restricted solely to the training data distribution of the target model. The attacker starts by building a local model, choosing a suitable training procedure, and training it using samples from data distribution to mimic the target classifier's learned model. Leveraging this local model, the adversary creates adversarial examples and subsequently applies these manipulated inputs against the target model to induce misclassifications.
- Strick Attack: The attacker lacks access to the training dataset and is unable to dynamically alter the input request to monitor the model's response. If the attacker attempts to request valid input samples and introduces slight perturbations to observe the output label, this activity most probably will be flagged by the target ML model as a malicious attack. Hence, the attacker is constrained to perform a restricted number of closely observed queries, presuming that the target ML system will only detect the malicious attacks after a specific number of attempts.
- Deployment phase: Adversarial attacks during the deployment or production phase represent the most realistic scenario where the attacker's knowledge of the target model is limited to its outputs, which correspond to a black-box scenario. Hence, the attack's success during deployment time relies on two main capabilities, the presumption of transferability or the feedback to inquiries. Consequently, the attacker's capability during the deployment phase can be categorized into two distinct groups, namely transfer-based attack and query-based attack.
- Transfer-based Attack: The fundamental concept underlying transfer-based attack revolves around the creation of adversarial examples on local surrogate models in such a way that these adversarial examples can effectively deceive the remote target model as well. The transferability propriety encompasses two types: task-specific transferability which applies to scenarios where both the remote victim model and the local model are concerned with the same task, for instance, classification. Cross-task transferability arises when the remote victim model and the local model are engaged in diverse tasks, such as classification and detection.
- Query-based Attack: The core idea behind query-based attacks lies in the direct querying of the target model and leveraging the outputs to optimize adversarial samples. To do this, the attacker queries the target model's output by providing inputs and observing the corresponding results, which can take the form of class labels or score values. Consequently, query-based attacks can be further categorized into two distinct types: decision-based and score-based.

3.4. Attacker's Strategy

Assuming different levels of knowledge available to the attacker, the adversary's strategy manifests as the optimal quantitative and qualitative choice of adversarial attack that achieves the optimum effect of the attacker's goal. Therefore, the attack strategy can be categorized into attack effectiveness and attack frequency.

- Attack effectiveness: It can be elaborated by the way to inject a bias in the input data to maximize the efficiency of the attack. In other words, it is nothing more than an

optimization problem aimed at maximizing the loss function of the target ML algorithm on a validation dataset or to minimize its loss function on a poisoned dataset.

- Attack frequency: Refers to the decision between a one-time attack and an iterative process that updates the attack multiple times to enhance its optimization. While iterative attacks often outperform their one-time counterparts, they come with the trade-off of increased computational time and the chance of being detected by the ML-based security system. In certain situations, opting for a one-time attack may be adequate or the only practical option available.

4. Adversarial Attack Generation Methods for IoT Networks

Adversarial attacks have been extensively studied in various domains, in contrast to the relatively limited attention they have received in the domain of IoT security, as shown in above Figure 2. The techniques for generating adversarial attacks vary depending on the nature of the data in the applied field. Hence the use of adversarial attack techniques in the IoT security context may differ significantly from its conventional use in other domains such as computer vision, for the simple reason that images and traffic data have different attributes that affect their suitability for machine learning input. An image file is formed by many pixels with the same attribute and every pixel consists of three values, representing three distinct colors: red, green, and blue. The data related to IoT traffic consists of various features, each representing specific physical meanings that are interconnected. In contrast to images, where minor adversarial perturbations in pixel color values generally manifest as only marginal overall effects, the alteration of specific pivotal features within IoT traffic data may culminate in the forfeiture of vital information. Consequently, this undermines the intrinsic behavioral robustness against malicious attacks.

Adversarial attack methods can be classified into three distinct groups: exploratory attack methods, causative attack, and inference attack, depending on the stage where the attack can be launched. They can additionally be classified according to the attacker's knowledge. Figure 6 summarizes the different adversarial attack generation methods in two-dimensional (2D) classification.

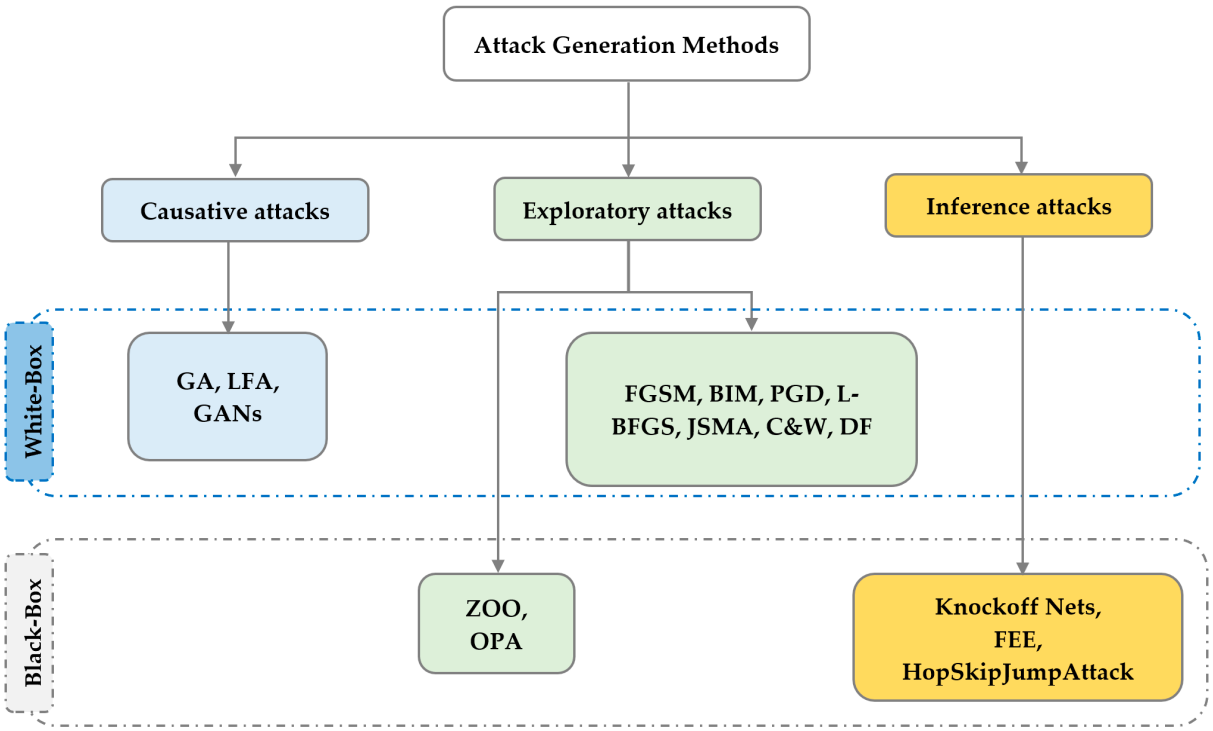

Figure 6. Classification of adversarial attack generation methods.

4.1. Exploratory Attack Methods

Those attacks, also called evasion attacks, are adversarial attacks launched during the test phase. In the exploratory attack, the adversary tries to deceive the ML model by modifying its input data in a manner that induces the model to incorrectly classify the input. In other words, the attacker aims to evade the model detection by crafting a malicious input that is incorrectly classified as benign. Because they occur during the test phase, these attacks are the most feasible and frequently employed against intrusion and malware detection systems. Exploratory attacks can manifest in two forms, white-box attacks, in which the attacker possesses information about the training data or learning algorithms, or black-box attacks, where the attacker lacks knowledge of the training data and learning algorithms and relies solely on observations of the model's input-output behavior to generate adversarial examples. The most popular exploratory attack methods used against ML-based systems in the context of IoT networks will be discussed in the next subsubsections.

4.1.1. Fast Gradient Sign Method

Fast Gradient Sign Method (FGSM) is a straightforward and efficient method for generating adversarial examples (AEs) [15]. Those AEs are inputs that have been intentionally modified in a way that optimizes the maximum quantity of perturbation applied to each pixel (i.e., image) to induce incorrect predictions by an ML model. The FGSM works by taking the gradient of the loss function relative to the input data and subsequently perturbing the input data in the direction of the sign of the gradient. The magnitude of the perturbation is established by a hyperparameter known as epsilon (ε), which controls how much the input data are modified. The output result is called the AE and its formula can be formalized by the Expression (1):

$$X_{adversarial} = X + \varepsilon.Sign(\nabla x\ J(\theta, X, Y)) \quad (1)$$

where ε represents a small value and ∇ denotes the gradient of loss function J relative to the original input data (i.e, image) X, the original input class label Y, and the model parameters θ.

The FGSM algorithm can be summarized in three steps. The first step computes the gradient of the loss relative to the inputs, the second step scales the gradient to have a maximum magnitude of ε, and the third step adds the scaled gradient to the input data (i.e., image) X to create the adversarial example $X_{adversarial}$.

Although this method is fast for generating AEs, its effectiveness is lower than that of other state-of-the-art methods for generating adversarial attacks because it generates only one AE per input data point and may not be able to explore the full space of possible AEs. Additionally, being a white-box attack is that it assumes full knowledge of the targeted model. This requirement limits its applicability in scenarios where the adversary possesses restricted access to the model's internal details, but it remains useful for manufacturers to assess the resilience of their ML models against adversarial attacks as well as in scenarios of insider attacks [36].

4.1.2. Basic Iteration Method

Proposed by Kurakin et al. in 2017 [50], the Basic Iteration Method (BIM) represents a basic extension of the FGSM, where instead of making a single large step, it adopts an iterative approach by applying FGSM multiple times to an input with small step-size perturbations in the direction that maximizes the model's loss. The goal is to generate an AE that appears similar to the original input but can mislead the model's predictions.

The basic idea behind the method is to start with an initial estimation of the solution and then iteratively improve the estimation by applying the Gradient Descent (GD) to the current guess. The resulting adversarial sample is then clipped to limit the maximum

perturbance for each pixel. The formula can be summarized by the following Expression (2).

$$X_0^{adv} = X, \qquad X_{N+1}^{adv} = Clip_{X+\varepsilon}\left\{X_N^{adv} + \alpha.Sign\left(\nabla x\, J\left(X_N^{adv}, Y\right)\right)\right\} \tag{2}$$

where J denotes the loss function, X is the original input data (i.e., image), Y is the original input class label, N denotes the iteration count and α is the constant that controls the magnitude of the disturbance. The *Clip* {} function guarantees that the crafted AE remains within the space of both the ε ball (i.e., $[x - \varepsilon, x + \varepsilon]$) and the input space.

The BIM algorithm involves starting with clean data (i.e., image) as the initial input. The gradient of the loss function is computed relative to the input, and a small perturbation is added along the gradient direction, scaled by a defined step size. The perturbed input is then clipped to ensure it stays within a valid range. These steps are iterated until a desired condition is met or for a set number of iterations.

Although this method is simple to generate AEs, it might demand an extensive series of iterations to find the most effective and optimal AEs, and this may be computationally expensive and may not converge for all functions or initial assumptions.

4.1.3. Projected Gradient Descent

Projected Gradient Descent (PGD) extends the idea of BIM by incorporating projection onto a feasible region or constraint set. Proposed by Madry et al. in 2018 [51], PGD is an optimization method that is used to identify the minimum of a function that is subjected to constraints. In the context of adversarial attacks, the feasible region often corresponds to a set of allowed perturbations that respect certain constraints, such as a maximum perturbation magnitude or spatial constraints.

The algorithm works by iteratively taking steps following the negative gradient direction of the function, but with an added step of projecting the new point onto the feasible region defined by the constraints. This ensures that the solution found by the algorithm always satisfies the constraints. The formula can be summarized by the Expression (3).

$$\prod_{C_\varepsilon}(X') = Argmin_{z \in C_\varepsilon}\|Z - X'\|, \quad X_{N+1}^{adv} = \prod_{C_\varepsilon}\left\{X_N^{adv} + \alpha.Sign\left(\nabla x\, J\left(X_N^{adv}, Y\right)\right)\right\} \tag{3}$$

here C_ε is constraint set where $C_\varepsilon = \{z{:}d(x,z) < \varepsilon\}$, $\prod_{C_\varepsilon}$ denotes projection onto the set C_ε, and α is the step size. For example, the projection $\prod_{C_\varepsilon}(z)$ *for* $d(x,z) = \|x - z\|_\infty$ is given by clipping z to $[x - \varepsilon, x + \varepsilon]$. J dedenotes the loss function of the model, X is the original input data (i.e., image), Y is the original input class label, N denotes the iteration count and α is constant to regulate the perturbation magnitude.

PGD ensures that the solution falls within the feasible space, making it suitable for solving constrained optimization problems. However, the projection step can be computationally expensive, particularly for complex constraint sets.

4.1.4. Limited-Memory BFGS

The Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method is a non-linear gradient-based optimization algorithm employed to minimize the quantity of perturbations introduced into images. It is a white-box adversarial attack introduced by Szegedy et al. [14] and it differs from the FGSM in two key aspects: the Distance Metric aspect and the Precision versus Speed aspect.

In terms of the distance metric, the L-BFGS attack is optimized for the L_2 distance metric, whereas the FGSM is designed for the L_∞ (infinity) distance metric. However, from the precision versus speed metric, the FGSM is known for its computational efficiency but may not always produce AEs that are visually imperceptible from the original data. The L-BFGS attack is formulated to generate AEs exceedingly similar to original inputs, but this quest for accuracy often results in heightened computational time as a trade-off.

By formalizing the optimization problem depicted in Equation (4), where the primary aim is to minimize the perturbations r introduced to the original input (i.e., image) while considering the L_2 distance.

$$Arg\ min_r f\ (X + r) = l \qquad s.t.\ (X + r) \in D \tag{4}$$

here, X denotes the original input data (i.e., image), r is the perturbation simple within the input domain D, f is the classifier's loss function and l is the incorrect predicted label ($l \neq h(X)$) of the adversarial example $X' = X + r$.

By optimizing for the L_2 distance and prioritizing precision over speed, the L-BFGS attack aims to generate perturbations that result in small changes across all dimensions of the input, rather than focusing on maximizing the change in a single dimension. Hence, this method excels in generating AEs, yet its feasibility is limited by a computationally demanding algorithm to explore an optimal solution.

4.1.5. Jacobian-Based Saliency Map Attack

The Jacobian-based Saliency Map Attack (JSMA) is a saliency-based white-box adversarial attack method. It was proposed by Papernot et al. [52] to generate AEs capable of deceiving the Deep Neural Networks (DNNs) by using the Jacobian matrix to identify the most influential input characteristics that lead to a substantial change in the DNNs output.

Unlike FGSM, JSMA aims to reduce the perturbations by controlling the number of features to be perturbated instead of the magnitude or quality of the perturbation. The goal then is to manipulate only a small number of pixels within the image, rather than disturbing the entire image, and monitoring the effects on the output classification. The observation is conducted through the computation of a saliency map using the gradient output of the network layer. Once the saliency map is calculated, the algorithm systematically identifies the pixel within an image that would have the most significant impact on fooling the neural network and proceeds to modify it. This iterative process continues until either the adversarial image has reached the maximum permissible number of altered pixels, or the intended deception is successfully achieved.

For an original input data (i.e., image) X, which is classified as label l, i.e., $f(X) = l$. The attacker's goal is to add a tiny perturbation δ_x to produce an adversarial sample X' where $f(X') = f(X + \delta_x) = l$. This can be summarized by following expressing (5).

$$Arg\ min_{\delta_x} \|\delta_x\| \qquad s.t.\ f(X') = f(X + \delta_x) = l' \tag{5}$$

calculating the positive derivative for a given input sample X, the Jacobian matrix is computed as expressed by the following Formula (6):

$$J_f(X) = \frac{\partial f(X)}{\partial X} = \left[\frac{\delta f_j(X)}{\delta x_i}\right]_{i \in 1 \ldots M;\ j \in 1 \ldots N} \tag{6}$$

When compared to FGSM, this technique demands more computational power due to the computation of saliency values. Nonetheless, it significantly limits the number of perturbed features, resulting in the generation of AEs that appear to be more similar to the original sample.

4.1.6. Carlini and Wagner

The Carlini and Wagner (C&W) attack is an optimization-driven technique based on the L-BFGS optimization algorithm. As proposed by Carlini et al. in [53], the C&W attack introduces modifications to the objective function and removes the box constraints typically used in L-BFGS. The authors evaluate three varieties of attacks according to three distance metrics, L_0, L_2, and L_∞. Furthermore, they use an alternative loss function, namely hinge loss, instead of the cross-entropy loss used by L-BFGS. Additionally, they introduce a novel variant denoted as k, transforming the problem from optimizing the perturbation δ

to optimizing k to circumvent the box constraints. The optimization problem is formulated by below Expressions (7) and (8).

$$\min_{\delta} D(X, X+\delta) + c.f(X+\delta) \qquad s.t.\ \ X+\delta \in\ [0,1] \tag{7}$$

$$X+\delta = \frac{1}{2}[tanh(k)+1] \tag{8}$$

where $c > 0$ is a suitably selected constant, δ denotes the adversarial perturbation, $D(.,.)$ denotes the L_0, L_2, and L_∞ distance metrics, and $f(X+\delta)$ define the loss function such that $f(X+\delta) \leq 0$ if and only if the model's prediction matches the attack target. k is the new variant substitute δ as per the above Expression (8).

C&W attack is a white-box adversarial attack. However, this technique shows the ability to transfer from unsecured networks to secured networks. This allows an adversary with limited knowledge of an ML-based security system to carry out a black-box attack. This method outperforms the L-BFGS method in crafting adversarial examples and has demonstrated its efficacy in defeating state-of-the-art defense mechanisms like adversarial training and defensive distillation; however, from a computation cost perspective, it is more expensive than FGSM, JSMA, and others.

4.1.7. DeepFool Attack

DeepFool Attack (DFA) is an untargeted adversarial example generation technique proposed by Moosavi-Dezfooli et al. in [54] to calculate the minimal Euclidean distance (i.e., L_2 distance metric) between the original input (i.e., image) and the adversarial example's decision boundary.

In neural networks, these decision boundaries invariably exhibit nonlinearity. However, to calculate a linear decision boundary that distinguishes samples from different classes, the authors assume that the neural networks operate as entirely linear systems, with class regions being defined by hyperplanes. From this linearization assumption, the DF algorithm calculates the smallest perturbation needed to reach the decision boundary. Then, from the new point, the same operation is iteratively performed multiple times until an adversarial example is found. Formally the minimal perturbation needed to produce an adversarial sample is expressed by (9).

$$\delta(X|f) = \min_{r} \|r\|_2 \quad s.t. \quad f(X+r) \neq f(X) \tag{9}$$

here r is the minimal perturbation, δ is the robustness of the affine classifier f to the original input X for $f(x) = W^T.x + b$ where W is the weight of the affine classifier and b is the bias of the affine classifier.

As white-box attack, the DFA method offers an efficient and precise approach to assess the resilience of ML models. It achieves this by generating adversarial samples with smaller perturbation sizes compared to those generated by FGSM and JSMA methods while having higher deception ratios. However, it is more computationally expensive than both.

4.1.8. Zeroth-Order Optimization

Zeroth-order optimization (ZOO) is a type of adversarial attack that targets ML models where the adversary has only partial knowledge about the targeted model and cannot access its internal parameters or gradients. The attacker's capability is limited to querying the model's output by providing inputs and observing the corresponding predictions. This type of attacks is also known as black-box optimization attacks.

Proposed by Chen et al. [55], the ZOO technique estimates the gradient of the classifier without accessing it ML model by using the symmetric difference quotient approach.

Based on the C&W attack method idea, Chen et al., in contrast, want to design black-box attacks. Therefore, they used the probability distribution instead of using the logit layer representation of a targeted model and they estimated the gradients of the

targeted model by finite differences. Then, the optimization problem is formulated by Expressions (10)–(13).

$$\min_{X'} \|X' - X\|_2 + c.f(X', t) \quad s.t.\ X' \in [0,1]^p \tag{10}$$

where, p is a dimensional column vector and $c > 0$ is a regularization parameter. For X is the original input (i.e., image) affiliated with the specified label l, X' is the adversarial sample affiliated with the specified label t (i.e., $f(X') = t \neq f(X) = l$, where $f(X' + t)$ is the loss function defined by below Expression (11):

$$f(X', t) = max\left\{\max_{l \neq t} log\left[F(X')\right]_l - log\left[F(X')\right]_t, -k\right\} \tag{11}$$

where $F(X') \in \mathbb{R}^K$ is the probability distribution of the back-box output, K is the number of classes and $k \geq 0$ serves as a tuning parameter to enhance attack transferability.

The approximated gradients, defined as $\hat{g}_i$, are computed using the finite differences method called also symmetric difference quotient as per the Expression (12).

$$\hat{g}_i := \frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x - he_i)}{2h} \tag{12}$$

with h being a small constant and e_i represents the i-th component of the standard basis vector. ZOO can be used in Newton's method with Hessian estimate $\hat{h}_i$ as per the following Expression (13).

$$\hat{h}_i := \frac{\partial^2 f(x)}{\partial x_{ii}^2} \approx \frac{f(x + he_i) + 2f(x) - f(x - he_i)}{h^2} \tag{13}$$

Although this method has proven its efficacity in estimating the gradient and Hessian while resulting in a similar performance to the C&W attack, without the requirement of training substitute models or information on the target classifier; however, it necessitates a considerable number of queries to the model, which can add to significant computational costs and time requirements and may cause detection of the attacker in real scenarios.

4.1.9. One-Pixel Attack

The One-Pixel Attack (OPA) is a method used in adversarial ML to deceive image classification models. Building upon the findings of JSMA's success in misleading a network through slight modifications to a few pixels in the input image, Su et al. conducted a study [56] in 2019 that pushed the boundaries even further by showing successful fooling of deep networks by altering as little as one pixel.

The authors used the Differential Evolution (DE) approach [57] to search for the optimal locations and color values that can be modified and creating child-image. Each child-image will be compared to the parent image and the criterion-based fittest is selected for the next iteration. Ultimately, the adversarial example is generated by manipulating the pixel of the last surviving child-image.

The used DE concept does not require knowledge about the system information, the ML model parameters, or its objective function, which is suitable for generating adversarial attacks in a black-box fashion. The problem statement can be mathematically defined as an optimization problem in the following Expression (14).

$$\max_{e(x)*} f_{adv}(x + e(x)) \quad s.t.\ \|e(x)\| \leq L \tag{14}$$

here, $f_t(x)$ is the probability of an image $x = (x_1, \ldots, x_n)$ to be classified as class t and $e(x) = (e_1, \ldots, e_n)$ is the additive perturbation to the each of the n pixels of the image. The constraint here is that the overall perturbation amount is limited to L. However, the

authors used a different approach by modifying the constraint to restrict the quantity of pixels that can be modified. The equation is slightly changed to the Expression (15)

$$\underset{e(x)*}{max} f_{adv}(x + e(x)) \quad s.t. \ \|e(x)\| \leq d \tag{15}$$

where d is a small number of dimensions and $d = 1$ in the case of OPA.

Although this method has proven its effectiveness in generating adversarial examples with a single-pixel change, which keeps the overall appearance of the image almost the same as the original sample and makes attack detection very challenging, evolutionary-based algorithms are computationally expensive.

4.2. Causative Attack Methods

A causative attack, also called a poisoning attack, is an adversarial attack launched while the model is being trained. In this attack, the attacker compromises the training data set by manipulating it or when the ML classifier is trained with limited data and requires additional training data to retrain itself. In this retraining process, the adversary can interfere by introducing incorrect training data. The attacker aims to either degrade the overall performance of the model or target specific training features or classes. This type of attack assumes that the adversary has access to the learning procedure and can influence the training data to deliberately introduce biases or inaccuracies in the model's learning process. Hence, causative attack is a kind of white-box or gray-box attack.

4.2.1. Gradient Ascent

The Gradient Ascent (GA) method is a causative attack proposed by Biggio et al. [58] to significantly decrease the Support Vector Machine (SVM) classification accuracy by inserting crafted data into the training dataset. The method identifies the values associated with local maxima in the model's test error. The authors utilize an incremental learning approach, which seamlessly fine-tunes data point parameters, thus enabling them to achieve an optimal solution by introducing carefully crafted data.

The attacker aims to discover a point (x_c, y_c) that, when added to the training dataset $D_{tr} = \{x_i, y_i\}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$ maximally decreases the SVM's classification accuracy. The attacker proceeds by drawing a validation dataset $D_{val} = \{x_k, y_k\}_{k=1}^{m}$ and maximizing the hinge loss function $L(x_c)$ of the SVM classifier induced on the validation dataset D_{val} and trained on $D_{val} \cup (x_c, y_c)$ as per following Expression (16).

$$\underset{x_c}{max} L(x_c) = \sum_{k=1}^{m} (1 - y_k f_{x_c}(x_c))_+ = \sum_{k=1}^{m} (-g_k)_+ \tag{16}$$

where g_k is the margin constraints impacted by x_c and defined by the Expression (17).

$$g_k = \sum_{j \neq c} Q_{kj} \alpha_j(x_c) + Q_{kc}(x_c)\alpha_c(x_c) + y_k b(x_c) - 1 \tag{17}$$

here, α represents the dual variables of the SVM, which correspond to each training data point. Q_{ss} denotes the margin support vector submatrix of Q.

The authors use the gradient ascent technique to iteratively optimize the non-convex objective function $L(x_c)$. This optimization procedure presupposes the initial selection of an attack point location $x_c^{(0)}$ and in each iteration updates the attack point using the formula $x_c^p = x_c^{p-1} - tu$, where p is the ongoing iteration, u is a norm-1 vector indicating the attack direction, and t denotes the magnitude of the step.

Although this method is a first-order optimization algorithm that only requires the gradient of the objective function calculation, it is sensitive to the starting parameter settings. In case the initial values are too far from the optimal values, the algorithm will most probably converge to a local maximum than a global maximum, or will slowly cover an optimal solution especially, when the objective function is highly non-convex.

4.2.2. Label Flipping Attack

Label-flipping attack (LFA) falls within the category of causative attack methods where the adversary poisons the training dataset by flipping the labels. There are two main methods to add label noise to the training dataset via LFA: random and targeted label flipping. When employing random flipping, the attacker arbitrarily picks a subset of training samples and alters their labels. In contrast, targeted label flipping involves the adversary's pursuit of the most optimal arrangement of label flips that maximizes the classification error rate on the testing data, while adhering to the predetermined number of allowed label flips.

The LFA method was proposed by Biggio et al. in [59] against SVM, following which they improved the method via optimization-based poisoning attacks [58], where the authors resolved a two-level optimization problem to ascertain the best poisoning samples that maximize the hinge loss for SVM. Likewise, Xiao et al. [60] describe the attack strategy as a bi-level Tikhonov regularization optimization problem, followed by the application of a relaxed formulation to identify data instances with near-optimal label flip. Subsequently, these optimization-driven poisoning attacks have been carried out against various types of ML models, including neural networks [61,62] and deep learning [63].

4.2.3. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a category of ML frameworks that have been used to generate adversarial attacks. Initially proposed by Goodfellow et al. [64], GAN is composed of two deep networks: the generator (G) and the discriminator (D), that compete with one another within the context of a zero-sum game. Designed as a Conventional Neural Network (CNN) with two subnetworks, the generator's goal is to generate synthetic data instances that closely resemble those in the training set by initializing its inputs with random noise. On the other hand, the discriminator's goal is to distinguish between synthetic samples produced by G and the original training dataset. A backward propagation is used to enhance the accuracy of G. G receives feedback from D through its loss and tries to minimize this loss while producing adversarial samples. The process concludes when D is unable to differentiate between samples from the training set and those produced by G.

Formally, G is trained to optimize for the probability of D committing wrong classification, and the value function $V(G, D)$ is defined by Goodfellow et al. in [14], by following Expression (18):

$$\min_{G}\max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} log D(x) + \mathbb{E}_{z \sim p_z(z)} log(1 - D(G(z))) \quad (18)$$

where $p_g(x)$ is the generator's distribution over data x, $p_z(z)$ is a prior on input noise variables. $D(x)$ corresponds to the probability that x comes from the original dataset rather than from the generated distribution p_g. $G(z, \theta_g)$ is a differentiable representation embodied by a multilayer perceptron parameterized by θ_g. The objective is to train D to maximize the probability of correctly labeling the training samples, while simultaneously training G to minimize it.

Since its introduction in 2014 by Goodfellow et al. [64], GAN has spawned numerous variants and extensions. These variants address various challenges and limitations associated with the original GAN formulation. For instance, Radford et al. [65] proposed Deep Convolutional GANs (DCGANs) to produce high-quality images compared to fully connected networks, and Mirza et al. [66] introduced a Conditional GAN (C-GAN) framework that can produce images conditioned on class labels. Arjovsky et al. [67] proposed Wasserstein GAN (WGAN) with a new loss function leveraging on the Wasserstein distance to better estimate the difference between the real and synthetic sample distributions. Since 2014, more than 500 papers presenting different variants of GANs have been published in the literature and can be all found in [68].

Although GAN methods excel at generating realistic samples different from once used in training this can help to evaluate the ML systems against adversarial attacks as well as help in data augmentation in scenarios where the available training dataset is limited. However, training GANs are typically characterized by high computational demands and can exhibit considerable instability.

4.3. Inference Attack Methods

An inference attack, alternatively referred to as a model extraction or model stealing attack, is an adversarial attack launched during the deployment or production phase. Inference attack is a technique used by attackers to obtain sensitive information about an ML model or its training data. In a black box scenario, the attacker does not possess access to the inner workings of the model and only has access to its input and output interfaces. The attacker may use various techniques to extract information about the model such as query-based attacks, membership inference attacks, and model inversion attacks.

Orekondy et al. [69] introduced Knockoff Nets as a model-stealing technique that can extract the features of a completely trained model through a two-step process. In the first step, the attacker collects model-generated predictions through a series of input data queries. Subsequently, the collected data–prediction pairs are utilized to construct a substitute model referred to as a "knock-off" model. Likewise, Jagielski et al. [70] proposed a method that involves creating an adversarial model that faithfully reproduces the architecture and weights of the target oracle model. The method is called the Functionally Equivalent Extraction (FEE) attack and prioritizes accuracy and fidelity objectives for model extraction. Chen et al. [71] introduced the Hop Skip Jump Attack, a decision-based attack that estimates the decision boundary of an ML model. The goal of this attack is to cross the estimated boundary deliberately to cause a misclassification.

5. Adversarial Defense Methods in IoT Networks

In addition to the inherent nature of IoT devices, the ML-based security systems in IoT networks are vulnerable to adversarial attacks. As demonstrated in the preceding section, there are various ML-based techniques capable of creating adversarial examples that can easily fool or degrade the performance of the ML models.

To detect and mitigate the various attack strategies discussed in Section 4, there has been a surge in promising defense techniques introduced in recent years, all geared towards enhancing the robustness and resilience of ML models against such attacks. However, the challenge of countering adversarial attacks remains open and continues to elude researchers to find an effective global solution. Most existing defense strategies lack adaptability against various forms of adversarial attacks. While a particular method may successfully counter one type of attack, it often exposes vulnerabilities that can be exploited by attackers who are aware of the fundamental defense mechanism. Additionally, implementing those defense strategies might result in performance burdens and potentially reduce the prediction accuracy of the model in practical usage.

In this section, we discuss the recent advancements in adversarial defense methods, and basing on various defense methods classifications in the literature [17,24,52,72–74], we propose our two-dimensional classification. The first dimension is a defense mechanism that can be a proactive defense mechanism or a reactive defense mechanism [52,72]. The second dimension is a defense strategy of three types: network optimization strategy, data optimization strategy, and network addition strategy [17]. In Figure 7 we summarize the most famous defense methods in use today classified according to our two-dimensional (2D) classification.

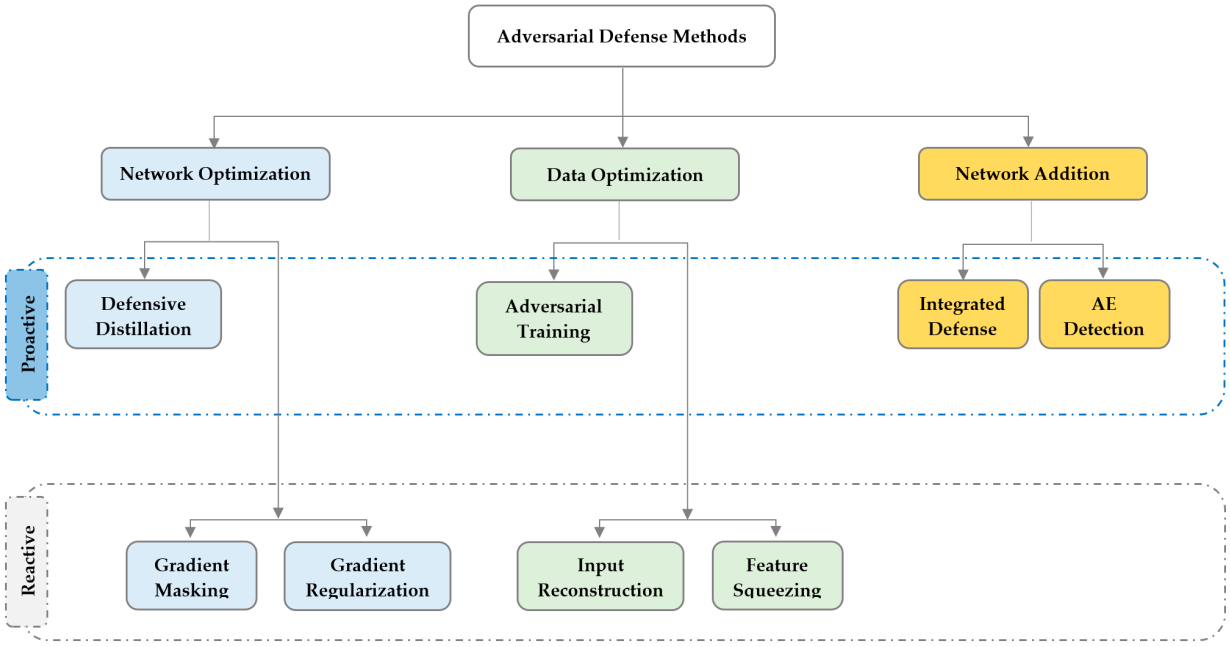

Figure 7. Classification of adversarial defense methods.

5.1. Network Optimization

This strategy involves the modification of the original ML model parameters such as adjusting or adding network layers, changing the loss and/or activation functions, etc. In the literature, numerous proposed defense methods adopt network optimization defense strategy; however, three famous defense methods are widely studied: Defensive Distillation [63], Gradient Masking [75], and Gradient Regularization [76].

5.1.1. Defense Distillation

The concept of distillation was initially put forth by Hinton et al. [77]; it is founded on the concept of transferring knowledge from complex networks to simple networks. Taking cues from this, Papernot et al. [63] proposed to use this concept as a technique to enhance the classifier's resilience against adversarial inputs. For that, the authors proposed a distillation variant called defensive distillation where instead of the traditional usage of distillation that involves training a small model from a large model, the defensive distillation suggests utilizing the knowledge acquired through the distillation process to enhance the classifier's ability to detect adversarial samples.

By setting the temperature T at which a neural network is trained on the Softmax layer. The teaching network inputs are the original examples and labels, and the resulting outputs show a high probability distribution across classes. Consequently, the proposal is to make use of this output in training the distillation network that has the same architecture as the teaching network, to produce a new probability distribution that considers new labels. In the test phase, the authors set the temperature T to 1 to defend against adversarial attacks, as increasing the empirical values of T during the training phase yields enhanced distillation performance.

5.1.2. Gradient Masking

In the context of adversarial defense, gradient masking [75] involves intentionally or unintentionally diminishing the effectiveness of a model's gradients to thwart potential attacks. It encompasses a collection of defensive techniques that operate under the assumption that "if the model is non-differentiable or if the model's gradient is zero at data points, then gradient-based attacks are ineffective" [78], this is because most adversarial attack methods rely on the model's gradient to create the adversarial samples. There-

fore, by obfuscating or hiding gradients it makes it harder for attackers to craft effective adversarial samples.

Folz et al. [79] proposed a gradient-masking method based on a defense mechanism, called the Structure-to-Signal Network (S2SNet). It comprises an encoder and a decoder framework where the encoder retains crucial structural details and refines the decoder using the target model's gradient, rendering it resistant to gradient-based adversarial examples. Lyu et al. [80] proposed a technique based on gradient penalty into the loss function of the network to defend against L-BFGS and FGSM. The study conducted by Nayebi et al. [81] demonstrated how gradient masking can be achieved by saturating the sigmoid network, leading to a reduced gradient impact and rendering gradient-based attacks less effective. The authors compelled the neural networks to operate within a nonlinear saturating system. Nguyen et al. [82] propose a new gradient masking approach to protect against C&W attacks. Their method involves adding noise to the logit layer of the network. Jiang et al. [83] introduce a defense method that modifies the model's gradients by altering the oscillation pattern, effectively obscuring the original training gradients and confusing attackers by using gradients from "fake" neurons to generate invalid adversarial samples.

5.1.3. Gradient Regularization

The concept of Gradient Regularization was introduced for the first time by [84]. It is a method that seeks to enhance the generalization ability of the ML Model by penalizing large changes in the output of the network, using regularization components within the cost function. Ross et al. [76] use this concept to propose a promising defense method against adversarial examples. The authors found that training differentiable models of DNNs with gradient regularization enhances their resilience against adversarial perturbations. Likewise. Lyu et al. [80], and Zhao and Griffin [85] applied a regularization technique to bolster the algorithm's robustness, yielding favorable outcomes in its ability to withstand adversarial attacks. Dabouei et al. [86] introduced a combined approach involving gradient phase and magnitude regularization to improve the robustness of ensemble models. Addepalli et al. [87] introduced a new regularization technique called Bit Plane Feature Consistency (BPFC); this method utilizes information from higher bit planes to form a preliminary understanding, and then refines predictions using only the lower bit planes. Ma et al. [88] proposed a regularization framework called Second-Order Adversarial Regularizer (SOAR) to improve the network's resilience to L_∞ and L_2 limit-bound perturbations produced by PGD [51].

As an adversarial defense method, the Gradient Regularization requires no prior knowledge of an adversarial attack. However, the main drawback is that it doubles the complexity of the training process. Yeats et al. [89] proposed a Complex-Valued Neural Network (CVNN) framework to improve gradient regularization.

5.2. Data Optimization

Unlike the network optimization strategy, which tackles the training models, the data optimization strategy involves modification of data used for training during the training process or modification of input data during the test phase. This strategy mainly includes three defense methods: Adversarial Training [51], Feature Squeezing [90], and Input Reconstruction [91].

5.2.1. Adversarial Training

It is one of the proactive approaches to countering against adversarial attacks. The fundamental goal is to intentionally add adversarial samples into the training set to increase the regularity and robustness of the target model.

When Goodfellow et al. [15] proposed the FGSM attack, they also introduced for the first time an adversarial training technique in the field of imaging by adding adversarial samples to the training set. However, Madry et al. [51] were the inaugural researchers to

theoretically formulate and provide proof through the perspective of robust optimization for DL. Researchers have displayed a notable level of interest in this area of study. This led to multiple contributions proposing several variants of adversarial training method trying to overcome the limitations of this method, such as the data generalization and overfitting, as well as the decreased efficiency to the black-box attacks and the cost can be substantial due to the iterative nature of training the model with adversarial examples.

For large models and data sets, Kurakin et al. [50] made suggestions for adversarial training. Building on the idea that brute force training regularizes the network and reduces overfitting, Miyato et al. [92] proposed the 'Virtual Adversarial Training' approach to smooth the outcome distributions of the neural networks. Zheng et al. [93] proposed the 'stability training' method to improve the resilience of neural networks against small distortions. In their work, Tramèr et al. [94] put forth the Ensemble Adversarial Training (EAT) to augment the diversity of adversarial samples. Song et al. [95] proposed a method known as Multi-strength Adversarial Training (MAT), which integrates adversarial training samples and diverse levels of adversarial strength. Kannan et al. [96] proposed the Mixed-minibatch PGD (M-PGD) adversarial training approach, which combines clean and adversarial examples. Their approach includes a logit pairing strategy with two methods: pairing clean with adversarial samples and pairing clean with clean samples. In the training process, Wang et al. [97] propose to take into consideration the distinctive impact of misclassified clean examples using the so-called Misclassification Aware adveRsarial Training (MART) method. In the objective to solve the generalization issue, Farnia et al. [98] suggested a spectral normalization-based regularization for adversarial training. Wang et al. [99] proposed a bilateral adversarial training method, which involves perturbing the input images and their labels during the training process. In their work, Shafahi et al. [100] proposed the Universal Adversarial Training (UAT) method that produces robust models with only two tile the cost of natural training. Vivek and Babu [101] also introduced a dropout scheduling approach to enhance the effectiveness of adversarial training by using a single-step method. For the overall generalization of adversarially trained models, Song et al. [102] suggested Robust Local Features for Adversarial Training (RLFAT) that involves randomly reshuffling a block of the input during training. Pang et al. [103] propose the integration of a hypersphere method. This method ensures that features are regularized onto a compact manifold.

5.2.2. Feature Squeezing

It is built upon the core fundamental principle that a significant portion of the input feature spaces have higher frequencies than required. Feature squeezing is a reactive data optimization strategy that aims to reduce the space of potential adversarial examples by applying operations that collapse multiple similar inputs into a single representative value. Xu et al. [90] propose the use of two techniques for feature squeezing, namely Bit-Reduction, and Image-Blurring, as a means to mitigate adversarial effects in image classification. The target model provides predictions for both inputs—the original image and the squeezed image. As a result, when a notable contrast emerges in these predictions, the image is recognized as an adversarial sample. In other work, Xu et al. [104] used their methods presented in [90] to mitigate against the C&W attack [53].

As an efficient and cost-effective adversarial defense method, feature squeezing greatly reduces the freedom of the attacker to create adversarial samples. Although the technique's primary application is the field of the image, it might also be transferable to other domains [105], especially in ML-based security systems in IoT networks [106].

5.2.3. Input Reconstruction

It is a reactive mechanism that aims to detect and mitigate the impact of adversarial attacks. The fundamental concept behind input reconstruction is to convert adversarial examples into legitimate data by eliminating the injected perturbations or noise in the original data. By restoring the original input, the ML model can make more reliable predictions by focusing on the original input and disregarding the introduced manipulations. A good

example of this approach is proposed by Gu and Rigazo in [91], where an autoencoder is used for cleaning the adversarial examples. A similar example is the ComDefend autoencoder proposed by Jia et al. [107]. In their work, Song et al. [108] proposed a detecting mechanism based on the PixelCNN autoregressive model to reconstruct adversarial images back to the training distribution.

Due to the inherent slowness of the autoregressive models as well as the difficulty of the autoencoder to remove tiny adversarial perturbations, Ramachandran et al. [109] introduced an accelerated variation of the model to expedite the process. In contrast, Gao et al. [110] introduced an innovative approach that integrates a reconstruction module with a denoising module. The reconstruction module is responsible for the restoration of the original features, while the denoising module ensures the efficient removal of adversarial perturbations, thereby enhancing the overall effectiveness.

5.3. External Model Addition

This strategy involves the use of auxiliary networks or modules to reinforce the resilience of the target model against adversarial attacks. These additional components are designed to detect or mitigate the effects of adversarial perturbations. Integrated defense is one of the common approaches that incorporates an adversarial training module into the training process to train the target neural network model. Another approach is AE detection, where an add-on network or module endeavors to process the data either prior to or subsequent to its transmission to the target model to assist in the detection and exclusion of injected adversarial samples during the prediction phase.

5.3.1. Integrated Defense

It is a common approach that incorporates an adversarial training network or module into the training process to train the target neural network model. One of the most popular frameworks based on GAN [64] is proposed by Lee et al. [111] to develop a robust model that can effectively withstand FGSM attacks [15]. Leveraging on the GAN training, the classifier is trained on both original and created samples. Consequently, the classifier's robustness against FGSM attacks surpassed that of the FGSM adversarially trained model. In a similar approach, Yumlembam et al. [112] proposed a GAN architecture to train and robust an Android Malware Detection using Graph Neural Network (GNN). Benaddi et al. [113] also used GAN to train Distributional Reinforcement Learning (DRL)-based IDS to identify and mitigate minority network attacks while enhancing the effectiveness and resilience of anomaly detection systems within the context of the Industrial Internet of Things (IIoT). In their work, Li et al. [114] proposed Decentralized Swift Vigilance (Desvig) framework, where a C-GAN [66] is integrated to train the network to attain ultra-low latency and highly effective security measures in industrial environments. Benaddi et al. [115] also used C-GAN [66] as an external training network to train and enhance the robustness of Hybrid CNN-LSTM (CNN-Long Short-Term Memory)-based IDS in IoT networks. Inspired by Auxiliary Classifier GAN (AC-GAN) [116] architecture, Liu et al. [117] proposed a framework known as ROB-GAN, combining a generator, discriminator, and PGD-based adversarial attacker as a tripartite game to parallelly enhance both GAN training's convergence speed and the discriminator's robustness under strong PGD adversarial attacks [51].

5.3.2. Adversarial Example Detection

This approach involves integrating an additional network or module that endeavors to manipulate the input data either prior to or after transmitting it to the target model. Its purpose is to aid in the identification and removal of adversarial input samples during the prediction phase. To enhance and generalize the ability of defense methods, Meng et al. [118] argue that it should not depend on the characteristics of adversarial examples originating from a specific generation process. Instead, the primary goal should be to unveil common inherent properties in the generation process of all adversarial examples. Therefore, the au-

thors introduced a defensive framework called MagNet, which solely interprets the results of the final layer of the target classifier as a black-box to detect adversarial samples. For that reason, the MagNet framework is composed of two modules: a Detector and a Reformer. The detector assesses the disparity or distance between a provided test sample and the manifold. If this distance surpasses a predefined limit, the detector rejects the sample. Some adversarial examples might be very close to the manifold of normal examples and are not detected by the Detector. Then, the role of the Reformer is to receive samples classified as normal by the Detector and eliminate minor perturbations that the Detector may have missed. The output from the Reformer is subsequently fed into the target classifier, which will conduct classification within this subset of normal samples.

Another family of defense approaches uses nearest neighbors. Cohen et al. [119] introduced an innovative method for detecting adversarial attacks by leveraging influence functions along with k-nearest Neighbor (k-NN)-based metrics. The influence function is used to evaluate the impact of slight weight adjustments on a particular training data point within the model's loss function, with respect to the loss of the corresponding test data point. On the other hand, the k-NN method is applied to explore the sorting of these supportive training examples in the deep neural network's embedding space. Notably, these examples exhibit a robust correlation with the closest neighbors among normal inputs, whereas the correlation with adversarial inputs is considerably diminished. As a result, this combined approach effectively identifies and detects adversarial examples. In another work, Paudice et al. [120] introduced a data sanitization approach geared towards removing poisoning samples within the training dataset. The technique addresses label-flipping attacks by utilizing k-NN to detect poisoned samples that have a substantial deviation from the decision boundary of SVM and reassign appropriate labels to data points in the training dataset. Shahid et al. [121] developed an extension of the k-NN-based defense mechanism presented by Paudice et al. [120] to evaluate its efficacy against Label-flipping attacks in the context of a wearable Human Activity Recognition System. The authors showed that this enhanced mechanism not only detects malicious training data with altered labels but also accurately predicts their correct labels.

Abusnaina et al. [122] proposed a cutting-edge adversarial example detection method pioneering a graph-based detection approach. The method creates a Latent Neighborhood Graph (LNG) centered on an input example to determine whether the input is adversarial or not. Hence the problem detection of adversarial attacks is reformulated as a graph classification problem. The process starts with the generation of an LNG for every individual input instance, after which a GNN is employed to discern the distinction between benign and adversarial examples, focusing on the relationships among the nodes within the Neighborhood Graph. To guarantee optimal performance in detecting adversarial examples, the authors optimize the parameters of both GNN and LNG node connections. Then, the Graph Attention Network (GAT) is employed to determine whether LNG originates from an adversarial or benign input instance. By employing GAT, the model focuses on the relevant nodes and their connections within the LNG to make an informed decision about the adversarial nature of the input example.

6. Research Works in ML-Based Security Systems of IoT Networks

In this section, we explore the most recent literature on adversarial attacks in the IoT network context. Specifically, we limit our study to the contemporary research on the vulnerability of three ML-based IoT security systems, an Intrusion Detection System (IDS), Malware Detection System (MDS), and Device Identification System (DISs), to the adversarial attacks. The discussion offers a more general outlook on the used system models, methods and techniques, tools, and datasets to evaluate those systems without in-depth technical details, assuming that the previous sections already provided the readers with the required knowledge of this area to understand the different experiment studies we present. Table 2 gives a summary of research works related to ML-based security systems in IoT networks.

The researchers in [123] examined the impacts of adversarial attacks on a variant of a Feedforward Neural Network (FNN) known as Self-normalizing Neural Network (SNN) [124] for IDS in IoT networks. The authors conducted first a performance evaluation of the two IDSes-based FNN and SNN in the absence of adversarial attacks using the Bot-IoT dataset [125]. Then, they created adversarial samples from the Bot-IoT dataset using FGSM, BIM, and PGD methods and they compared the performance of both models against adversarial attacks in the testing phase. The authors demonstrated that while the FNN-based IDS excels in some metrics, such as precision, accuracy, and recall in the absence of adversarial attacks, the SNN-based IDS demonstrates greater robustness in the presence of adversarial examples. Additionally, they analyzed the impact of feature normalization on the ability of DL-based IDS to withstand adversarial attacks in the IoT domain, demonstrating that this defensive approach can have a detrimental impact on the model's resilience to adversarial attacks.

In the context of MDS in IoT Networks, Luo et al. [126] proposed an adversarial attack using a partial-model attack in which the attacker has control of a portion of the available IoT devices. At the stage of data collection and aggregation in IoT systems, the adversary poisons the data inputs by creating adversarial samples using controlled IoT devices. The authors demonstrate that the SVM-based MDS of the IoT network is highly vulnerable to adversarial attacks even when dealing with the manipulation of a small portion of device-generated data. The authors deliberated on the importance of evaluating the effectiveness of defense mechanisms and stated that they would investigate this in their upcoming research.

Papadopoulos et al. [127] proposed to evaluate the robustness of both shallow and DL models against adversarial attacks. Using the BoT-IoT [125] dataset, the authors adopted a methodology that included two main approaches to assess the resilience of SVM-base IDS and Artificial Neural Networks (ANNs)-based IDS against LFA and FGSM adversarial attacks, respectively. In the first approach, targeted and untargeted label poisoning has been used to flip up to 50% of training labels based on the LFA method to cause misclassification by the SVM model. In the second approach, adversarial examples-based FGSM method were experimented on the binary and multi-class ANNs to evade the detection measures. In their experiments, the authors demonstrated a noteworthy probability for an attacker to effectively manipulate or bypass the detection mechanisms. However, the study did not cover the issue related to the imbalanced classes of the BoT-IoT dataset as well as the effect of manipulating high-margin labels from the SVM hyperplane. Also, the study postponed the analysis of the countermeasures' effects on future work.

Table 2. Summary of research works related to adversarial attacks in ML-based security systems of IoT networks.

Ref.	Year	Network	Security System(s)	Target Model (s)		Dataset(s)	Adversarial Attack Methods	Threat Model(s)	Threat Scenario	Adversarial Defense Techniques
				ML	DL					
[123]	2019	IoT	IDS		FNN, SNN	Bot-IoT	FGSM, PGD, BIM	- Evasion	- White-box	- Feature Normalization
[126]	2020	IoT	IDS	SVM		Gaussian Distributions	Gaussian Distributions	- Model Extraction	- Black-box	×
[127]	2021	IoT	IDS	SVM	ANNs	Bot-IoT	LFA, FGSM	- Poisoning - Evasion	- White-Box	×
[128]	2021	IoT	IDS		Kitsune	Kitsune (Mirai)	Saliency Maps, iFGSM	- Evasion - Model Extraction	- Black-box	×
[129]	2021	IoT	IDS		CNN, LSTM, GRU	CSE-CIC-IDS2018	FGSM	- Evasion	- White-box	- Adversarial Training
[130]	2021	IoT	IDS	SVM, DT, RF	MLP	UNSW-NB15, Bot-IoT	JSMA, FGSM, W&C	- Poisoning	- White-box	×
[131]	2021	IoT	IDS	48 DT, RF, BN, SVM		Smart Home Testbed	Rule-Based Approach	- Evasion	- White-box	- Adversarial Training
[132]	2021	IIoT	IDS		DNNs	CIFAR-10, GTSRB	One-Pixel	- Poisoning	- White-box	- Image Recovery
[115]	2022	IoT	IDS		CNN-LSTM	Bot-IoT	C-GAN	- Poisoning	- White-box	- Adversarial Training by C-GAN
[113]	2022	IIoT	IDS		DRL	DS2OS	GAN	- Poisoning	- White-box	- Adversarial Training by GAN
[133]	2022	IoT	IDS	DT	FGMD, LSTM, RNN	MedBIoT, IoTID	Rule-Based Approach	- Poisoning	- Black-box	×
[134]	2022	IoT	IDS		GCN, JK-Net	UNSW-SOSR2019	HAA	- Poisoning - Model Extraction	- Black-box	×
[135]	2022	IoT	IDS		DNNs	CIFAR-10, CIFAR-100	NGA	- Poisoning	- White-box	- Adversarial Training
[136]	2021	IoT	DIS	RF, DT, K-NN	NN	UNSW IoT Trace	IoTGAN	- Evasion - Poisoning	- Black-box	- Device Profiling

Table 2. *Cont.*

Ref.	Year	Network	Security System(s)	Target Model (s)		Dataset(s)	Adversarial Attack Methods	Threat Model(s)	Threat Scenario	Adversarial Defense Techniques
				ML	DL					
[137]	2021	IoT	DIS		CVNN	Generated Device Dataset	FGSM, BIM, PGD, MIM	- Poisoning	- White-box	×
[138]	2022	IoT	DIS	GAP	FCN, CNNs	IoT-Trace	CAM, Grad-CAM++	- Poisoning	- Black-box	×
[139]	2022	IoT	DIS		LSTM-CNN	LwHBench	FGSM, BIM, MIM, PGD, JSMA, C&W, Boundary Attack	- Evasion	- White-box	- Adversarial Training - Model Distillation
[140]	2019	IoT	MDS		CFG-CNN	CFG dataset	GEA	- Evasion	- White-box	×
[141]	2020	IoT	MDS		CNN	Drebin, Contagio, Genome	SC-LFA	- Poisoning	- White-box	- LSD - CSD
[112]	2023	IoT	MDS		GNNs	CMaldroid, Drebin	VGAE-MalGAN	- Evasion	- White-box	- Adversarial Training by VGAE-MalGAN

Qiu et al. [128] studied adversarial attack against a novel state-of-the-art Kitsune IDS within the scenario of black-box access in the IoT network. The authors designed a method leveraging model extraction to create a shadow model with the same behaviors as the target black-box model using a limited quantity of training data. Then, the saliency map technique is used to identify the critical features and to reveal the influence of each attribute of the packet on the detection outcomes. Consequently, the authors granularly modified the critical features using iterative FGSM to generate adversarial samples. Using the Kitsune (Mirai) [142] dataset in their experiments, the authors demonstrated that using their novel technique to perturb less than 0.005% of bytes in the data packets secure an average attack success rate of 94.31% which significantly diminishes the ability of the Kitsune IDS to distinguish between legitimate and malicious packets.

Fu et al. [129] conducted an experiment to assess the efficiency of LSTM, CNN, and Gated Recurrent Unit (GRU) models against adversarial attacks created by FGSM. The evaluation was performed on the CSE-CIC-IDS2018 dataset [143], utilizing three distinct training configurations: training with normal samples, training with adversarial samples, and a hybrid approach involving pretraining with normal samples followed by training with adversarial samples. The results revealed that adversarial training enhanced the robustness of the models, with LSTM showing the most significant enhancement. However, it was observed that adversarial training also led to a reduction in the accuracy of the models when dealing with normal examples. This phenomenon occurred because adversarial training makes the models' decision boundaries more adaptable to adversarial examples, but at the same time, it results in a more fragile decision boundary for normal samples. As a result, the ability of the models to correctly classify normal examples was relatively undermined.

Pacheco et al. [130] assessed the efficiency of the popular adversarial attacks, JSMA, FGSM, and C&W against various ML-based IDSes, such as SVM, Decision Tree (DT), and Random Forest (RF), using multi-class contemporary datasets, BoT-IoT [125] and UNSW-NB15 [27], that represents the contemporary IoT network environment. The study's agenda is to reveal how those several attacks can effectively degrade the detection performance of the three selected target models in comparison to the baseline model Multilayer Perceptron (MLP), and how the performance results vary over the two datasets. The results of the experiment validated the potency of the aforementioned adversarial attacks to decrease the overall effectiveness of SVM, DT, and RF classifiers, respectively for both datasets. However, the decrease in all metrics was less pronounced in the UNSW-NB15 dataset when compared to the Bot-IoT dataset. The limited feature set of Bot-IoT renders it more vulnerable to adversarial attacks. Regarding the attacks, C&W proved to be the most impactful when used with the UNSW-NB15 dataset. In contrast, the FGSM technique displayed robust effectiveness on the Bot-IoT dataset. However, the JSMA had a lesser impact on both datasets. From the classifier's model robustness perspective, the SVM classifier experienced the most significant impact, resulting in an accuracy reduction of roughly 50% in both datasets. Conversely, the RF classifier demonstrated remarkable robustness compared to other classifiers, with only a 21% decrease in accuracy.

Anthi et al. [131] proposed to evaluate the vulnerability of ML-based IDSes in an IoT smart home network. Various pre-trained supervised ML models, namely J48 DT, RF, SVM, and Naïve Bayes (NB) are proposed for DoS attack detection. Using a Smart Home Testbed dataset [144], the authors suggested a Rule-based method to create indiscriminate adversarial samples. For adversarial exploratory attack, the authors proposed to use the Information Gain Filter [145], a feature importance ranking method, to select the crucial features that best distinguish malicious from benign packets. Then, the adversary proceeded to manually manipulate the values of these features, together and one at a time, to force IDSes to wrongly classify the incoming packet. The experiential outcomes revealed that the performance of all IDSes models was impacted by the presence of adversarial packets, resulting in a maximum decrease of 47.2%. On the flip side, the use of adversarial training defense by injecting 10% of generated adversarial samples into the original dataset

improved the models' robustness against adversarial attacks by 25% in comparison to the performance results in the absence of adversarial defense. The approach proposed in this study is restricted to the generation of adversarial examples specifically for DoS attacks, with an exclusive focus on supervised ML-based IDSes.

Husnoo et al. [132] suggested a pioneering image restoration defense mechanism to answer the problem of high susceptibility and fragility of modern DNNs to the state-of-the-art OnePixel adversarial attacks within IIoT IDSes. The authors argue that the existing solutions either result in image quality degradation through the removal of adversarial pixels or outright rejection of the adversarial sample. This can have a substantial impact on the accuracy of DNNs and might result in a hazard for some critical IoT use cases, such as healthcare and self-driving vehicles. The proposed defense mechanism leverages on Accelerated Proximal Gradient approach to detect the malicious pixel within an adversarial image and subsequently restore the original image. In their demonstration experiments, the researchers chose two DNNs-based IDS, LeNet [146] and ResNet [147], and they trained them using the CIFAR-10 [148] and MNIST [149] datasets. The experimental outcomes revealed a high efficacy of the suggested defensive approach against One-Pixel attacks, achieving detection and mitigation accuracy of 98.7% and 98.2%, respectively, on CIFAR-10 and MNIST datasets.

Benaddi et al. [115] suggested an adversarial training approach to enhance the efficiency of hybrid CNNLSTM-based IDS by leveraging C-GAN. The authors introduce the C-GAN in the training pipeline to handle classes with limited samples and address the data imbalance of the BoT-IoT dataset [125]. First, the IDS model is trained on the BoT-IoT dataset, and specific classes with low performance, often those with sparse samples, are identified. Subsequently, C-GAN is trained using these identified classes, and the generator from C-GAN is utilized to retrain the IDS model, thereby improving the performance of the identified classes. The authors plan to further enhance their model by exploring strategies to defend against adversarial attacks to improve the CNNLSTM-based IDS's robustness. In their other work, the authors conducted a similar approach to enhance the robustness and effectiveness of IDS in the IIoT [113]. The study suggests the application of DRL in conjunction with a GAN to boost the IDS's efficiency. By using the Distributed Smart Space Orchestration System (DS2OS) dataset [150], the author's experiments showed that the proposed DRL-GAN model outperforms standard DRL in detecting anomalies in imbalanced dataset within the IIoT. However, the proposed model demands substantial computational resources during the training phase.

Jiang et al. [133] introduced an innovative framework called Feature Grouping and Multi-model Fusion Detector (FGMD) for IDS against adversarial attacks in IoT networks. The framework integrates different models, with each model processing unique subsets of the input data or features to better resist the effects of adversarial attacks. The authors used two existing IoT datasets, MedBIoT [151] and IoTID [152], to validate their model in comparison with three baseline models DT, LSTM, and Recurrent Neural Network (RNN) against adversarial examples which are generated based on a rule-based approach that selects, alters and modifies the features of data samples. The experimental outcomes validated the efficacy of FGMD in countering adversarial attacks, exhibiting a superior detection rate when compared to the baseline models.

Zhou et al. [134] introduced a state-of-the-art adversarial attack generation approach called the Hierarchical Adversarial Attack (HAA). This approach aims to implement a sophisticated, level-aware black-box attack strategy against GNN-based IDS in IoT networks while operating within a defined budget constraint. In their approach, the authors used a saliency map method to create adversarial instances by detecting and altering crucial feature complements with minimal disturbances. Then, a hierarchical node selection strategy based on the Random Walk with Restart (RWR) algorithm is used to prioritize the nodes with higher attack vulnerability. Using the UNSW-SOSR2019 dataset [153], the authors assessed their HAA method on two standard GNN models, specifically the Graph Convolutional Network (GCN) [154] and Jumping Knowledge Networks (JK-Net) [155],

and considering three baseline methodologies, Improved Random Walk with Restart (iRWR) [156], Resistive Switching Memory (RSM) [157] and Greedily Corrected Random Walk (GCRW) [158] when compromising the targeted GNN models. The experiment results proved that the classification precision of both GNN models can be reduced by more than 30% under the adversarial attacks-based HAA method. However, the authors did not examine the effectiveness of their HAA method in the presence of an adversarial defense technique.

Fan et al. [135] argued the limitation of existing evaluation methods that use gradient-based adversarial attacks to assess the Adversarial Training (AdvTrain) defense mechanism [15,51,159]. The authors suggested an innovative adversarial attack method called Non-Gradient Attack (NGA) and introduced a novel assessment criterion named Composite Criterion (CC) involving both accuracy and attack success rate. The NGA method involves employing a search strategy to generate adversarial examples outside the decision boundary. These examples are iteratively adjusted toward the original data points while maintaining their misclassification properties. The researchers carried out their experiments on two commonly utilized datasets, CIFAR-10 and CIFAR-100 [148], to systematically assess the efficiency of the AdvTrain mechanism. In this evaluation, NGA with CC serves as the main method to measure the effectiveness of AdvTrain in comparison with four gradient-based benchmark methods, FGSM, BIM, PGD, and C&W. The study deduced that the robustness of DNNs-based IDSes of IoT networks might have been overestimated previously. By employing NGA and CC, the reliability of DNNs-based IDSes can be more accurately assessed in both normal and AdvTrain defense mechanism scenarios. At the end of this study, the authors recognized their proposed NGA method drawback related to convergence speed and promised to optimize it in their future works.

In the context of Device Identification Systems (DISes), Hou et al. [136] suggested a novel method called IoTGAN, designed to tamper with an IoT device's network traffic to evade ML-based IoT DIS. Inspired by GANs, IoTGAN employs a substitute neural network model in black-box scenarios as its discriminative model. Meanwhile, the generative model is trained to inject adversarial perturbations into the device's traffic to deceive the substitute model. The efficiency of the IoTGAN attack method is evaluated against five target ML-based DIS models: RF, DT, SVM, k-NN, and Neural Networks (NNs) proposed in [160]. The experiments are conducted using the UNSW IoT Trace dataset [161], which is collected within an authentic real-world setting, encompassing data from 28 distinct IoT devices. The experiment outcomes showed that IoTGAN was successful in evading the five target DIS models with a success rate of over 90%. The authors proposed a defense technique called Device Profiling to countermeasure against IoTGAN attacks. This technique leverages unique hardware-based features of IoT devices' wireless signals such as frequency drifting, phase shifting, amplitude attenuation, and angle of arrival. When tested, Device Profiling maintained a high identification rate (around 95%), even under IoTGAN attacks, indicating its resilience against such adversarial strategies.

Likewise, Bao et al. [137] assessed the susceptibility of ML-based DIS against adversarial attacks in IoT networks. The study aims to evaluate the impact of state-of-the-art adversarial attacks on the identification of specific wireless IoT devices based on received signals. For that, the authors launch a single-step attack technique, FGSM, along with three iterative attack techniques, i.e., BIM, PGD, and MIM (Momentum Iterative Method) in targeted and non-targeted scenarios on CNN-based DIS leveraging on a Complex Value Neural Network (CVNN) model [162]. In their experiments, the authors created a generated dataset that contains four main features: Signal Source, Power Amplifier, Channel Attenuation, and Receiver Device. The generated dataset will serve as the foundation for training the CVNN model, which will then be applied for device identification purposes. Leveraging a combined set of evaluation criteria to better assess the model's performance, the study finds that iterative attack methods typically perform better than one-step attacks in fooling ML-based DIS models. However, as perturbation levels increase, their

success rate becomes stable. The outcomes also revealed the ML models' susceptibility to targeted attacks.

Kotak et al. [138] suggested a novel method to produce real-time adversarial examples using heatmaps from Class Activation Mapping (CAM) and Grad-CAM++. They explored the vulnerabilities of ML-based IoT DISes using payload-based IoT identification models such as Fully Connected Neural Network (FCN), CNNs, and Global Average Pooling (GAP). Using a portion of the publicly accessible IoT Trace dataset [161], these models processed the first 784 bytes within the TCP payload and converted them into a 28×28 greyscale image. Experiments involved manipulating unauthorized IoT device data and altering a specific number of bytes to see how these adversarial examples perform when exposed to the target models. Surprisingly, adversarial examples were transferable to varied model architectures. The GAP model displayed unique behavior against these samples, hinting at its defensive potential. Despite vulnerabilities in the target models, advanced architecture like Vision Transformer [163] might resist these adversarial attacks better.

The researchers in [139] delved deep into the performance of ML-based IoT DIS using hardware behavior identification. Therefore, the authors proposed a combined LSTM and CNN (LSTM-1DCNN) model for IoT DIS and evaluated its robustness against adversarial attacks where adversaries alter device environmental and contextual conditions such as temperature changes, CPU load, and device rebooting to hinder its proper identification. To assess the effectiveness of LSTM-1DCNN, the model was trained and tested using the LwHBench dataset [164] and exposed to various adversarial attacks like FGSM, BIM, MIM, PGD, JSMA, Boundary Attack, and C&W. The LSTM-CNN model showcased superior performance, achieving F1-Score of 0.96 in average, identifying all devices with a True Positive Rate (TPR) of 0.80 as threshold for device identification. When exposed to various evasion adversarial attacks, the model remained resilient to temperature-based attacks. However, certain evasion techniques such as FGSM, BIM, and MIM were successful in fooling the identification process. In response, the researchers employed adversarial training and model distillation as defense mechanisms. These mechanisms enhanced the model's robustness. The combination of adversarial training and model distillation provides strong protection against various evasion attacks.

7. Challenges

7.1. Dataset

The scarcity of publicly accessible IoT datasets is evident. Most recent studies have relied on the Bot-IoT [125], Kitsune [142], and CIFAR-10 [148] datasets. Thus, it is essential to create an up-to-date dataset that captures the varied nature of recent IoT applications and considers the newest emerging threats. This would enable a more accurate assessment of IoT ML-based security systems against adversarial attacks in scenarios closely resembling real-world use cases.

Another challenge related to the dataset is unbalanced classes. The procedure to train an IoT ML-based security model involves feeding a specific ML algorithm with a training dataset for learning purposes. Consequently, there is a risk when using datasets such as BoT-IoT [125], UNSW-NB15 [27], and NSL-KDD [26], which are unbalanced with a larger representation of benign data. Such datasets can cause the model to have a bias towards the dominant classes, leading to the "accuracy paradox" problem. For an effective performance evaluation of IoT ML-based security against adversarial attacks it must start by choosing a well-balanced dataset. However, finding a balanced dataset is not always possible. To counteract this, various data balancing methods can be employed:

- Under-sampling: Here, entries from the over-represented class are eliminated to equalize the distribution between the minority classes and majority classes. However, if the original dataset is limited, this approach can result in overfitting.
- Over-sampling: In this technique, we replicate entries from the lesser-represented class until its count matches the dominant class. A limitation is that since the minority

class has few unique data points, the model might end up memorizing these patterns, leading to overfitting.

- Synthetic Data Generation: This method uses Generative Adversarial Networks (GANs) to mimic the real data's distribution and create authentic-seeming samples.

The last challenge from our point of view related to the dataset is features constraints. Most of the studies overlooked the inherent constraints of IoT networks. In contrast to unconstrained domains like computer vision, where the main feature for adversarial perturbation is the image's pixels, the structure of IoT network traffic features involves a combination of different data types and value ranges. These features can be binary, categorical, or continuous. Moreover, the values of these features are closely correlated, with some being constant and others being unalterable.

Given the challenges presented by these data considerations, it is essential to engage in a comprehensive discussion and comparison of datasets when evaluating IoT ML-based security systems, adversarial attacks, or adversarial defense methods. Recent studies in the literature focused on dataset benchmarking [165–168], aiming to elucidate the construction procedures and characteristics of various benchmarking datasets. These studies offer valuable insights for researchers, aiding them in quickly identifying datasets that align with their specific requirements and maintaining the necessary conditions for simulating in the most realistic IoT traffic flows.

7.2. Adversarial Attacks

Diverse methods for generating adversarial attacks have been employed, yet a prominent observation is that a majority of these strategies (60%) rely on a white-box framework. However, this threat model is often unrealistic for potential adversaries. In real-world situations, black-box attacks hold greater practicality, underscoring the need to comprehensively tackle the challenges posed by these attacks and their corresponding defense strategies.

When examining attack methodologies, numerous techniques for crafting adversarial attacks have been put forth. It becomes evident that FGSM holds the highest frequency of usage, with JSMA and C&W attacks following closely. However, FGSM's applicability in the context of IoT ML-based security systems could be an impractical option, given that it operates by perturbing each potential feature to create adversarial examples.

7.3. Adversarial Defenses

Many defense techniques showcased their robustness against some specific adversarial attack but later fell victim to a minor modification of the attack. Additionally, an essential aspect of defensive strategies involves their capacity to endure any form of attack. However, most defense methods prove inadequate when confronted with black-box attacks.

Some defense ideas, like adversarial training and the application of GANs in various variants, are repeated across various research studies. However, a noticeable gap exists in research studies that introduce novel defenses or evaluate the effectiveness of alternative existing adversarial defense mechanisms within the IoT ML-based security domain.

8. Conclusions and Future Works

This paper focuses on the research domain of adversarial machine learning within the context of IoT security. We conducted a review of recent literature that addresses the vulnerability of IoT ML-based security models to adversarial attacks. Our analysis concentrated on three primary IoT security frameworks: Intrusion Detection Systems (IDS), Malware Detection Systems (MDS), and Device Identification Systems (DIS).

Initially, we proposed a taxonomy that can be employed to identify adversarial attacks in the context of IoT security. We subsequently classified adversarial attack techniques using a two-dimensional framework. The first dimension pertains to the phase of attack initiation, encompassing exploratory, causative, and inference attack methods. The second dimension relates to the level of attack knowledge, distinguishing between black-box and white-box attacks. Furthermore, we presented a two-dimensional classification for adversarial

defense methods. In this scheme, the first dimension delves into defense mechanisms, consisting of proactive and reactive approaches. The second dimension encompasses defense strategies, which encompass network optimization, data optimization, and network addition strategies. In the end, we reviewed the recent literature on adversarial attacks within three prominent IoT security systems: IDSs, MDSs, and DISs.

In future works, we aim at using the most recent and realistic IoT dataset in which classes are sufficiently balanced for unbiased learning. We also aim at developing a technique that takes into consideration the nuanced connections between classes to reflect the inherent constraints of IoT networks. Then, we propose an adversarial generation method that maintains these conditions while minimizing the number of perturbed features to ensure the creation of realistic traffic flows. For IoT security systems, we noticed that most of the studies (65%) are dedicated to IDS. Therefore, we will give more attention to MDS and DIS in our future works.

Author Contributions: Conceptualization, H.K., M.R., F.S. and N.K.; methodology, H.K.; validation, H.K., M.R., F.S. and N.K.; formal analysis, H.K.; investigation, H.K.; resources, H.K.; data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, H.K., M.R., F.S. and N.K.; supervision, M.R., F.S. and N.K.; project administration, M.R., F.S. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Global IoT and Non-IoT Connections 2010–2025. Available online: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/ (accessed on 10 December 2023).
2. Khanna, A.; Kaur, S. Internet of Things (IoT), Applications and Challenges: A Comprehensive Review. *Wirel. Pers Commun* **2020**, *114*, 1687–1762. [CrossRef]
3. Riahi Sfar, A.; Natalizio, E.; Challal, Y.; Chtourou, Z. A Roadmap for Security Challenges in the Internet of Things. *Digit. Commun. Netw.* **2018**, *4*, 118–137. [CrossRef]
4. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]
5. Namanya, A.P.; Cullen, A.; Awan, I.U.; Disso, J.P. The World of Malware: An Overview. In Proceedings of the 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 6–8 August 2018; pp. 420–427.
6. Liu, Y.; Wang, J.; Li, J.; Niu, S.; Song, H. Machine Learning for the Detection and Identification of Internet of Things Devices: A Survey. *IEEE Internet Things J.* **2022**, *9*, 298–320. [CrossRef]
7. Benazzouza, S.; Ridouani, M.; Salahdine, F.; Hayar, A. A Novel Prediction Model for Malicious Users Detection and Spectrum Sensing Based on Stacking and Deep Learning. *Sensors* **2022**, *22*, 6477. [CrossRef] [PubMed]
8. Ridouani, M.; Benazzouza, S.; Salahdine, F.; Hayar, A. A Novel Secure Cooperative Cognitive Radio Network Based on Chebyshev Map. *Digit. Signal Process.* **2022**, *126*, 103482. [CrossRef]
9. Benazzouza, S.; Ridouani, M.; Salahdine, F.; Hayar, A. Chaotic Compressive Spectrum Sensing Based on Chebyshev Map for Cognitive Radio Networks. *Symmetry* **2021**, *13*, 429. [CrossRef]
10. Jordan, M.I.; Mitchell, T.M. Machine Learning: Trends, Perspectives, and Prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
11. Talaei Khoei, T.; Kaabouch, N. Machine Learning: Models, Challenges, and Research Directions. *Future Internet* **2023**, *15*, 332. [CrossRef]
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
13. Talaei Khoei, T.; Ould Slimane, H.; Kaabouch, N. Deep Learning: Systematic Review, Models, Challenges, and Research Directions. *Neural Comput. Appl.* **2023**, *35*, 23103–23124. [CrossRef]
14. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. *arXiv* **2013**, arXiv:1312.6199. [CrossRef]
15. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572. [CrossRef]
16. Biggio, B.; Roli, F. Wild Patterns: Ten Years after the Rise of Adversarial Machine Learning. *Pattern Recognit.* **2018**, *84*, 317–331. [CrossRef]
17. Akhtar, N.; Mian, A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *arXiv* **2018**, arXiv:1801.00553. [CrossRef]

18. Akhtar, N.; Mian, A.; Kardan, N.; Shah, M. Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey. *IEEE Access* **2021**, *9*, 155161–155196. [CrossRef]
19. Naitali, A.; Ridouani, M.; Salahdine, F.; Kaabouch, N. Deepfake Attacks: Generation, Detection, Datasets, Challenges, and Research Directions. *Computers* **2023**, *12*, 216. [CrossRef]
20. Xu, H.; Ma, Y.; Liu, H.; Deb, D.; Liu, H.; Tang, J.; Jain, A.K. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *arXiv* **2019**, arXiv:1909.08072. [CrossRef]
21. Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A.; Li, C. Adversarial Attacks on Deep-Learning Models in Natural Language Processing: A Survey. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–41. [CrossRef]
22. Qin, Y.; Carlini, N.; Goodfellow, I.; Cottrell, G.; Raffel, C. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. *arXiv* **2019**, arXiv:1903.10346. [CrossRef]
23. Jmila, H.; Khedher, M.I. Adversarial Machine Learning for Network Intrusion Detection: A Comparative Study. *Comput. Netw.* **2022**, *214*, 109073. [CrossRef]
24. Ibitoye, O.; Abou-Khamis, R.; el Shehaby, M.; Matrawy, A.; Shafiq, M.O. The Threat of Adversarial Attacks on Machine Learning in Network Security—A Survey. *arXiv* **2019**, arXiv:1911.02621. [CrossRef]
25. Carlini, N. A Complete List of All Adversarial Example Papers. Available online: https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html (accessed on 28 October 2023).
26. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
27. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
28. Alatwi, H.A.; Aldweesh, A. Adversarial Black-Box Attacks Against Network Intrusion Detection Systems: A Survey. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10 May 2021; pp. 0034–0040.
29. Joshi, C.; Aliaga, J.R.; Insua, D.R. Insider Threat Modeling: An Adversarial Risk Analysis Approach. *IEEE Trans. Inform. Forensic Secur.* **2021**, *16*, 1131–1142. [CrossRef]
30. Aloraini, F.; Javed, A.; Rana, O.; Burnap, P. Adversarial Machine Learning in IoT from an Insider Point of View. *J. Inf. Secur. Appl.* **2022**, *70*, 103341. [CrossRef]
31. Elrawy, M.F.; Awad, A.I.; Hamed, H.F.A. Intrusion Detection Systems for IoT-Based Smart Environments: A Survey. *J. Cloud Comput.* **2018**, *7*, 21. [CrossRef]
32. Bout, E.; Loscri, V.; Gallais, A. How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 248–279. [CrossRef]
33. Li, J.; Liu, Y.; Chen, T.; Xiao, Z.; Li, Z.; Wang, J. Adversarial Attacks and Defenses on Cyber–Physical Systems: A Survey. *IEEE Internet Things J.* **2020**, *7*, 5103–5115. [CrossRef]
34. He, K.; Kim, D.D.; Asghar, M.R. Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 538–566. [CrossRef]
35. Aryal, K.; Gupta, M.; Abdelsalam, M. A Survey on Adversarial Attacks for Malware Analysis. *arXiv* **2021**, arXiv:2111.08223. [CrossRef]
36. Alotaibi, A.; Rassam, M.A. Adversarial Machine Learning Attacks against Intrusion Detection Systems: A Survey on Strategies and Defense. *Future Internet* **2023**, *15*, 62. [CrossRef]
37. Perwej, Y.; Haq, K.; Parwej, F.; Hassa, M. The Internet of Things (IoT) and Its Application Domains. *IJCA* **2019**, *182*, 36–49. [CrossRef]
38. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
39. Balaji, S.; Nathani, K.; Santhakumar, R. IoT Technology, Applications and Challenges: A Contemporary Survey. *Wirel. Pers. Commun.* **2019**, *108*, 363–388. [CrossRef]
40. Tange, K.; De Donno, M.; Fafoutis, X.; Dragoni, N. A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2489–2520. [CrossRef]
41. HaddadPajouh, H.; Dehghantanha, A.M.; Parizi, R.; Aledhari, M.; Karimipour, H. A Survey on Internet of Things Security: Requirements, Challenges, and Solutions. *Internet Things* **2021**, *14*, 100129. [CrossRef]
42. Iqbal, W.; Abbas, H.; Daneshmand, M.; Rauf, B.; Bangash, Y.A. An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security. *IEEE Internet Things J.* **2020**, *7*, 10250–10276. [CrossRef]
43. Atlam, H.F.; Wills, G.B. IoT Security, Privacy, Safety and Ethics. In *Digital Twin Technologies and Smart Cities*; Farsi, M., Daneshkhah, A., Hosseinian-Far, A., Jahankhani, H., Eds.; Internet of Things; Springer International Publishing: Cham, Switzerland, 2020; pp. 123–149. ISBN 978-3-030-18731-6.
44. Chebudie, A.B.; Minerva, R.; Rotondi, D. Towards a Definition of the Internet of Things (IoT). *IEEE Internet Initiat.* **2014**, *1*, 1–86.
45. Krco, S.; Pokric, B.; Carrez, F. Designing IoT Architecture(s): A European Perspective. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Republic of Korea, 6–8 March 2014; pp. 79–84.

46. Gupta, B.B.; Quamara, M. An Overview of Internet of Things (IoT): Architectural Aspects, Challenges, and Protocols. *Concurr. Comput.* **2020**, *32*, e4946. [CrossRef]
47. Milenkovic, M. *Internet of Things: Concepts and System Design*; Springer: Cham, Switzerland, 2020; ISBN 978-3-030-41345-3.
48. Sarker, I.H.; Khan, A.I.; Abushark, Y.B.; Alsolami, F. Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions. *Mob. Netw. Appl.* **2023**, *28*, 296–312. [CrossRef]
49. Wang, C.; Chen, J.; Yang, Y.; Ma, X.; Liu, J. Poisoning Attacks and Countermeasures in Intelligent Networks: Status Quo and Prospects. *Digit. Commun. Netw.* **2022**, *8*, 225–234. [CrossRef]
50. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Examples in the Physical World. *arXiv* **2016**, arXiv:1607.02533. [CrossRef]
51. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2017**, arXiv:1706.06083. [CrossRef]
52. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. *arXiv* **2015**, arXiv:1511.07528. [CrossRef]
53. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 39–57.
54. Moosavi-Dezfooli, S.-M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. *arXiv* **2015**, arXiv:1511.04599. [CrossRef]
55. Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; Hsieh, C.-J. ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models. *arXiv* **2017**, arXiv:1708.03999. [CrossRef]
56. Su, J.; Vargas, D.V.; Sakurai, K. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Trans. Evol. Computat.* **2019**, *23*, 828–841. [CrossRef]
57. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
58. Biggio, B.; Nelson, B.; Laskov, P. Poisoning Attacks against Support Vector Machines. *arXiv* **2012**, arXiv:1206.6389. [CrossRef]
59. Biggio, B.; Nelson, B. Pavel Laskov Support Vector Machines Under Adversarial Label Noise. In Proceedings of the Asian Conference on Machine Learning, PMLR, Taoyuan, Taiwan, 17 November 2011; Volume 20, pp. 97–112.
60. Xiao, H.; Eckert, C. Adversarial Label Flips Attack on Support Vector Machines. *Front. Artif. Intell. Appl.* **2012**, *242*, 870–875. [CrossRef]
61. Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E.C.; Roli, F. Towards Poisoning of Deep Learning Algorithms with Back-Gradient Optimization. *arXiv* **2017**, arXiv:1708.08689. [CrossRef]
62. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *arXiv* **2015**, arXiv:1505.07818. [CrossRef]
63. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. *arXiv* **2015**, arXiv:1511.04508. [CrossRef]
64. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
65. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434. [CrossRef]
66. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784. [CrossRef]
67. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875. [CrossRef]
68. Hindupur, A. The GAN Zoo. Available online: https://github.com/hindupuravinash/the-gan-zoo (accessed on 28 October 2023).
69. Orekondy, T.; Schiele, B.; Fritz, M. Knockoff Nets: Stealing Functionality of Black-Box Models. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4949–4958.
70. Jagielski, M.; Carlini, N.; Berthelot, D.; Kurakin, A.; Papernot, N. High Accuracy and High Fidelity Extraction of Neural Networks. *arXiv* **2019**, arXiv:1909.01838. [CrossRef]
71. Chen, J.; Jordan, M.I.; Wainwright, M.J. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–20 May 2020; pp. 1277–1294.
72. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef]
73. Barreno, M.; Nelson, B.; Sears, R.; Joseph, A.D.; Tygar, J.D. Can Machine Learning Be Secure? In Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, Taipei, Taiwan, 21 March 2006; pp. 16–25.
74. Rosenberg, I.; Shabtai, A.; Elovici, Y.; Rokach, L. Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain. *ACM Comput. Surv.* **2022**, *54*, 1–36. [CrossRef]
75. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; 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 April 2017; pp. 506–519.
76. Ross, A.; Doshi-Velez, F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients. *AAAI* **2018**, *32*, 1–10. [CrossRef]
77. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531. [CrossRef]

78. Duddu, V. A Survey of Adversarial Machine Learning in Cyber Warfare. *Def. Sc. Jl.* **2018**, *68*, 356. [CrossRef]
79. Folz, J.; Palacio, S.; Hees, J.; Dengel, A. Adversarial Defense Based on Structure-to-Signal Autoencoders. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 3568–3577.
80. Lyu, C.; Huang, K.; Liang, H.-N. A Unified Gradient Regularization Family for Adversarial Examples. In Proceedings of the 2015 IEEE International Conference on Data Mining, Atlantic City, NJ, USA, 14–17 November 2015; pp. 301–309.
81. Nayebi, A.; Ganguli, S. Biologically Inspired Protection of Deep Networks from Adversarial Attacks. *arXiv* **2017**, arXiv:1703.09202. [CrossRef]
82. Nguyen, L.; Wang, S.; Sinha, A. A Learning and Masking Approach to Secure Learning. *arXiv* **2017**, arXiv:1709.04447. [CrossRef]
83. Jiang, C.; Zhang, Y. Adversarial Defense via Neural Oscillation Inspired Gradient Masking. *arXiv* **2022**, arXiv:2211.02223. [CrossRef]
84. Drucker, H.; Le Cun, Y. Improving Generalization Performance Using Double Backpropagation. *IEEE Trans. Neural Netw.* **1992**, *3*, 991–997. [CrossRef] [PubMed]
85. Zhao, Q.; Griffin, L.D. Suppressing the Unusual: Towards Robust CNNs Using Symmetric Activation Functions. *arXiv* **2016**, arXiv:1603.05145. [CrossRef]
86. Dabouei, A.; Soleymani, S.; Taherkhani, F.; Dawson, J.; Nasrabadi, N.M. Exploiting Joint Robustness to Adversarial Perturbations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1119–1128.
87. Addepalli, S.; Vivek, B.S.; Baburaj, A.; Sriramanan, G.; Venkatesh Babu, R. Towards Achieving Adversarial Robustness by Enforcing Feature Consistency Across Bit Planes. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1017–1026.
88. Ma, A.; Faghri, F.; Papernot, N.; Farahmand, A. SOAR: Second-Order Adversarial Regularization. *arXiv* **2021**, arXiv:2004.01832.
89. Yeats, E.C.; Chen, Y.; Li, H. Improving Gradient Regularization Using Complex-Valued Neural Networks. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning PMLR, Online, 18 July 2021; Volume 139, pp. 11953–11963.
90. Xu, W.; Evans, D.; Qi, Y. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In Proceedings of the 2018 Network and Distributed System Security Symposium, San Diego, CA, USA, 18–21 February 2018.
91. Gu, S.; Rigazio, L. Towards Deep Neural Network Architectures Robust to Adversarial Examples. *arXiv* **2014**, arXiv:1412.5068. [CrossRef]
92. Miyato, T.; Dai, A.M.; Goodfellow, I. Adversarial Training Methods for Semi-Supervised Text Classification. *arXiv* **2016**, arXiv:1605.07725. [CrossRef]
93. Zheng, S.; Song, Y.; Leung, T.; Goodfellow, I. Improving the Robustness of Deep Neural Networks via Stability Training. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4480–4488.
94. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble Adversarial Training: Attacks and Defenses. *arXiv* **2017**, arXiv:1705.07204. [CrossRef]
95. Song, C.; Cheng, H.-P.; Yang, H.; Li, S.; Wu, C.; Wu, Q.; Chen, Y.; Li, H. MAT: A Multi-Strength Adversarial Training Method to Mitigate Adversarial Attacks. In Proceedings of the 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, 8–11 July 2018; pp. 476–481.
96. Kannan, H.; Kurakin, A.; Goodfellow, I. Adversarial Logit Pairing. *arXiv* **2018**, arXiv:1803.06373. [CrossRef]
97. Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; Gu, Q. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
98. Farnia, F.; Zhang, J.M.; Tse, D. Generalizable Adversarial Training via Spectral Normalization. *arXiv* **2018**, arXiv:1811.07457. [CrossRef]
99. Wang, J.; Zhang, H. Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6628–6637.
100. Shafahi, A.; Najibi, M.; Xu, Z.; Dickerson, J.; Davis, L.S.; Goldstein, T. Universal Adversarial Training. *arXiv* **2018**, arXiv:1811.11304. [CrossRef]
101. Vivek, B.S.; Venkatesh Babu, R. Single-Step Adversarial Training With Dropout Scheduling. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 947–956.
102. Song, C.; He, K.; Lin, J.; Wang, L.; Hopcroft, J.E. Robust Local Features for Improving the Generalization of Adversarial Training. *arXiv* **2019**, arXiv:1909.10147. [CrossRef]
103. Pang, T.; Yang, X.; Dong, Y.; Xu, K.; Zhu, J.; Su, H. Boosting Adversarial Training with Hypersphere Embedding. *arXiv* **2020**, arXiv:2002.08619. [CrossRef]
104. Xu, W.; Evans, D.; Qi, Y. Feature Squeezing Mitigates and Detects Carlini/Wagner Adversarial Examples. *arXiv* **2017**, arXiv:1705.10686. [CrossRef]
105. Jiang, W.; He, Z.; Zhan, J.; Pan, W. Attack-Aware Detection and Defense to Resist Adversarial Examples. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *40*, 2194–2198. [CrossRef]

106. Asam, M.; Khan, S.H.; Akbar, A.; Bibi, S.; Jamal, T.; Khan, A.; Ghafoor, U.; Bhutta, M.R. IoT Malware Detection Architecture Using a Novel Channel Boosted and Squeezed CNN. *Sci. Rep.* **2022**, *12*, 15498. [CrossRef]
107. Jia, X.; Wei, X.; Cao, X.; Foroosh, H. ComDefend: An Efficient Image Compression Model to Defend Adversarial Examples. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6077–6085.
108. Song, Y.; Kim, T.; Nowozin, S.; Ermon, S.; Kushman, N. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. *arXiv* **2017**, arXiv:1710.10766. [CrossRef]
109. Ramachandran, P.; Paine, T.L.; Khorrami, P.; Babaeizadeh, M.; Chang, S.; Zhang, Y.; Hasegawa-Johnson, M.A.; Campbell, R.H.; Huang, T.S. Fast Generation for Convolutional Autoregressive Models. *arXiv* **2017**, arXiv:1704.06001. [CrossRef]
110. Gao, S.; Yao, S.; Li, R. Transferable Adversarial Defense by Fusing Reconstruction Learning and Denoising Learning. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 10 May 2021; pp. 1–6.
111. Lee, H.; Han, S.; Lee, J. Generative Adversarial Trainer: Defense to Adversarial Perturbations with GAN. *arXiv* **2017**, arXiv:1705.03387. [CrossRef]
112. 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.* **2023**, *10*, 8432–8444. [CrossRef]
113. Benaddi, H.; Jouhari, M.; Ibrahimi, K.; Ben Othman, J.; Amhoud, E.M. Anomaly Detection in Industrial IoT Using Distributional Reinforcement Learning and Generative Adversarial Networks. *Sensors* **2022**, *22*, 8085. [CrossRef] [PubMed]
114. Li, G.; Ota, K.; Dong, M.; Wu, J.; Li, J. DeSVig: Decentralized Swift Vigilance Against Adversarial Attacks in Industrial Artificial Intelligence Systems. *IEEE Trans. Ind. Inf.* **2020**, *16*, 3267–3277. [CrossRef]
115. Benaddi, H.; Jouhari, M.; Ibrahimi, K.; Benslimane, A.; Amhoud, E.M. Adversarial Attacks Against IoT Networks Using Conditional GAN Based Learning. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4 December 2022; pp. 2788–2793.
116. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. In Proceedings of the 34th International Conference on Machine Learning, PMLR, Sydney, Australia, 6 August 2017; Volume 70, pp. 2642–2651.
117. Liu, X.; Hsieh, C.-J. Rob-GAN: Generator, Discriminator, and Adversarial Attacker. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–19 June 2019; pp. 11226–11235.
118. 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 2017; pp. 135–147.
119. Cohen, G.; Sapiro, G.; Giryes, R. Detecting Adversarial Samples Using Influence Functions and Nearest Neighbors. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14441–14450.
120. Paudice, A.; Muñoz-González, L.; Lupu, E.C. Label Sanitization Against Label Flipping Poisoning Attacks. In *ECML PKDD 2018 Workshops*; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11329, pp. 5–15, ISBN 978-3-030-13452-5.
121. Shahid, A.R.; Imteaj, A.; Wu, P.Y.; Igoche, D.A.; Alam, T. Label Flipping Data Poisoning Attack Against Wearable Human Activity Recognition System. In Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, 4 December 2022; pp. 908–914.
122. Abusnaina, A.; Wu, Y.; Arora, S.; Wang, Y.; Wang, F.; Yang, H.; Mohaisen, D. Adversarial Example Detection Using Latent Neighborhood Graph. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 7667–7676.
123. 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), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
124. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. *arXiv* **2017**, arXiv:1706.02515. [CrossRef]
125. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
126. 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, Linz, Austria, 13 July 2020; pp. 13–18.
127. Papadopoulos, P.; Thornewill Von Essen, O.; Pitropakis, N.; Chrysoulas, C.; Mylonas, A.; Buchanan, W.J. Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT. *JCP* **2021**, *1*, 252–273. [CrossRef]
128. 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.* **2021**, *8*, 10327–10335. [CrossRef]
129. 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]
130. 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, Online, 11–13 February 2021; pp. 160–171.

131. 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]
132. Husnoo, M.A.; Anwar, A. Do Not Get Fooled: Defense against the One-Pixel Attack to Protect IoT-Enabled Deep Learning Systems. *Ad Hoc Netw.* **2021**, *122*, 102627. [CrossRef]
133. Jiang, H.; Lin, J.; Kang, H. FGMD: A Robust Detector against Adversarial Attacks in the IoT Network. *Future Gener. Comput. Syst.* **2022**, *132*, 194–210. [CrossRef]
134. Zhou, X.; Liang, W.; Li, W.; Yan, K.; Shimizu, S.; Wang, K.I.-K. Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System. *IEEE Internet Things J.* **2022**, *9*, 9310–9319. [CrossRef]
135. Fan, M.; Liu, Y.; Chen, C.; Yu, S.; Guo, W.; Wang, L.; Liu, X. Toward Evaluating the Reliability of Deep-Neural-Network-Based IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 17002–17013. [CrossRef]
136. Hou, T.; Wang, T.; Lu, Z.; Liu, Y.; Sagduyu, Y. IoTGAN: GAN Powered Camouflage Against Machine Learning Based IoT Device Identification. In Proceedings of the 2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Los Angeles, CA, USA, 13 December 2021; pp. 280–287.
137. Bao, Z.; Lin, Y.; Zhang, S.; Li, Z.; Mao, S. Threat of Adversarial Attacks on DL-Based IoT Device Identification. *IEEE Internet Things J.* **2022**, *9*, 9012–9024. [CrossRef]
138. Kotak, J.; Elovici, Y. Adversarial Attacks Against IoT Identification Systems. *IEEE Internet Things J.* **2023**, *10*, 7868–7883. [CrossRef]
139. Sánchez, P.M.S.; Celdrán, A.H.; Bovet, G.; Pérez, G.M. Adversarial Attacks and Defenses on ML- and Hardware-Based IoT Device Fingerprinting and Identification. *arXiv* **2022**, arXiv:2212.14677. [CrossRef]
140. Abusnaina, A.; Khormali, A.; Alasmary, H.; Park, J.; Anwar, A.; Mohaisen, A. Adversarial Learning Attacks on Graph-Based IoT Malware Detection Systems. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 1296–1305.
141. Taheri, R.; Javidan, R.; Shojafar, M.; Pooranian, Z.; Miri, A.; Conti, M. On Defending against Label Flipping Attacks on Malware Detection Systems. *Neural Comput. Appl.* **2020**, *32*, 14781–14800. [CrossRef]
142. Understanding the Mirai Botnet; USENIX Association, Ed. 2017. Available online: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf (accessed on 13 November 2023).
143. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Madeira, Portugal, 22–24 January 2018; pp. 108–116.
144. Anthi, E.; Williams, L.; Slowinska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
145. Weka 3—Data Mining with Open Source Machine Learning Software in Java. Available online: https://www.cs.waikato.ac.nz/ml/weka/ (accessed on 28 October 2023).
146. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
147. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
148. Krizhevsky, A. CIFAR-10 and CIFAR-100 Datasets. Available online: https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 28 October 2023).
149. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition. *Neural Netw.* **2012**, *32*, 323–332. [CrossRef] [PubMed]
150. DS2OS Traffic Traces. Available online: https://www.kaggle.com/datasets/francoisxa/ds2ostraffictraces (accessed on 28 October 2023).
151. Guerra-Manzanares, A.; Medina-Galindo, J.; Bahsi, H.; Nõmm, S. MedBIoT: Generation of an IoT Botnet Dataset in a Medium-Sized IoT Network. In Proceedings of the 6th International Conference on Information Systems Security and Privacy, Valletta, Malta, 25–27 February 2020; pp. 207–218.
152. Kang, H.; Ahn, D.H.; Lee, G.M.; Yoo, J.D.; Park, K.H.; Kim, H.K. IoT Network Intrusion Dataset. IEEE Dataport. 2019. Available online: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset (accessed on 28 October 2023).
153. Hamza, A.; Gharakheili, H.H.; Benson, T.A.; Sivaraman, V. Detecting Volumetric Attacks on loT Devices via SDN-Based Monitoring of MUD Activity. In Proceedings of the 2019 ACM Symposium on SDN Research, San Jose, CA, USA, 3 April 2019; pp. 36–48.
154. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907. [CrossRef]
155. Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.; Jegelka, S. Representation Learning on Graphs with Jumping Knowledge Networks. *arXiv* **2018**, arXiv:1806.03536. [CrossRef]
156. Zhou, X.; Liang, W.; Wang, K.I.-K.; Huang, R.; Jin, Q. Academic Influence Aware and Multidimensional Network Analysis for Research Collaboration Navigation Based on Scholarly Big Data. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 246–257. [CrossRef]
157. Sun, Z.; Ambrosi, E.; Pedretti, G.; Bricalli, A.; Ielmini, D. In-Memory PageRank Accelerator with a Cross-Point Array of Resistive Memories. *IEEE Trans. Electron. Devices* **2020**, *67*, 1466–1470. [CrossRef]

158. Ma, J.; Ding, S.; Mei, Q. Towards More Practical Adversarial Attacks on Graph Neural Networks. *arXiv* **2020**, arXiv:2006.05057. [CrossRef]
159. Wong, E.; Rice, L.; Kolter, J.Z. Fast Is Better than Free: Revisiting Adversarial Training. *arXiv* **2020**, arXiv:2001.03994. [CrossRef]
160. Bao, J.; Hamdaoui, B.; Wong, W.-K. IoT Device Type Identification Using Hybrid Deep Learning Approach for Increased IoT Security. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 565–570.
161. Sivanathan, A.; Gharakheili, H.H.; Loi, F.; Radford, A.; Wijenayake, C.; Vishwanath, A.; Sivaraman, V. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Trans. Mob. Comput.* **2019**, *18*, 1745–1759. [CrossRef]
162. Trabelsi, C.; Bilaniuk, O.; Zhang, Y.; Serdyuk, D.; Subramanian, S.; Santos, J.F.; Mehri, S.; Rostamzadeh, N.; Bengio, Y.; Pal, C.J. Deep Complex Networks. *arXiv* **2017**, arXiv:1705.09792. [CrossRef]
163. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
164. Sánchez Sánchez, P.M.; Jorquera Valero, J.M.; Huertas Celdrán, A.; Bovet, G.; Gil Pérez, M.; Martínez Pérez, G. LwHBench: A Low-Level Hardware Component Benchmark and Dataset for Single Board Computers. *Internet Things* **2023**, *22*, 100764. [CrossRef]
165. De Keersmaeker, F.; Cao, Y.; Ndonda, G.K.; Sadre, R. A Survey of Public IoT Datasets for Network Security Research. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1808–1840. [CrossRef]
166. Kaur, B.; Dadkhah, S.; Shoeleh, F.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Lamontagne, P.; Ray, S.; Ghorbani, A.A. Internet of Things (IoT) Security Dataset Evolution: Challenges and Future Directions. *Internet Things* **2023**, *22*, 100780. [CrossRef]
167. Alex, C.; Creado, G.; Almobaideen, W.; Alghanam, O.A.; Saadeh, M. A Comprehensive Survey for IoT Security Datasets Taxonomy, Classification and Machine Learning Mechanisms. *Comput. Secur.* **2023**, *132*, 103283. [CrossRef]
168. Ahmad, R.; Alsmadi, I.; Alhamdani, W.; Tawalbeh, L. A Comprehensive Deep Learning Benchmark for IoT IDS. *Comput. Secur.* **2022**, *114*, 102588. [CrossRef]