

How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey

Emilie Bout[✉], *Student Member, IEEE*, Valeria Loscri[✉], *Senior Member, IEEE*, and Antoine Gallais

Abstract—The Internet of Things (IoT) has continued gaining in popularity and importance in everyday life in recent years. However, this development does not only present advantages. Indeed, due to the number of sensitive and private data produced by IoT systems, they have become the new privileged targets for cyberattackers. At the same time, Machine Learning (ML) has gained a phenomenal success in various fields like telecommunications, transport or cybersecurity. Nonetheless, the application of ML can cause significant damage when put in the hands of an attacker. Contrary to many previous works, we do not focus on the potential contributions of ML in the IoT security systems. Indeed, this survey aims to provide a comprehensive overview of ML approaches to enable more effective and less detectable attacks. Thereby, the purpose of this article is to identify and discuss the advantages of the elaboration of ML attacks and the possible solutions already evoked in the literature. Firstly, we provide an identification of the main threats and potential attacks on IoT networks. Then, we investigate on cyberattacks integrating machine learning algorithms during the last few years and we provide future research directions, especially for jamming, side channel, false data injection and adversarial machine learning attacks.

Index Terms—Cyberattacks, machine learning, cyber-security, Internet-of-Things (IoT) networks.

I. INTRODUCTION

THE CONCEPT of machine learning emerged in the middle of the 20th century, nevertheless, it was not until 1990s that the application took off. This revolution has made it possible to simplify and automate numerous tasks in a plethora of fields such as industry, medicine, or marketing. Indeed, the latter offers many benefits like the ability to process immense amounts of complex data and handle tedious and repetitive tasks in record time. The traditional approaches used in cybersecurity during threat detection are mainly based on manual data analytics and statistical rules which require significant time. That is why the use of machine learning algorithms has become an essential element in this domain due to its many advantages. This new ally has made threat detection

more immediate, reactive, and rapid while limiting false positives and being uninterrupted. Machine learning has therefore found several applications in the field of cybersecurity like spam detection and malware analysis [1].

However, as with most of the landmark innovations, machine learning is a double-edged sword. Placed in the hands of a malicious person, this latter can quickly turn into a dangerous weapon as highlighted by reports of Europol and the Royal United Services Institute for Defence and Security Studies [2], [3]. The increasing use of machine learning in the upcoming years will lead to a significant change in the threat landscape in two notable manners. The first will be the expansion of existing attacks. Indeed, by integrating ML algorithms in current cyberattacks, these will become more resistant, more reactive, and less recognizable by the existing detection methods. This new generation of attacks will better target the vulnerabilities of victims and adapt to changes in their environment. The second will represent the creation of new threats which, until then, were not achievable due to their massive demand for data or their excessive manual processing time. Besides, the utilization of machine learning in the protection systems represents a new vector that could be exploited to design advanced attacks.

These intelligent attacks, improbable a few years ago, constitute today real threats. The cyber criminals now dispose of all the necessary tools to be able to establish them in place. Indeed, the emergence of free and open-source frameworks like Tensorflow allow unfamiliar people with the field of ML to implement algorithms quickly [4]. Furthermore, the storage and processing of the data, on which these algorithms are based, represent no longer a problem due to technological advances in big data [5].

The Internet of Things (IoT) refers to a type of network that allows any object to be connected to each other using communication protocols. Connected objects can be of all kinds (e.g., watches, scooters, smart fridges) and can have both sensor and actuator functions. IoT devices are increasingly present in our modern lives and have found applications in a wide range of environments. According to Cisco, by 2030, the number of connected devices is expected to exceed 500 billion [6]. Indeed, the progress of the IoT has made it possible to respond to many problems and to develop several sectors. For example, the integration of sensors such as cameras or intelligent objects (e.g., smart trash, home automation plug) has improved certain points like security or energy consumption problems in a city. Thanks to IoT the transportation sector has been revolutionized with the generation of Intelligent Transportation

Manuscript received September 15, 2020; revised February 12, 2021, June 11, 2021, and September 13, 2021; accepted October 20, 2021. Date of publication November 12, 2021; date of current version February 24, 2022. This work was supported in part by the General Armament Direction, France, and in part by the Defense Innovation Agency, France. (*Corresponding author: Emilie Bout.*)

Emilie Bout and Valeria Loscri are with the FUN—Self-Organizing Future Ubiquitous Network, Inria Lille—Nord Europe, 59650 Villeneuve d'Ascq, France (e-mail: emilie.bout@inria.fr; valeria.loscri@inria.fr).

Antoine Gallais is with the Université Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201 INSA Hauts-de-France, 59313 Valenciennes, France (e-mail: antoine.gallais@uphf.fr).

Digital Object Identifier 10.1109/COMST.2021.3127267

1553-877X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

TABLE I
DIFFERENT AREAS OF IoT APPLICATIONS

Areas	Improvement domains	References
Smart Environment	<ul style="list-style-type: none"> Ecology Security 	[7]
Smart Agriculture	<ul style="list-style-type: none"> Animal farming/tracking Logistics Ecology 	[8]
Smart Transport	<ul style="list-style-type: none"> Fleet Management Traffic Management Driver Assistance Ecology 	[9]
Industry 4.0	<ul style="list-style-type: none"> Custom industry Productivity 	[10]
Health and Sport	<ul style="list-style-type: none"> Live health monitoring 	[11]
Defense 4.0	<ul style="list-style-type: none"> Maintain Equipments Monitoring the Combatants Training Fleet Inventory Decision-making 	[12]
Smart energy 4.0	<ul style="list-style-type: none"> Ecology Resource Management 	[13]

Systems (ITS) allowing the optimization of logistics and fleet management, providing new goods and services, traffic management, driver assistance, etc. Moreover, the analysis and the decision-making has been improved with the help of IoT networks in the military field. Indeed, IoTs can aid defense in six ways: provide situational awareness on the battlefield proactively maintain equipment, monitor a combatant's health, to perform distance training, and provide a fleet and inventory in real time. The main areas of IoT application are detailed in table I. Consequently, these types of equipment carrying private and sensitive data and used in critical infrastructure have become ideal playgrounds for cyber criminals.

Nevertheless, many of the attacks present a lot of deficiencies. Indeed, most require an analysis phase to find the optimal attack strategy, which can be tedious and demanding in terms of human resources and time. Additionally, many attacks could be easily automated to focus on more targets while optimizing their strategies. This is why machine learning algorithms within attack creation could bring many benefits. The consequences of intelligence attacks in the IoT networks range from the loss of information to the damage of the whole network and go far beyond the digital world. Indeed, they have severe repercussions in the real world like the intrusion in a critical environment or distorting important decision-making.

Several attacks against IoT systems have been reported in recent years. One of the most significant is the Mirai bot-net which has infected over half a million IoT devices in the space of a few months and led to various Distributed Denial-of-Service (DDoS) attacks against the Dyn company, whose critical Domain name system-related services are used by numerous Internet actors [14]. This resulted in a massive blackout of Internet services including Twitter, Netflix, and Cable News Network (CNN). Another critical attack on IoT networks in a Ukrainian Power Grid took place in December 2015 [15]. This malware-based attack allowed access to IoT devices and deprived more than 230,000 users of electricity for more than three hours. Other vulnerabilities in IoT networks that could impact a person's life have also been presented in recent years. This is the case with the St. Jude hospital cardiac

devices which have been studied by the research firm MedSec (Miami, Florida) [16], [17]. The team demonstrated this type of device is subject to two major vulnerabilities that can put a patient's life at risk 1) a "crash" attack that can lead to disabling of the device communication; 2) "Battery drain" attack which can waste the energy of the device and makes it out-of-service. The medical sector is not alone in being subjected to the attack targeting IoT devices; it is also the case for the transport domain. In fact, in 2015, 1.4 million vehicles were recalled because it was possible to take remote control of the digital system of a jeep. In another report, a team of hackers managed to take control of a Tesla considering a large distance between them and the car [18]. As a result, attacks on IoT networks have real consequences on our world, potentially putting a person's life at risk.

In addition, recent attacks including machine learning are developing. Social Media Automated Phishing and Reconnaissance (*SnapR*) which allows the attacker to automatically phish users by generating personalized messages based on their hobbies on Twitter is an example [19]. *DeepLocker*, demonstrated the intentional use of machine learning for harmful purposes [20]. Indeed, created by IBM, this attack is a new type of malware integrating machine learning which is designed to identify its target and adapt its strategies automatically using indicators such as voice recognition or geolocation.

Due to the increasing use of IoT devices in critical infrastructures and the threats that develop there, we can expect ML-based attacks to become more and more robust and present in the coming years. Thereby, intelligent attacks in IoT networks constitute an imminent danger, so there is an urgent need to investigate them in order to design new security systems and communication protocols.

A. Scope of This Survey and Contributions

Many comprehensive surveys explaining the security and privacy issues in the IoT networks exist in the literature. A brief discussion on the relevant limitations of the IoT devices which are perceived as potential vulnerabilities is offered in [21], [22]. In parallel, the use of machine learning methods has transformed security systems in IoT in recent years. Several researchers have conducted surveys on security methods integrating machine learning on IoT networks to give a practical guide to existing solutions. A state-of-the-art of various machine learning-powered technique security systems is given in [23]. Moreover, reports demonstrating the imminent dangers of ML-powered attacks are beginning to emerge [24], [25]. They also describe the fact that there is a lack of awareness of possible malicious uses of machine learning on the part of the various players in IoT networks.

This is why, this survey presents a comprehensive review of the machine learning-based attacks on IoT networks, in order to provide a roadmap for future work. The main ambition of this paper is to identify the new intelligent cyberattacks integrating Machine Learning mechanisms in order to exploit this knowledge by designing a new generation of robust communication protocols. This paper also aims at contributing to

manufacturers and researchers' awareness about the ease of developing attacks using machine learning. Indeed, this growing threat should be considered for the conception of IoT devices and security systems.

The fundamental contributions of this work comprise the following.

- We provide a detailed state of the art of the existing surveys on IoT security, Machine Learning in IoT and ML-based security solutions on IoT networks and we highlight the main contributions of our work in respect of the existing surveys.
- We provide a description of how smart attacks, namely attacks based on the integration ML schemes are generated and the main features of such a type of attacks. Moreover, we add a new classification of attacks based on machine learning algorithms integration.
- This survey focused on the Machine learning based attacks on IoT networks that we found in the literature since 2014. State-of-the-art on the different machine learning methods used to create each attack is provided. The possible challenges and the research perspectives to improve each intelligent attack is presented.
- Thanks to our various researches, a definition of the common characteristics of an attack more likely to succeed is elaborated.

B. Organization

Fig. 1 provides a visual representation of the article organization. Section II discusses related works to highlight the major differences of this survey from the previous surveys on IoT security and ML in IoT networks. An overview of machine learning algorithms to design attacks is provided in Section III. The advantages of machine learning methods to develop intelligent attacks are discussed in Section IV. We also introduce in this part, a taxonomy of the different intelligent attacks. The next four Sections V, VI, VII, and VIII show a study of the articles published on different machine learning-based attacks. In these sections we also discuss the knowledge gained thanks to the analysis of the different ML schemes. Finally, a discussion on the direction on the future research is presented in Section IX following by a conclusion in Section X.

II. RELATED SURVEYS

Many surveys have been published that cover different aspects of the IoT security. In this section we summarize the existing surveys on IoT Networks threats, the machine learning algorithms applied to them and more specifically ML-based security solutions.

A. IoT Network Threat and Security Surveys

Several comprehensive surveys have been conducted on the IoT characteristics and their potential vulnerabilities to provide a practical guide to IoT security and a roadmap for future works. A summary of all these surveys is given in Section III. The phenomenal growth in employment of IoT devices in some fields in recent years is shown in [26]–[28]. Numerous surveys explain the considerable advances and challenges of

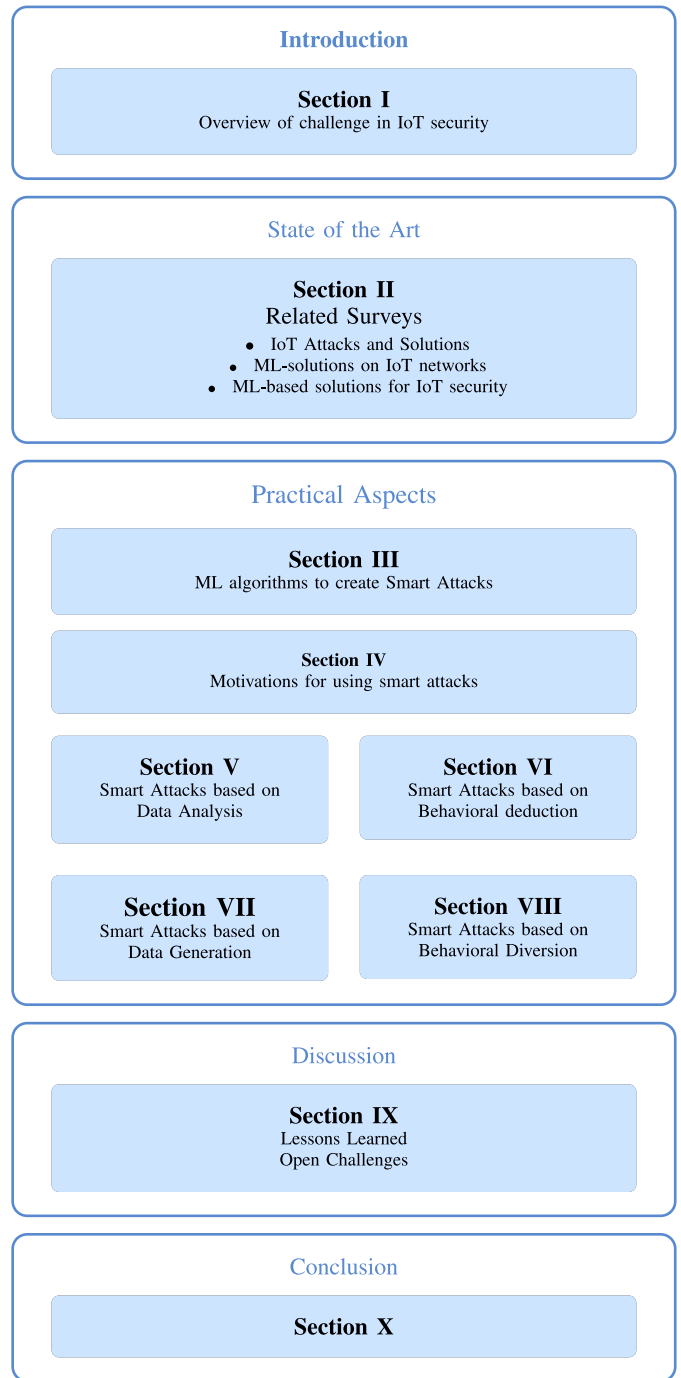






Fig. 1. Structure of this article and relations.

using IoT devices in certain domains that we have succinctly summarized in Table I.

Moreover, several criteria that an IoT network must satisfy such as inter/outer-connectivity, heterogeneity, dynamic changes, scale and safety are presented in [29]. These needs give rise to a complex architecture of an IoT system that can be divided into several distinct layers. The IoT system can be divided in three distinct layers: physical, network and application as mentioned in surveys [30], [31], [32], [33]. The first includes the IoT physical sensors and actuators that allow the implementation of various functionalities like data collection. The second level establishes the connection between the

TABLE II
THE DIFFERENT CLASSES OF ATTACKS ON IoT NETWORKS

Physical Attacks	Network Attacks	Software Attacks	Encryption Attacks
			
Vulnerabilities exploited			
Centered on the hardware components of the IoT system and the communication medium	Based of the vulnerabilities of communications protocols	Modify or exploit known vulnerabilities of the software code of the IoT device	Attempt to break the encryption scheme
Examples of attacks			
<ul style="list-style-type: none"> • Jamming Attacks • Tampering Attacks • Physical Damage 	<ul style="list-style-type: none"> • Replay Attacks • Routing Attacks • Traffic Analysis • Damage • Spoofing Attacks 	<ul style="list-style-type: none"> • Virus and Worms • Spyware Attacks • Trojan Horse 	<ul style="list-style-type: none"> • Side Channel Attacks • Cryptanalysis Attacks

physical object and the end-user. It is mainly responsible for the communication and connectivity of all the devices in the IoT system with the help of various communication protocols. We can also add the integration of Big Data or Cloud analytical methods permitting the creation of cognitive IoT systems. The last layer represents the interface with which the end-user interacts with IoT devices, it can be a button or an application on a smartphone for example.

Due to the specific characteristics of each layer of the IoT system, IoT networks are subject to many critical vulnerabilities as mentioned in survey [34]. These vulnerabilities are primarily related to poor physical security, resource constraints, insufficient authentication and encryption, insecure access controls, and inadequate update management. Indeed in many cases, IoT devices are deployed in poorly secured areas. This accessibility may subject the device to tampering such as circuit modification and even replacement. Moreover, the attacker can easily take control of them and extract cryptography primitives to obtain unlimited access to the information stored on the memory chip.

Another vulnerability may be the fact that IoT devices are objects constrained in energy and storage. Most of this equipment run on non renewable and easily exhaustible batteries. An attacker can play on this flaw by forcing the node to waste all its available energy in order to make it unusable.

Moreover, most of the time, IoT devices do not force the user to use complex and secure authentication. However, if an IoT system does not have reliable authentication, then it becomes easy for an attacker to impersonate an IoT device or to intercept its data, especially if the data is not a minimum encrypted.

Several authors have proposed a classification of possible attacks according to the vulnerability of each layer composing the architecture of an IoT system in [30], [35] and [36].

This classification includes four distinct categories: physical, network, software, and encryption attacks. The first groups all types of attacks that target the hardware components of

the IoT system and the communication medium. Most of the attacks of this category are primarily intended to cause a Denial-of-Service (DOS). Wood and Stankovic determine a Denial-of-Service as “any event that diminishes or eliminates a network’s capacity to perform its expected function” [37]. This is generally done by targeting the resources essential to the proper functioning of the equipment, such as its connectivity, memory, or even its battery. We can find jamming attacks [38] which aims to voluntarily interfere with the signal transmission to disrupt or prevent it.

The second is focused on the network layer attacks. It classifies all the attacks which are based on the vulnerabilities related to the communication protocol. Traffic analysis attacks [39], Black-hole attacks [40], Sinkhole attack [41] belong to this category. The software category gathers all the attacks that modify or exploit known vulnerabilities in IoT device software code. This kind of attack is often conducted at the application layer. Trojan horse programs, worms, viruses, spyware, and malicious scripts are attacks that can be employed in this category. To finish, the last class gathers attacks that attempt to break the encryption scheme in order to gain access to data. The side-channel attack is an encryption attack, based on the analysis of many outside elements of an IoT device like the power consumption during a specific task or the timing, an attacker tries to recover the encryption scheme [42]. Cryptanalysis attacks are one of the oldest and best-known encryption attack. An attacker possesses a component of the encryption mechanism such as ciphertext or plaintext and attempts to find the encryption key. A brief overview of this classification with different example of attack is given in Table II. Some attacks will be more detailed in the following sessions.

However, other classifications of attacks on IoT networks exist. Butun *et al.* categorized the types of potential attacks in IoT networks according to the attacker’s activity (active/passive) and the targeted OSI (open Systems Interconnections) layer [43]. Several surveys providing studies

TABLE III
SUMMARY OF EXISTING SURVEYS RELATED TO IOT SECURITY SORTED IN CHRONOLOGICAL ORDER

Date	Ref	Major Contribution(s)	Vulnerabilities	Attacks	Solutions	Challenges
2015	[35]	Categorization of IoT attacks under four classes and brief explanation of existing solutions for each category	full	full	partial	none
2017	[30]	Classification of different attacks on IoT networks according to 4 classes: Physical, Network, Software and Encryption Attack	full	full	none	none
2017	[31]	Classification of attack according to the layers of IoT architecture	full	full	none	none
2018	[26]	Overview of the major existing and upcoming solutions for IoT security: blockchain based solutions, fog computing based solutions, machine learning based solutions and edge computing based solutions	full	full	full	full
2019	[32]	Classification of attacks and solutions according to the layers of IoT architecture	full	full	full	partial
2019	[36]	Classification of attack according to damage-level and attack-target for IoT	full	full	partial	none
2020	[34]	An unique taxonomy of IoT vulnerabilities	full	full	full	full
2020	[43]	Categorisation of IoT attacks according to the attacker activity and the targeted OSI layers	full	full	full	full
2021	Our work	Overview of existing ML-based attacks on the IoT network	full	full	full	full

of a specific kind of attack on IoT networks have also been conducted, focusing on, e.g., distributed denial-of-service attacks or routing layer attacks [44], [45].

Possible solutions to security problems in IoT networks have been proposed in [21], [22], [26], [46]. However, until now no published paper has proposed standard solutions to all the security issues in IoT networks. Indeed, each manufacturer designs and uses different IoT devices and middleware for reasons of cost, time but also need. Additionally, many communications technologies like 6LoWPAN, Bluetooth, IEEE 802.15.4, WiFi, ultra-wide bandwidth, radio frequency identification (RFID) and near-field communication (NFC) are used to connect IoT systems [47]. This is why creating a single solution against attacks in IoT networks still remains an open challenge and many surveys focus on the implementation of a solution for a specific problem. Ngu *et al.* focused on an analysis of existing security solutions in relation to vulnerabilities related to middleware and network layer in [48]. The pros and cons of various trust management techniques for IoT are surveyed in [49]. Researchers discuss Software-Defined Network (SDN) and Network Function Virtualization (NFV) based security solutions in IoT environments, comparing them to traditional security countermeasures in [50].

Finally, a new type of security solution has emerged in recent years: the Light Weight solutions. These are solutions that respect the characteristics of IoT networks such as memory or battery constraints. Lightweight encryption algorithms to improve security in the IoT is identified in [51]. Research is pushed further by additionally studying works on the two most important symmetric cryptographic ciphers: stream ciphers and block ciphers in [52]. A summary of different kinds of lightweight cryptographic algorithms that are easy to use for hardware and software implementations applied in smart home environment are given in [53].

B. Surveys on Machine Learning Algorithms Applied to IoT Networks

With machine learning applications in IoT on the rise in recent years, in-depth analyses have been performed on

the scientific literature to study current tendencies as well as elaborate helpful guidelines. Indeed, machine learning algorithms have improved or solved many open and challenging problems such as resource control, networking, mobility management and location in IoT networks.

In [54], the authors principally focus on the existing ML based solutions for the management resources problems like power consumption. Alsheikh *et al.* compare the advantages, disadvantages and complexity of each ML algorithm which has been implemented in several sub-problems like localization, improvement of quality of service (QoS) or security, in [55].

In addition, due to the growth in the use of IoT devices and therefore various problems that this implies like spectrum management, it has become interesting to integrate cognitive radios (CR) within them. The need for learning and provide a review of the application of ML algorithms for CRs is explained in several surveys [56]–[58]. Deep learning (DL) algorithms can be used at the physical layer of wireless communication systems to substitute elements of the conventional communication system and create a new DL-based architecture [59]. Several applications of using DL have also emerged as alternative systems such as modulation recognition and channel decoding are presented in [60]. Moreover, a comprehensive analysis also centred on DL algorithms in IoT networks for upper layers has been performed in [60]–[63].

A comprehensive study, covering all layers, on the different machine learning algorithms and not just those based on deep learning methods is presented in [64]. Indeed, a profound investigation of the applications of machine learning is provided according to the problem types, training data availability, the time cost and the motivation to adopt them. In addition, each area is clearly divided into multiple sub-domains like power control, spectrum control and the resource management. Finally, Machine Learning algorithms can also make it possible to respond to many challenges related to the preparation and processing data essential for the proper functioning of IoT networks. In another survey, the authors explain the different smart methods that have been put in place to meet these needs and gives an overview of the application of machine learning algorithms to the IoT use cases [65].

TABLE IV
SUMMARY OF EXISTING SURVEYS RELATED TO MACHINE LEARNING APPLIED TO IoT NETWORKS

Date	Ref	Major Contribution(s)	ML Paradigms		Resources Management ³	Objectives		
			DL 1	Other ML 2		Routing 4	Data 5	Security
2012	[57]	Classification of the use of ML algorithms for CR in two main categories: decision-making and feature classification	✗	✓	✓	✗	✗	✗
2015	[56]	A comprehensive survey considering all the learning techniques that were used in cognitive networks	✗	✓	✓	✗	✗	✗
2016	[54]	Classification of future ML applications in wireless networks according to ML paradigms	✗	✓	✓	✗	✗	✗
2017	[59]	Summary of Emerging DL-Based Physical Layer Studies	✓	✗	✓	✗	✗	✗
2017	[60]	Investigations of the motivation and limitations of DL approaches for physical layer	✓	✗	✓	✗	✗	✗
2018	[55]	Comparison of strengths and weaknesses of ML algorithms for wireless networks applications	✗	✓	✓	✓	✓	✓
2018	[63]	Taxonomy of DL application in wireless networks.	✓	✗	✓	✓	✓	✓
2018	[66]	Overview of the application of machine learning algorithms for an IoT use case: the smart-city	✓	✗	✗	✗	✓	✗
2019	[61]	Review of DL approaches to address emerging issues for communications and networking	✓	✗	✓	✓	✓	✓
2019	[65]	State-of the art application of various application of ML approaches for IoT network for each layer	✓	✓	✓	✓	✗	✗
2019	[64]	Up-to-date survey for DL applications and their pros and cons in wireless networks	✓	✓	✓	✓	✓	✓
2021	Our work	Overview of existing ML-based attacks on the IoT network	✓	✓	✗	✗	✗	✓

¹ DL: Deep learning ²: Groups the supervised and unsupervised algorithms ³: Refers to medium access control, power allocation, signal classification and modulation classification problems

⁴: Groups together all the problems that refer to routing as the choice of the best path ⁵: Combines data aggregation and sorting issues

TABLE V
SUMMARY OF EXISTING SURVEYS RELATED TO IoT SECURITY TECHNIQUES BASED ON MACHINE LEARNING

Date	Ref	Major Contribution(s)	ML Paradigms		Category of attack				Detection approach		
			DL 1	Other ML 2	DOS 3	U2R 4	R2L 5	Probe/Scan	Misuse	Anomaly	Hybrid
2015	[75]	Paper focuses on ML and DM techniques for cyber security, with an emphasis on the ML/DM methods and their descriptions	✓	✓	✓	✓	✓	✓	✓	✓	✓
2018	[68]	Overview of ML-based solutions against IoT offloading and malware	✓	✓	✗	✗	✗	✓	✓	✗	✗
2018	[73]	Summary of existing IDS approaches based on ML and their performances.	✓	✓	✓	✓	✓	✓	✓	✓	✓
2019	[79]	Comparison and evaluation of the various machine learning contributions for IoT NIDSs	✓	✓	✓	✓	✓	✓	✓	✓	✓
2019	[72]	Taxonomy of ML technologies for malware detection.	✓	✗	✓	✗	✗	✓	✗	✓	✗
2020	[69]	Disadvantage and advantages for each ML algorithms for detection systems.	✓	✓	✓	✗	✗	✓	✗	✓	✗
2020	[70]	In-Depth Review of the ML and DL Methods for IoT Security and their applications for each layer.	✓	✓	✓	✓	✓	✓	✓	✓	✓
2021	Our work	Overview of existing ML-based attacks on the IoT network	✓	✓	✓	✓	✓	✓	✗	✗	✗

¹ DL: Deep learning ² Groups the supervised and unsupervised algorithms ³ DOS: Denial of Service attack ⁴ U2R: User to Root

⁵ R2L: Remote to Local

Table IV provides a summary of the above surveys. These studies are categorized according to their objectives. Indeed, the category resource management groups the medium access control, the power allocation, the signal classification and the modulation classification problems. ‘Data’ refers to the aggregation of data and the sorting of data. These papers are also classified according to the ML paradigms mentioned. ‘DL’ is the equivalent of Deep learning and ‘Other ML’ refers to all other machine learning algorithms such as supervised /unsupervised algorithms.

C. Surveys on IoT Security Techniques Based on Machine Learning on the IoT Networks

As seen in the previous sub-section, the use of ML methods has transformed security systems in IoT in recent years. Several researchers have conducted surveys on security methods integrating machine learning on IoT networks to give a practical guide to existing solutions. These researches are recapitulated in table V. Four types of attack categories: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe/Scan were defined when creating different data sets

for the detection [66]. Therefore, this classification has been used in many studies, which is why we have classified the following studies in the same way. A DoS attack aims to limit the resources of an IoT device or the network. A U2R attack grants root access to the attacker while an R2L attack gives network access to the attacker. The Probe/Scan attack collects information about network or users. Moreover, the use of machine learning for security solutions can be employed with different approaches. We have also categorized the different works according to them in Table V. The first is misuse-based technique designed to detect known attacks by using signatures of those attacks. The second is a technique based on anomalies: the normal behavior of the system and the network are known and when an unfamiliar element is traced, an attack is detected. The last is a hybrid solution that combines misuse and anomaly detection.

Defence solutions created with the help of machine learning are investigated by the authors in [67]. One of the points addressed is the study of the solutions against IoT offloading (e.g., jamming, flooding, eavesdropping attacks) with machine learning. They are also studying the different learning-based IoT malware detection methods in IoT networks. Surveys take the investigations even further by considering only the Machine Learning security solutions adapted to the characteristics of an IoT networks such as the restrictions on computing resources [68]. Different machine learning techniques and their applications to address various IoT attacks are explained in [23], [69]. Additionally, the recent advances in Deep Learning Methods for IoT Security have been reviewed in-depth [69].

Machine Learning algorithms can also be very useful in detecting malware at the application layer of IoT systems. The idea is not new, the survey written by Shabtai *et al.* in 2009 is the first of this topic [70]. However, many advancements in this subject have taken place. Indeed, it is nowadays possible to hijack the initial function of a security system thanks to attacks generated using machine learning (evasion attack). Therefore, [71] takes this vulnerability into account and only deals with robust malware detection systems in the face of these attacks.

Another efficient way to secure IoT networks can be the use of network intrusion detection (IDS). These surveillance systems were initially based on rules/signatures or attack behavior specifications. However, generating multiple false positives, many replacement systems incorporating Machine Learning algorithms have emerged. As a result, there are several IDS based on learning and therefore surveys techniques on this topic. References [72]–[74], give a comprehensive overview of the different machine learning-based IDS, their advantages and their limitations. Moreover, a comprehensive review of the datasets that exist to feed these IDSs is provided in [74]. Other investigations are more specific and focus exclusively on IDSs designed for IoT networks [75]–[77]. They provide a comprehensive guide and overview of the pros and cons of each ML-based IDS introduced in the literature. They focus on the limits of these algorithms applied to the IoT network and compare each solution according to extremely specific criteria like its location in the network or its performance.

D. Motivation of This Survey

The major contribution of Section II is an overview of existing surveys on IoT security and the application on ML in IoT networks. However, there are some important points like the urgent need to develop solutions against new potential, more powerful threats based on the increasing integration of ML to generate smart attacks, that need to be pointed out. Indeed, due to the pervasive use of IoT in critical infrastructures (e.g., hospitals, military infrastructures), many sensitive and private data circulate on IoT networks. In addition, it is extremely difficult to protect them and develop a unique and hundred percent reliable security framework due to the high complexity of such a type of structures and their heterogeneity.

In particular:

- An IoT system is complex and is based on three main layers which can also be divided into sublayers. Each contains many attack surfaces and is consequently vulnerable. It is difficult for a person to determine all the attack vectors that exist and those that may be possible when developing such a framework.
- Most of the time, each IoT network is unique in its field of application, its functionalities and the sensors used. Indeed, each manufacturer can develop their own IoT device and integrate their personal firmware into it. In addition, there are many communication protocols available to ensure the connectivity of an IoT network. This heterogeneity of networks does not facilitate the establishment of a common solution.

In addition, many Machine Learning algorithms are implemented and used nowadays in IoT networks in order to satisfy certain challenges like security. However, these advances have opened the door to new types of attacks. Indeed, machine learning algorithms themselves have vulnerabilities that can be exploited by a malicious user [78]. These kinds of attacks, named adversary machine attacks, aim to target the ML algorithms of a system in order to alter these initial functions. We can assume that with the rise in the use of ML algorithms in IoT, this type of attack in the coming years will continue to expand. Therefore, it is important to highlight that it is essential to take into account this threat when setting up security solutions integrating ML systems.

Finally, if machine learning algorithms can be used to improve security systems, it is quite possible to think the latter can be employed for malicious purposes. Indeed many of attacks still have a lot of deficiencies. Most require an analysis phase to find the optimal attack strategy, which can be tedious and demanding in terms of human resources and time. In addition, this step is most often specific for each victim. This is the case with traffic analysis attacks, for example where for each victim the attacker must study the data of the networks referring to the latter. Additionally, we believe that many attacks such as jamming attacks, could be easily automated to focus on more targets while optimizing their strategies. This is why machine learning algorithms within attack creation could bring many benefits. In contrast with the other surveys mentioned above, we are not going to take the point of view of the defense side but that of an attacker by showing here the different advantages of creating

TABLE VI
MACHINE LEARNING ALGORITHMS EXPLOITED FOR GENERATING SMART ATTACKS

Algorithm	Paradigms	Advantages	Disadvantages	Application in Attack
K-NN	Supervised	<ul style="list-style-type: none"> Intuitive and Simple. Versatile: different distance criteria. Classification and Regression use. 	<ul style="list-style-type: none"> Curse of Dimensionality. Slow algorithm. Need homogeneous features. 	Deduce IoT information like activities or sensitive data
DT		<ul style="list-style-type: none"> Requires little data preprocessing. Work with numerical and categorical features. 	<ul style="list-style-type: none"> Instability. Complexity. 	
RF		<ul style="list-style-type: none"> Solve the high variance estimator problem of Decision tree. Classification and Regression use. 	<ul style="list-style-type: none"> High computational costs. Predictions are slower. 	
ET		<ul style="list-style-type: none"> Higher performance in presence of noisy features than RT. Lower cost than RT. 	<ul style="list-style-type: none"> Predictions are slower. 	
SVM		<ul style="list-style-type: none"> Effective in high dimensional spaces. Versatile: different Kernel functions. High Accuracy. Works well on smaller cleaner datasets 	<ul style="list-style-type: none"> Training time with SVMs can be high. Less effective on noisier datasets. Isn't suited to larger datasets. 	
LR		<ul style="list-style-type: none"> Easier to implement, interpret and very efficient to train. Small number of hyperparameters. 	<ul style="list-style-type: none"> Based on Assumption of linearity. Very sensitive to anomalies in the dataset. 	Generate false data injection
K-means	Unsupervised	<ul style="list-style-type: none"> Easy to understand and implement. Applicable to large data. 	<ul style="list-style-type: none"> Find the optimal parameter k. Dependent on initial values. Less effective on noisier datasets. 	Deduce the frame activities in the network
GAN		<ul style="list-style-type: none"> Produces very realistic data. No Markov chain Monte Carlo needed. 	<ul style="list-style-type: none"> Unstable to train. Huge computation. 	Generate false data to deceive a IoT network or machine learning algorithm.
MAB	Reinforcement	<ul style="list-style-type: none"> Online algorithm. No need prior information. 	<ul style="list-style-type: none"> Converging on the right solution can be slow. The learning process must start again when the environmental changes. 	Carry out an attack without prior information on the victim
Temporal-Difference Learning		<ul style="list-style-type: none"> Not require a model of the environment. Online algorithm. 	<ul style="list-style-type: none"> Less stable than Q-learning. May converge to the wrong solution. 	
Q-Learning		<ul style="list-style-type: none"> Not need to know the transition probability matrix. 	<ul style="list-style-type: none"> Behave poorly in some stochastic environments. 	
MLP	Deep Learning	<ul style="list-style-type: none"> Learns nonlinear models. possesses back propagation. 	<ul style="list-style-type: none"> Large amount of computation in the training phase. 	Deduces IoT information like activities or sensitive data
CNN		<ul style="list-style-type: none"> Reduces risk of over-learning. Very good feature extractors. 		

an attack with machine learning and the existing work on this subject.

III. MACHINE LEARNING ALGORITHMS AND THEIR PERFORMANCE

Recently, some smart attacks, namely attacks exploiting and integrating ML algorithms have started to raise the interest of the research community. In this section, we will present the common ML approaches that have been used for creating intelligent attacks. These attacks are detailed in Section V.

These machine learning algorithms can be classified into three different types: Supervised, Unsupervised and Reinforcement learning. Table VI summarizes the different machine learning algorithms mentioned below.

A. Machine Learning Algorithms

1) *Supervised Learning*: Supervised learning is one of the most common machine learning approach, explained in detail in [79]. Its purpose is to find a mapping function between an input variable X and an output variable Y based on examples (input and output pairs) so that:

$$Y = f(X).$$

It must create mapping functions to deduce the output variable Y from a new unknown entry X . To be able to develop these mapping functions, a supervised algorithm needs to practice beforehand using labeled data. Supervised learning approaches can be grouped into regression and classification problems.

- *Regression*: The output of the algorithm corresponds to a real value such as a price for example.
- *Classification*: The output of the algorithm corresponds to a category such as for example the gender of the population: female or male.

Many supervised algorithms are used to make attacks more optimal and allow an attacker to avoid solving statistical analysis problems manually. The supervised algorithms used to create smarter attacks are listed below.

K-nearest neighbors algorithm (KNN) [80]: Nicknamed nearest neighbors, K-NN algorithm, is used most of the time in classification problems, but it can also be applied for regression problems. In order to classify a new input, the system finds the K nearest neighbors among the training data set and retains the most represented category among these K neighbors. Several methods, detailed in [81], for calculating the shortest path are used, such as Euclidean distance, Manhattan distance, or Hamming distance.

Decision Tree Learning (DT) [82]: A decision tree allows you to break down a set of data into smaller and smaller subsets to create a decision tree that will include nodes (tests to be performed), branches (possible values of the test), and leaves (decisions made). It is built from a set of labeled data that includes attributes and classes. Many algorithms make it possible to build these trees. The two best known: IDS3 and C4.5 invented by Ross Quinlan are correctly described in [83].

Random Forest (RF) [84]: The random forest, as the name suggests, consists of a large number of individual decision trees that function as a set. A tree is constructed from a sub-sample drawn at random from the learning set. Each individual

tree in the random forest predicts a class, and the class with the most votes becomes the prediction. This technique solves the high variance estimator problem present in Decision Tree Learning.

Extra-Tree (ET) [85]: very similar to random forest algorithm, it differs only in the manner of constructing the decision tree. Indeed, during the conception of the decision tree, the split to divide the parent node into two random child nodes is made by a random value in a Extra-Tree Classifier. In a random forest, the best split is selected to divide the parent into the two most homogeneous child nodes.

Support Vector Machine (SVM): Introduced for the first time by Cortes and Vapnik in 1995, the support-vector machine has recently gained prominence [86]. This concept can be employed for both classification and regression problems, although it is more often used for classification issues. The idea of this algorithm is to find a hyperplane that best divides the data into n classes. When the data is not linear, and the hyperplane cannot be traced, a kernel function can be used. A kernel function transforms non-linear spaces into linear spaces. Several kernel functions exist, such as Polynomial kernel, Gaussian kernel, the most common the Gaussian radial basis function (RBF), and Sigmoid kernel, but it is also possible to specify custom kernels.

Linear Regression (LR) [87]: This algorithm permits to discover the relation between variables and forecasting. This method predicts a dependant variable values (y) based on a given independent variable (x). The goal is to find a linear relationship between the input (x) and the output (y).

2) Unsupervised Learning: Unsupervised Learning is based on non labeled training datasets. The algorithm must deduce the different patterns or clusters in the training datasets. Two problems can be distinguished in the unsupervised learning method.

- **Clustering:** the goal is to find common features to regroup data in many categories (cluster). For example, for a group of people, these criteria can be age or gender.
- **Association:** Association rules aim to uncover relationships between variables in large databases. For example, $\{Bread, Egg\} \Rightarrow \{Milk\}$. Here all the customers who bought bread and eggs also took milk; it is an association rule.

Above are explained the different unsupervised algorithms that have been used to create a smarter attack in Section V. Nevertheless, a more generalized overview of all unsupervised algorithms can be found in the book [88].

K-means [89]: K-means not to be confused with the K-nearest neighbors algorithm is based on a clustering problem. This algorithm aims to regroup in K distinct clusters the observations of the dataset. To compare the degree of similarity between the different observations, k-means uses the concept of dissimilarity distance. Thus, the smaller the distance is between two data points, the closer they are in similarity, and vice versa. Euclidean and Manhattan distances are the most widespread methods in the k-means problem to calculate the resemblance. The system to choose the right value for K is explained in [90].

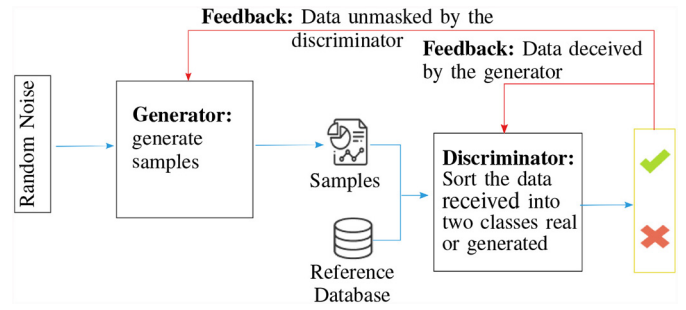


Fig. 2. GAN principles.

Generative Adversarial Network (GAN): Introduced by Goodfellow *et al.* in 2014, this unsupervised algorithm aims to generate new synthetic instances of data that can pass for real data [91]. GAN is composed of two deep networks: the generator and the discriminator, as show in Figure 2. Designed as a conventional neural network, the generator takes a random noise as input and produces new instances from it. Discriminator receives data from generator and database and tries to sense if samples are real or if they were generated. A backpropagation is used to improve network accuracy.

Indeed the generator receives a return on the data which did not succeed in deceiving the discriminator and the latter a report on the false sample, which was perceived as real.

3) Reinforcement Learning: Unlike the two categories above, reinforcement learning is not based on massive upstream data learning. It is an active learning method. In fact, an agent immersed in an environment makes decisions on actions to be carried out according to a current state. The environment will provide him with a reward in return. The agent's goal is to maximize the total rewards over time. Reinforcement learning is a technique widely used when creating smart attacks. This method is often used to find the optimal attack strategy in an unknown environment. This is why we will present here the main reinforcing algorithms mentioned in this survey. More information on each algorithm is given in the book [92].

Multi armed bandit (MAB): Generally a Multi armed bandit problem is modeled in the form of a Markov Decision Process which can be described thus in the form of 5-tuples $\langle S, A, P, R, \gamma \rangle$, where:

- S is a finite set of states s ;
- A is a finite set of actions a ;
- $P_a(s^n, s^{n+1})$ is the probability that an action a in state s^n in time $n + 1$;
- $R_a(s^n, s^{n+1})$ is the expected immediate reward received after transitioning state s^n to state s^{n+1} due to action a ;
- $\gamma \in [0, 1]$ is a discount factor.

The main objective of a MDP is to find a policy π that associates an action with each state $\pi : S \rightarrow A$ to maximize the reward.

Several multi-armed bandit algorithms exist as indicated in [93]. Here we mention those used to create smarter attacks. The main problem of multi armed bandit algorithms is to find a balance between the exploration and exploitation phase. The ϵ -greedy algorithm tries to answer this problem by choosing with a probability of $1 - \epsilon$ the best-known action (Exploitation)

and by selecting with a probability of ϵ an unknown action with a uniform law (Exploration).

Temporal-difference learning - Delayed RL [94]: Temporal-difference learning is a combination of Monte Carlo methods and dynamic programs. Like Monte Carlo methods, Temporal-difference methods can learn directly from raw experience without a model of the environment's dynamics and, like dynamic program, they can update their estimations based in part on other learned estimates.

Q-learning [95]: This very famous technique in reinforcement learning does not require any initial model of the environment. The functioning of this algorithm is based on a Q-table which allows to record for each chosen action its maximum reward. This is a variant of the Temporal-difference learning algorithm.

4) *Deep-Learning Algorithms*: Deep Learning is based on a network of artificial neurons inspired by the human brain. A neural network is generally composed of a succession of layers, each of which takes its inputs from the outputs of the previous one. Each layer is composed of k neurons interconnected with that of layer $n - 1$. Multiple deep-learning architecture exist, we will describe here the two most used architectures for the creation of intelligent attacks.

Multi-layer perceptron (MLP) [96]: Multi-layer perceptron consists of three different types of layers.

- Input layer: composed of several neurons, its goal is to present the data to the first hidden layer, it is a passive layer.
- Hidden layer: the multi-layer perceptron algorithm requires at least one hidden layer. These latter are located between the input and output layers where artificial neurons take in a set of weighted inputs and produce an output through an activation function.
- Output layer: corresponds to the last layer of the multi-layer perceptron algorithm. It is composed by one neuron which gives the final prediction.

They are often applied to supervised learning problems and permit to resolve non linear problems.

Conventional neural networks (CNN): Conventional neural networks are very similar to the multilayer perceptron algorithm, but it also has a convolutional layer based on convolutional filtering and a pooling layer. The first element is the heart of the Conventional neural networks. Indeed, this layer seeks to identify the presence of a pattern in order to adjust the parameters as quickly and as efficiently as possible. The pooling layer makes it possible to reduce the size of the data in order to keep only the most essential elements and thus limit the risk of over-learning.

B. Model Assessment and Selection

The performance of a machine learning model can be assessed thanks to various tools implemented by the community a few years ago. Many effective methods to estimate the reliability of a machine learning algorithm and to select the optimal parameters for its operation exist. In this part, we will only discuss the metrics that were used in the articles mentioned in Section V, therefore this list is not exhaustive. The

standard and basic metric employed in the academic literature is **accuracy**. It has the following definition:

$$accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP is a True Positive; model correctly predicted the positive class.
- FP is a False Positive; model incorrectly predicted the negative class.
- FN is a False Negative; model incorrectly predicts the negative class
- TN is a True Negative; model correctly predicts the negative class

In some specific cases, the accuracy may be too low an indicator to prove the performance of an algorithm. This can be the example when we work on a dataset with over-represented class. This is why the metric **precision**, formulated below, can be used:

$$precision = \frac{TP}{TP + FP}$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Thereby, compared to the accuracy metric, the precision metric is more efficient when we have imbalanced dataset.

More advanced metrics have also been considered in recent years as mentioned by Reich and Barai in [97]. **k-fold cross-validation**, a cross-validation method, is one of the most used in the articles listed below [98]. This basic and widely used approach is one of the least biased and permits to provide an estimator of the performance of the metric explained below. Before executing a machine learning algorithm, the common usage is to divide the data set into three subsets: the training data set, the validation data set, and the test data set. Nonetheless, this process can cause an overestimation or an underestimation of the results obtained. The k-fold cross-validation aims to answer this problem by dividing the original dataset D into k mutually exclusive subsets D_1, D_2, \dots, D_k of approximately equal sizes. The cross-validated estimate of the prediction error, $CV(D)$, is given as:

$$CV(D) = \frac{1}{K} \sum_{i=1}^K L(y_i, f_{-k}(x_i))$$

where f_{-k} is the model trained without the k^{th} subset of the learning set, and $f_{k(x_i)}$ is the predicted value for the real class label y_i , of case x_i , which is an element of the k^{th} subset. $L(y_i, f_{-k}(x_i))$ represents the *loss function* that quantifies the estimation error.

C. Discussion

In this section we present the most promising Machine learning algorithms in IoT attacks with their benefits,

disadvantages and applications. Other machine learning algorithms as well as other models assessment exist. However, we here focused on those employed in the academic literature to develop cyberattacks on IoT networks.

We can notice that each type of machine learning approach has a specific area of application. Indeed, supervised algorithms are employed to analyze information and therefore to infer data such as an encryption key, while reinforcement learning methods make it possible to deduce an optimal attack strategy. However, most of these algorithms still have many weaknesses such as a demand for significant energy and computing resources. Methods to resolve these issues are in progress, we will identify them in more detail for each type of attack in Sections V–VIII.

IV. ADVANTAGES AND EXPLOITATIONS OF ML-BASED ATTACKS

In this section, an overview of the advantages of ML-based attacks is given. From this analysis, we categorize the existing attacks according to the type of machine learning. Following this classification, a pattern emerges and we can see that each type of machine learning algorithm is used for very specific motivations.

A. Advantages of ML-Based Attacks

With the advance of machine learning over the past few years, many areas have known an evolution in terms of innovation and automation. Thereby, many means of detection and countermeasures have been developed in the field of cybersecurity. Unfortunately, the growth of machine learning is also a double-edged sword. Indeed the latter has become more and more affordable in recent years thanks to the development of tools like Tensorflow or OpenAI Gym, allowing easy implementation of algorithms [4], [99]. In addition, machines have become increasingly powerful and accessible on the market, so it has become easy for a person to use machine learning algorithms for malicious purposes.

There exist three types of possible changes in the threat landscape with the use of machine learning according to [100].

- *The expansion of existing attacks:* The goal is to improve certain aspects of current attacks, such as the speed of execution, reactivity, or automation of tasks that require human intervention.
- *The introduction of new threats:* The exploitation of particular vulnerabilities was not feasible before the arrival of machine learning because it required many human resources. Besides, the introduction of machine learning into the new means of detection has made it possible to create new vulnerabilities with adversarial machine learning attacks. Indeed, machine learning algorithms themselves present vulnerabilities, so by integrating them into the security systems new attack vectors are created in networks. These flaws can be exploited through adversarial machine learning attack.
- *Change of the typical character of threats:* Through the use of machine learning, an attack that has been until now specific for an IoT device on a particular type of

network can become more widespread. Indeed, nowadays the attackers are often forced to make a compromise between the number of targets and the effectiveness of their attack due to a lack of time. Most of the time, they abandon the fact of having an optimal strategy in order to aim more targets and consequently increase their rate of success. However, the arrival of machine learning in certain attacks could automate them more easily, thus making it possible to target more people with greater frequency. We think that the typical character of threats will change in a few distinct ways. Indeed, today, specific attacks may be generalized in the near future.

The malicious use of machine learning can be classified according to different aspects like the level of risk generated, the type of algorithm used or the level of ML's intelligence.

The primary purpose of using machine learning from an attacker's point of view is to best respond to the characteristics of an attack on an IoT network. Based on a comparative analysis of different attacks targeting existing communication protocols, we have identified some specificities common to these as follows.

- *Undetectable and untraceable:* The faster an attack, the less it is recognizable by a detection means and thus allows the attacker to replay it several times without being countered from the first attempt, or both.
- *Proactive:* An effective attack must foresee the behavior of the victims to be able to determine the optimal strategy to put in place beforehand.
- *Frugal:* A perfect attack must consume few resources (e.g., memory, energy) so that it is the least expensive possible to set up and achievable by everyone.
- *Adaptive:* The attack can be adapted according to the attack environment, such as the type of protocol, the network topology, or the kind of IoT device. An assault must also be able to attack one or more targets at the same time without requiring many changes.
- *Autonomous:* The objective is to avoid the maximum interaction with the human factor during its execution to avoid false manipulations, for example, and therefore, to be easily detected.
- *Robust:* Ideally, an attack should not be spotted and stopped by a known method of detection or countermeasure. During the conception of the latter, it must also be tested against protective means.
- *Rapid spread:* If the goal is to assault all the devices in the IoT network, this propagation must be done quickly in order to limit the detection of the latter. For this, it is possible to draw inspiration from the epidemic theory generally used to propagate code updates or sensitive information in wireless sensor networks. Several strategies exist such as the pull based epidemic algorithms, the push based epidemic algorithms and the combination of the two: the pull-push based epidemic algorithm [101].
- *Little knowledge:* A perfect attack should require very little basic knowledge in order to limit eavesdropping, upstream, on a network. It thus avoids being spotted before even leading the attack. The attacker must quickly

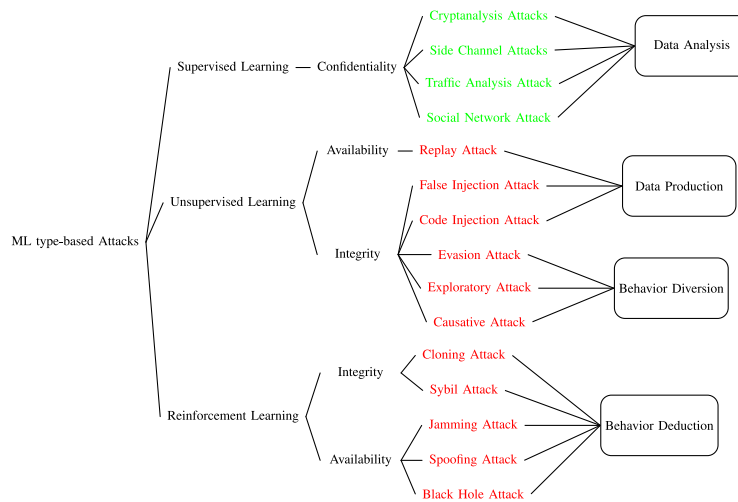


Fig. 3. The different exploitation of ML into attack generation, Colors indicate the implication of the attacker : Passive attacks, Active attacks.

learn the main features and components of the attack environment.

Due to the development of intelligent detection means [102] and the resistance of new protocols against conventional attacks, assailants are forced to increasingly create smart attacks to best respond to the characteristics mentioned above.

B. Exploitation of ML for Generating IoT Attacks

For all of these reasons, the integration of machine learning in the field of cyberattacks has become a new and open subject in the literature. The use of this technology could simplify certain phases when creating an attack like decision making or victim analysis. Consequently, in order to better answer and explain what ML brings to the creation of an attack we have established a taxonomy composed of three main categories each divided into several part, as we can see in Fig. 3.

1) *Classification Description*: For this classification we relied on the type of machine learning algorithm used to create smart attacks. Each category has been divided according to the intended security objective: Confidentiality, Integrity and Availability. Confidentiality regroups all attacks allowing access to personal information or to a system without prior authorization. Integrity attacks also operate in a similar fashion without authorization, however this time attempting to modify information instead (data, network configuration...). Availability attacks refer to all threats that allow a system to be taken out of service for authorized users. For each category, different examples of attacks are given. Moreover, by relying on this classification we were able to represent the implication of the attacker for each attack: Passive or Active. Most of the time passive attacks targets the confidentiality of a system. This kind of attack involves no interaction with the attacker and the target system and does not affect its function. On the other hand, active attacks, have a direct impact on the system, requiring the attackers full investment. Most often, these attacks target the integrity and availability of data or systems.

2) *Exploitation of Machine Learning Algorithms*: Based on this classification, a pattern emerges and we can see that each

type of machine learning algorithm is used for very specific motivations, as we can observe in Fig. 3.

- *Data analysis*: One of the primary reasons for choosing to use ML algorithms in creating attacks is to facilitate data analysis and decision making. Indeed in some attacks, this phase is often costly in terms of time and human resources and most of the time allows only one victim to be targeted at a time. The use of machine learning could make it possible to automate this phase and therefore save many resources. In these categories, two sub-objectives can be cited: the deduction of cryptography information like password and the selection of the most valuable target. Indeed, analyzing the data circulating on an IoT network or even certain information depending on an IoT device such as the energy it consumes, can lead an attacker to guess the cryptography keys. This is the case for side channel attack, where an attacker aims at extracting cryptographic information from an IoT device, through measurement and analysis of physical parameters [42]. The other goal of data analysis is the selection of the most important target on the IoT network and the best time to attack. By analyzing the traffic of the IoT network, we can deduce several pieces of information such as the frequency of use of an IoT device, its type and even the habits of users. This type of passive attack is called traffic analysis attack [39]. In certain cases, if the data is not encrypted the attacker can analysis and re-structure the information of the network. *Supervised algorithms* and more particularly classifications algorithm are often used for the data analysis. This method can be a huge help for the attacker because he can carry attacks more easily and quickly on several victims. However, for the moment the most of these attacks need a long preparation phase during their first use. Indeed in order to deduce certain information such as device type, it is necessary to train classification algorithms with a lot of data from various sources. Therefore, first the attacker must listen to, record and process the data of several IoT networks in order to design the envisioned attack.

- *Behavioral deduction:* Machine learning algorithms can also help analyze the behavior of an IoT network. Indeed, the development of an attack with ML can contribute to deduce the behavior of a victim in order to automate attack or to pretend to be a legal IoT device. By analysing the behavior of an IoT network, an attacker is capable of deduce the strategy and select the best moment of attack and the best parameters in order to be the least detectable possible. This information is very useful during active attacks, which directly disrupt the network such as during a jamming attack [38]. The aim of this attack is to voluntarily interfere the signal transmission to disrupt or prevent it. So in this case, to be as much undetectable as possible the attacker must have the same network parameters of the victim, therefore the fast acquiring of information about it, is essential. The use of ML becomes an important assistant because, by trying several configurations in an intelligent way, it will allow the attacker to create an attack strategy easily and save a lot of time. Moreover, the use of ML in creation of attack is more important when the dynamic environment modifies configuration quickly. Indeed, the study of the behavior being automated and faster, the attacker does not waste time developing a strategy as soon as a parameter of the environment changes. This method is not negligible because it can make it possible to avoid many decision errors and consequently reduces the probability of being detected. Inferring the functioning of an IoT network can also allow an attacker to copy the behavior of a legitimate node in IoT network. In this case the attacker can simply listen to the conversation or have a direct action on the network as sending false data. In the same way as for the first sub-objective; automating the behavior deduction allows an attacker to be less quickly detectable when the network configuration changes. It also makes it possible to react more quickly. In this category, we can find cloning, spoofing or Man-in-the-Middle attack. These different attacks intend to impersonate another IoT device by copying its behavior to gain access to private information or the entire networks. This objective is often done using *Reinforcement Learning algorithms* where the ML algorithm will determine the behavior and configuration of a network via a reward system. Therefore, it requires very little information and preparation (collecting data, sorting data, etc...) compared to and machine learning classification algorithms.
- *Data production:* Another benefit of integrating Machine Learning into attacks is the data creation. Indeed, many attacks in IoT systems rely on the creation of data in order to severely compromise their behavior. In this category, several sub-goals are present like the creation of content or the alteration of data. These two categories are in fact very similar but it differs in the fact that for the first, the attacker creates new information in order to deceive the IoT network while in the second the attacker modifies an existing data. Creating data can be very risky for an attacker because the data must be comparable to existing data so as not to alert defense systems. Some machine

learning algorithms like GANs permit data generation. They automate the production of data and therefore the creation of attacks based on it. This action diminishes the probability of being detected because by generating the data automatically, we limit the error rate related to humans (e.g., calculation or inattention errors). In this class, we can list the fake injection data attack where an attacker attempts to trick the system into making the wrong decision by injecting false data. Data creation can also help attacker to force security systems like password. Indeed, one of the most famous attacks to force passwords on IoT networks are brute force attacks [103]. This type of attack is based on the creation of a dictionary gathering the most popular and probable passwords and based on this the attacker tests all the possibilities. The creation of the dictionary is a long and boring step. But with the help of machine learning, the attacker must generate easily and quickly new patterns on dictionary. The other purpose is to modify the exiting data. In this case, an attacker intercepts the data to modify it in a certain way and reinjects it in order to disrupt the functioning of the network. ML applied for replay attacks can be used for this specific case [104]. Indeed, the latter aims to maliciously repeat a transmission which has intercepted and altering. This attack can lead to network overload or identity theft. Like for the creation of new data, the alteration must be less undetectable than possible. The use of machine learning algorithms therefore makes it possible to limit human errors. For this category, the most used machine learning algorithm is the *Generative Algorithm Network (GAN)*. This type of algorithm first requires the creation of a dataset. Therefore, before creating an attack, the attacker must first listen to the network for a while. In this class, we can list the fake injections. The latter aims to maliciously repeat a transmission which has intercepted or to generate new ones from it. This attack can lead to network overload or identity theft.

- *Behavioral diversion:* One of the latest motivation for integrating machine learning algorithms into attacks is to modify the behavior of a system in an IoT network. This last category mainly covers attacks of the adversarial machine type. This type of attack is not specific to IoT networks because it targets machine learning algorithms. Nevertheless, this new technology is increasingly used in the field of IoT allowing to solve many complex problems such as signal/traffic classification, resource management or security [105]. This growing application has given rise to new attack vectors within IoT networks. The goal of an adversarial machine learning attack is to deceive the machine learning algorithm by modifying the training data or introducing maliciously designed data. These modifications are mostly invisible to humans, but have a significantly impact the decisions made by algorithms. By falsifying the data, it becomes easy to force them to make decisions that will modify the initial behavior of the IoT device and which generate new flaws or new threats. Three types of adversarial attacks can be listed according to their process: the causative attack, the evasion attack

and the exploratory attack [78]. The first, also known as poisoning attack, focuses during the learning phase, and aims to reduce the reliability of the learning process by corrupting the data (by changing the label for example). During the exploratory attack, the attacker attempts to infer how the machine learning algorithms operate (e.g., the output and input of the algorithm). After such an attack it is possible for an opponent to proceed to an evasion attack where the main goal is to provide false input which will result in an incorrect output (e.g., label).

There are several main use cases of machine learning for creating attacks. One of the primary purposes of this use may be to help choose and target the victim. Indeed, several attacks need a precise and complete preparation phase. The ability of machine learning to interpret patterns in a large amount of data can be used to prepare and find the ideal target. The second objective can help automate attacks. In fact, the behavior of a network is not fixed over time and can change. The ability of a machine learning algorithm to react and process information quickly reduces the chances of an attacker being detected during this adaptation phase. The use of ML for the creation of attack can also allow the attacker to remain hidden. Indeed, the ability of ML to create new data very similar to existing data can allow an attacker to remain undetectable and effective.

C. Lessons Learned

In this section, we discuss promising ML algorithms in the field of attacks on IoT networks. Indeed, this technology can provide excellent assistance to the attacker.

- They mainly reduce the chances of being detected by being as proactive as possible and autonomous.
- They allow to be more and more adaptive. An attacker will not necessarily need to create different attacks for each communication protocol if the attack itself deduces its strategy according to the behavior of the network.
- They require little network knowledge from the attacker if the analysis is done by machine learning algorithms.

One of the main obstacles to creating attacks based on machine learning could be the induced cost by these algorithms (e.g., computation, energy). However, it is possible to design these algorithms outside of IoT networks and store them on serverless platforms [106]. This system could be suitable for passive attacks based on data analysis such as cryptanalysis attacks. Moreover, it is possible to integrate simple machine learning algorithms in IoT networks. For instance, a reinforcement learning method can be integrated on resource-constrained devices, in order to derive the optimal communication channel [107]. Finally, as federated learning and machine learning platforms are becoming more and more accessible, the use of such advances by attackers will obviously increase.

This is why it is urgent to start studying them and gain insight into the attacks that already exist.

V. SMART ATTACKS BASED ON DATA ANALYSIS

In this section we will detail smart attacks, namely attacks relying on machine learning approaches. In particular, we identify attacks using machine learning algorithms in order to

perform automatic data analysis and hence acquire information on the network. Several published works relating to two types of attacks that we explain in more detail just below have been carried out. For every kind of attack, we first define the attack, its objectives and its consequences. Then, we explain the type of learning approach that has been integrated for generating a smart attack.

We introduce alternative methods to counter them before eventually providing an overview of future feasible research in this area. The identified works are briefly summarized at the end of each subsection in a table according to diverse characteristics such as the algorithm used.

A. Privacy Attacks

1) *Machine Learning-Based Privacy Attacks*: The exponential growth in the number of connected objects in many areas in recent years has largely favored privacy threats. In [108], the authors classify the privacy threats in five categories.

- *Identification*: The objective is to link personal digital data such as a nickname or a telephone number to a natural person.
- *Localization and Tracking*: The goal is to locate in time and space a physical person or a device.
- *Profiling*: Here, an attacker collects numerous data from several sources to infer information about the target (e.g., behavior, habits).
- *Privacy-violating Interaction and Inventory Attack*: This corresponds to the transmission of sensitive information obtained without consent on public media (e.g., Web, newspapers).
- *Linkage*: By crossing data from several sources together, the attacker seeks to infer relevant information that was previously unavailable.

This type of threat can have serious consequences such as the physical intrusion of a person into an environment (a military base), blackmail or carry out active attacks such as jamming or spoofing. The advance of ML method has made traffic fingerprinting attack one of the most popular privacy attacks. This attack is so named because it aims to define patterns (fingerprints) from data in order to identify with high probability the network protocols, operating systems, hardware devices, software, among others. Applied to the IoT domain, the concept is as follows: an attacker listens to traffic on the IoT networks in order to guess what type of IoT device is used and to deduce information such as the type of protocol used or the activities and habits of an user.

Many researches integrating machine learning in this type of attack are in development on this subject nowadays. Indeed, sorting or correlating data are sometimes tedious and long tasks for an attacker. However, thanks to a machine learning algorithm to classify this data, the attack can become much faster and easier to execute. During our research on this subject, and to the best of our knowledge, only two levels of network protection have been tested to carry out this type of attack: an encrypted network and an encrypted network using additional protection means such as Virtual Private Network

(VPN) or proxies. We therefore sorted the papers relating to fingerprinting attacks using machine learning according to the level of protection installed on the networks tested.

- *Level of protection 1 (Encrypted data):* The first test environment is a protected wireless network, which means that an authentication protection or data encryption mechanism is used. This can be the standard WPA2 protocol in the case of WiFi for example [109]. The correlation between the events of IoT devices (on / off) and network activity are demonstrated in [110] by the authors. They confirmed their hypothesis on the two most common IoT devices of 2016: a thermostat, “Nest Thermostat,” and a “Nest Protect” smoke and carbon dioxide detector. They were able to determine the change of state between two thermostat modes at 88% accuracy. In other work, the authors prove that it is possible to recognize and identify 21 unique IoT devices present on a campus thanks to the use of the Random Forest algorithm [111]. To achieve this, they relied on 13 different characteristics, such as sleep time or the average packet size for each device. To validate the efficiency of their classifier, they used the 10-fold cross-validation method and obtained a high accuracy of 97%. They also achieve 95% reliability during independent testing. Thanks to these results, they were able to prove that it was possible to determine the type of IoT device on an encrypted network. These two previous hypotheses were further used to take into account fully encrypted data [112]. Therefore, as the attacker can only rely on the packet headers (metadata), the sole payload does not allow finding new features available that would identify a device (e.g., traffic volume). Five other classification algorithms are compared: K-nearest Neighbors, Support Vector Machine, Random Forest, Adaboost and Extra-Trees to improve attack performance. They find a significant increase in an average accuracy of 18.5% with the AB learning model and an execution acceleration rate of 18.39 times with the K-NN algorithm compared to the random algorithm forest used in [112]. It has been demonstrated that a simple classification algorithm can help deduce the type of device easily, thus proving that each change of activity of an IoT device (switch on / off) can be correlated with a change in traffic rate [113]. To be able to guess which device a user has, they use a K-NN (3-nearest-neighbors) classification algorithm with the traffic volume of each device as input. By validating the reliability of their algorithm thanks to 10-fold stratified cross-validation, an accuracy of 95% was obtained. Similar works have been done, where the authors set up a two-stage classification in [114]. The first is to recognize the type of device, and the second is its state. They test their algorithm with data based on a real environment that takes into account IoT and non-IoT devices. For the first objective, they evaluate the performance of three classification algorithms: K-NN, Decision Tree (DT), and Random Forest (RF). They get better results (92% accuracy) for K-NN, but it takes longer to run (5 to 13 times). The second point is achieved thanks to the help of two types of algorithms DT

and RF, which obtain similar performances (97% accuracy). Fingerprinting attacks have also been studied, by taking into account several types of IoT devices operating on several different protocols (e.g., Bluetooth Low Energy, WiFi, ZigBee) [115]. Not only is it possible to guess what type of IoT device is used in a given environment, but also the activity of the device (on / off) can be identified. Thus, using a machine learning algorithm can help deduce the activity of a person in an environment (e.g., presence in a room of a house). To solve this problem, a new attack was proposed: ‘multi-stage privacy attack’, which is divided into four stages dependent on each other. The first phase is the distinction of the type of IoT device used in the environment thanks to a K-NN algorithm. The next step is the distinction of the transition state. For this, a compromised characteristic vector of three variables is extracted: the average length of the packets, the average time between arrivals, the median absolute deviation of the size of the packets. Two supervised learning algorithms are also applied: the Random Forest classifier and the K-Nearest Neighbors (K-NN) classifier. These two types of algorithms have similar performances, the accuracy rate is around 90%. The third step is the decision on the state of the device (activated or deactivated) with a random forest classification. The authors obtained an accuracy of 92% in detecting the state of the device. The final step is to infer user activity in the smart home by observing the state of all devices with an unsupervised Hidden Markov model (HMM). They deduce that an attacker can know the activity of a resident of a smart home to 95%. So here the authors automate the IoT device detection thanks to machine learning and uses HMM in order to guess the actions and activities of a person.

- *Level of protection 2: (Encrypted data and defense methods):* Many techniques to counter privacy attacks have been implemented in recent years, such as the use of Virtual Private Network (VPN) or Network Address and Port Translation (NAPT). To best respond to reality, new research on privacy attack in a network protected by encryption and transformed by a network gateway has been tested. One of the first investigations was that of Dong *et al.* in which the latter sought to guess the type of IoT device in an encrypted network protected by a VPN and a NAPT. Due to this environment, researchers had to restrict the number of data to identify a device [116]. In fact, unlike the work carried out in [113], the data containing the domain name in DNS response and the destination IP can no longer be used. As there is a dependency between the packets according to the operating states, the authors set up a short-term memory neuron network (LSTM-RNN) to determine the type of a device. Their final classification has for reliability a result of 92% on an encrypted system using a VPN or NATP. A study on information leaks from a network that may be useful for conducting fingerprinting attacks is provided in [117]. Their analysis covers 81 IoT devices deployed on two different continents and on data collected during different

TABLE VII
PRIVACY ATTACKS WITH ML

Ref	Date	Protection level	Heterogeneous networks	ML Algorithms	Remarks
[113]	2017	Data encrypted	Yes	RF	Distinguish IoT from non-IoT traffic and recognize 21 unique IoT devices.
[114]	2019	Data encrypted	Yes	K-NN, SVM, RF, AD, ET	Compare five classification algorithms and conclude that Adaboost (AD) has the best accuracy
[115]	2017	Data encrypted	No	K-NN	Device and Activity identification
[116]	2019	Data encrypted	Yes	DT,RF,K-NN	Set up a 2-stage classification in a real environment.
[117]	2018	Data encrypted	No	K-NN, RF, HMM	Deduce the actions and activities of a person. Take into account several protocols: Bluetooth, Wifi, ZigBee.
[118]	2019	Data encrypted and defense methods	Yes	LSTM-RNN	Realistic network environment, where common techniques like NAPT and VPN are enabled.
[120]	2019	Data encrypted and defense methods	Yes	DBSCAN	Automatically extract packet-level signatures and infer the type of IoT device
[119]	2019	Data encrypted and defense methods	No	RF	Study on 81 different IoT devices.

phases, such as when IoT devices are started up interactions and the device inactivity phase. After a study of the data obtained and thanks to a classification algorithm (Random Forests), they obtain that it is possible to guess the activity of IoT devices even on a network that uses protection means such as VPN. In [118], the authors have the same objectives as [115] but develop a new attack more resistant to detection methods and countermeasures such as traffic shaping. For this, the authors have created a ‘Ping-Pong’ tool, which makes it possible to automate the collection of data and to deduce the type of activity of a device. This new technique is called ping-pong because, during their data analysis, the authors noticed that before each event, an event packet of predictable length was sent. It corresponds to a request packet from the device or the server (Ping) and a reply packet back to the device/server (Pong). In most cases, the length of the packets is different depending on the type of device, so it can be used to infer the type of device and its activity. Ping pong has two components: Training and Detection. The first one automates data collection by collecting, filtering, and then sorting it using a supervised learning algorithm: DBSCAN. The purpose of this step is to create a list that contains signatures. The second phase detects the type of device by matching a network trace with a signature contained in the file. Using this method, they can guess the device category at 97% over a secure network.

The researches above have shown that it is possible to improve fingerprinting attacks through the use of machine learning algorithms. This makes it possible to considerably reduce human analysis while increasing the rate of precision of attacks. In Table VII we summarize the most representative papers for privacy attacks exploiting machine learning approaches.

2) *Countermeasures*: Due to its passive nature (the assailant does not need to modify the transmission), the traffic fingerprinting attack is very complex to detect. However, there are a few strategies to mitigate this type of attack, such as blocking traffic, tunneling traffic, or traffic shaping. Blocking traffic is a relatively naive method that consists of using a firewall that prevents specific flows from leaving the local network. However, this method quickly becomes unusable when IoT devices need to collaborate with cloud services. The second is to route all network traffic through a proxy or

VPN to mask the destination IP address of the traffic. Thus the attacker has less sensitive information and has more difficulty in separating the traffic. However, as seen above, new means make it possible to identify a device despite the use of a VPN. The traffic shaping strategy remains the most reliable these days, but it is known to limit network performance by adding processing overhead. Another approach is to inject false data to generate false activity.

3) *Challenges in ML-Based Privacy Attacks*: This type of attack is still an open subject of research and includes many areas for improvement such as:

- *Network diversity*: Most of the attacks presented above are not tested on any large network which would take into account IoT and non-IoT devices using many protocols (e.g., WiFi, BLE, ZigBee).
- *Multi-user vs. single user*: All of the above work enabling the deduction of a person’s activity in an environment using implanted IoT devices is only tested with one user. However, it is often likely that an environment has multiple users; for example, a smart home can be controlled by numerous residents. Ignoring that an environment can have multiple users can lead to many false positive or negative during an machine learning-based fingerprinting network attack.
- *Local vs. remote control*: Much research has been done by listening to the network locally. This configuration, therefore, does not correspond to reality because an attacker needs to infiltrate beforehand on the system.
- *Requires basic knowledge of the attack environment*: Using supervised classification algorithms requires training and testing them upstream. However, this means having a basic knowledge of the IoT devices installed in the environment to be attacked or having a database that is regularly updated with all the traffic characteristics of the IoT devices existing on the market. One of the future works could be to reduce this need for necessary knowledge before carrying out an attack. In addition, autonomous collection and sorting systems could be envisaged.
- *Accuracy vs Learning speed*: The algorithms used to carry out these types of attacks provide new solutions to improve the precision of the attack or its speed. Nowadays an attacker still has to choose between these two parameters to lead an attack. A future line of

improvement could be to take into account its two parameters jointly to carry out an fingerprinting attack.

B. Side Channel Attacks

1) *ML-Based Side Channel Attacks*: With the optimization of micro-controllers (MCUs) in recent years, an increasing number of IoT devices has entered our lives. These equipments collect and propagate many sensitive, personal and private data. To satisfy the need for confidentiality, cryptography algorithms such as AES or 3DES have been implemented within micro-controllers [119]. Their objective is to cypher the information transmitted, often using a key, in such a way that only the receiver can read it. However, devices are for most of the time vulnerable to a cryptography attack known as a side-channel attack. This threat aims to deduce sensitive data by exploiting information leakage from the physical implementations of cryptographic algorithms. Analyses can be based on execution time, power consumption or electromagnetic emissions, light emissions or cache behavior of a program.

This attack threatens the privacy of IoT device users and can have serious consequences in certain areas where data must remain private such as the medical or military sectors. Besides, many manufacturers like Philips use these algorithms to encrypt and authenticate each new firmware update on an IoT device. Through the use of a side channel attack, an attacker can discover cryptographic primitives, thereby providing them the opportunity to inject valid malicious updates [120]. This kind of attack, therefore, has consequences on the confidentiality of data but can also lead to more serious results such as a denial of service.

Two classes of side channel attack can be distinguished:

- The first is the **profiling side-channel attack**, which is currently the most widespread and the most used. The attacker possesses a device similar to the target device to collect information like executions traces and elaborates its strategies in advance. A profiling side-channel attack proceeds in two stages. The first called *profiling phase* consists of analyzing leaks on a model, and the second, *attack phase*, exploits these latter on the target to extract the information-dependent key. Many profiling attack approaches have been introduced in the literature, such as the Attack Templates and its alternative Stochastic Model (also called Linear Regression Analysis) [121].
- The second category is **non-profiling side-channel attack**. The attacker does not have access to a similar device and must focus only on exploitable physical leakages of the target. The Differential Power Analysis (DPA), the Correlation Power Analysis (CPA), and the Mutual Information Analysis (MIA) are methods for the non-profiling side-channel attacks [122]–[124].

However, the collection and the statistical analysis of data are very tedious tasks for attackers and can easily be turned into a classification problem. This is why new and more effective attacks are developed thanks to the use of machine learning. In the course of our research, we have seen that three different approaches had been studied: the use of supervised

algorithms of relatively simple classification such as Random Forest or SVM, the use of the multi-layer perceptron algorithm and the use of the conventional neural network. All the studies identified relate to power analysis.

- *Side-Channel Attack and Classification Algorithms*:

Many side-channel attacks based on conventional machine learning classification algorithms have been developed over the past decade. Indeed, it is possible to infer the 16 bytes of an AES-128 key by using a Random Forest algorithm as shown by the authors in [125]. This method dramatically outperforms the Template Attack method. The bytes of a secret key of a symmetric 3DES encryption and asymmetric RSA-512 encryption were identified with the help of Random Forest and SVM algorithm [126]. This kind of attack is more effective compared to a Template Attack when only a few traces of energy consumption are available. These results are also attested by deducing the 3DES key using a random forest algorithm [127]. Complementary work was also carried out with a study of the conditions under which attacks based on machine learning can overcome the Attack template [128]. Their studies focus on the curve of the dimensionality problem. This problem, identified for the first time by Richard Bellman, highlights the fact that it is important to choose the right model based on the amount of data and features available by the attacker. For this, they analyze the consequences of surplus or deficit of information during an attack carried out with a template attack or a classification algorithm. They conclude that, in theory, if the data is well-sampled thanks to the *Point of Interest selection* method, for example, it is more advisable to use a Template Attack. However, if a large amount of data is available or if they are not relevant, it is preferable to conduct a side channel attack with a machine learning classification algorithm. This theory is later revoked in [129]. The authors prove that an attack based on a machine learning algorithm can be as effective as a Template attack even if the data is well selected. Their experiments relate to the study of the performances of four types of classification algorithms, which are Sequential Minimal Optimization (SMO), Random Forest (RF), Rotation Forest (RTF), and MultiBoost (MB). They conclude that with a pre-processing data phase, it is more judicious to choose SMO. Indeed, with a proper configuration and preparation of features upstream, the SMO algorithm has a higher success rate (91.1%) than with a Template Attack (73.44%) even with few data. However, Rotation Forest (RTF) must be taken into account if the attacker does not have much time to pre-process the data.
- *Side Channel Attack and Multi-layer Perceptron learning (MLP)*: As detailed above, data pre-processing is mandatory during a side-channel attack with a classification algorithm. Indeed the categorization and selecting data are crucial steps that can significantly affect the performance of a side-channel attack. These points still require human intervention, which can have a severe impact on the results. On the other hand, deep learning techniques allow features to be

TABLE VIII
SIDE CHANNEL ATTACKS WITH ML

Ref	Date	Type	Target	Countermeasures	ML Algorithm	Remarks
[127]	2014	Profiling	AES	None	RF	Random Forest correctly extracted all 16 bytes of the AES key.
[128]	2014	Profiling	3DES, RSA	None	RF, SVM	Shows that for the 3DES and RSA key, machine learning algorithms improve the accuracy of the side-channel attack.
[129]	2018	Profiling	3DES	None	SVM, RF	Comparison among different algorithms and feature selection methods.
[130]	2015	Profiling	AES	None	SVM, RF	Study on the Curse of Dimensionality with machine learning based side-channel attack.
[131]	2017	Profiling	AES	Masking	SVM, RF, RTE, MB	Comparison among different algorithms according to the number of traces available. Traces are protected by a masking method.
[133]	2015	Profiling	AES	Masking	MLP	One MLP is used to remove data randomization due to the mask and a second to deduce the encryption key.
[134]	2015	Profiling	AES	Masking	MLP	Find the secret key of the masked AES only with 23 traces.
[135]	2015	Profiling	AES	Masking	MLP	Adversary needs about 18 guesses to determine the correct secret key using the original implementation of the MLP attack.
[132]	2016	Profiling	AES	Masking	MLP, CNN	Shows that our proposed DL-based attacks are more efficient than the ML-based.
[138]	2017	Profiling	AES	Masking, Hiding	CNN	Implements CNN-based side channel attack against jitter-based countermeasures.
[139]	2018	Profiling	AES	Masking, Hiding	CNN	Comparison between CNN-based side channel attack and Template attack.
[140]	2019	Profiling	AES	None	CNN	A practical side channel attack of a LoRaWAN module.
[141]	2018	Profiling	AES	Masking, Hiding	CNN	Creation of public dataset: ASCAD.
[142]	2018	Profiling	AES	None	CNN	New kind of layer for neural networks called 'Spread layer', it speeds up the learning phase.
[143]	2017	Non profiling	AES	Masking, Hiding	MLP, CNN	The first non-profiling side channel attacks with deep learning algorithm.

selected automatically from data. Consequently, applying deep learning algorithms to side-channel attacks would reduce the human factor and increase its stability and performance. Multi layer perceptron was the first algorithm used to set up this type of assault. Indeed, the effectiveness of an algorithm based on MLP deep learning and SVM / RF algorithms to carry out this type of attack is compared in [130]. An MLP algorithm is used to easily analyze and break AES hardware and software implementation. A 'AES-rotating Sbox masking' (RSM) architecture was further implemented [131]–[133]. Rotating Sbox masking is a countermeasure of side-channel attack rested on the mask method [134]. This method removes correlations between information leakages and sensitive values such as encryption keys by randomly dividing them into k bytes. It is achievable to deduce the encryption key even if this latter is protecting by a mask system with an MLP algorithm [131], [132]. They show that if the adversaries have access to the mask values during the profiling phase, MLP can be used to eliminate the randomization of the values and thus leads to the construction of another MLP algorithm allowing to recover the encryption key. They prove that the error rate of this method is lower than that used with an SVM algorithm (90% against 91.75%). Finally, it has appeared also possible to break the AES key, without knowing any information beforehand on the mask [130].

- *Side Channel Attack and Convolutional Neural Networks (CNNs)*: Sometimes, environmental noise as well as countermeasures implemented on wireless channels can distort traces which are useful for creating side-channel attacks. In addition, it is widely known that Convolutional Neural Networks (CNNs) perform better than MLP when information is perverted. Many studies have shown the effectiveness of CNNs in problems where data are easily distorted, such as image classification [135]. This is why many works have been based on CNN deep learning algorithms to create side channel attacks. Numerous

papers demonstrate the robustness of CNN side-channel attack against countermeasures such as masking or hiding [130], [136]. This type of protection creates misalignment in the traces, for example, by inserting random information (masking) or by generating an unstable clock signal (jitter). However, the application of CNN algorithms can overcome these difficulties. The comparison of this type of algorithm with other SCA algorithms (e.g., template attacks) have exhibited better results for CNN-DL algorithms [137]. This could be confirmed by the inference of an AES key used in the LoRaWAN protocols, with less than 100 transmissions, thanks to a CNN algorithm [138]. The issue of reproducibility of this attack during the experiments by proposing a complete study of SCA attacks using deep learning techniques is highlighted in [139]. The authors list their results in a database: ASCAD. Investigations are based on this dataset to improve the learning speed of neural networks applied to side-channel attacks [140]. The speed of learning is enhanced by implementing a new type of layer for neural networks called the 'Spread layer'. All the studies seen above are carried out on the profiling side-channel attacks. Nonetheless, we were able to identify one article which relates to a non-profiled side channel attack [141]. The authors rely on a CNN-deep learning algorithm and the ASCAD base and prove that it is possible to overcome the results obtained with simple non-profiled attacks like CPA.

Many studies have been conducted on side-channel attack based on machine learning. The first studies make use of basic classification algorithms and keeps requiring human expertise to be able to sort out the data and select the interesting features. These tedious actions could easily lead to false positives. However, the arrival of deep learning, and more specifically CNNs, has achieved to automate the preprocessing of data and to improve the robustness of attacks when faced with means of protection such as masking. Table VIII gives a summary of the Machine learning based side-channel attack studies.

2) *Countermeasures*: Diverse countermeasures have been put in place over the years against this type of attack. The first is *masking* (also called secret sharing), an accessible and relatively inexpensive mitigation strategy to eliminate the statistical dependence between sensitive data and emissions from secondary channels. Its purpose is to remove possible correlations between information leaks and private values.

The idea is to randomly divide the confidential values into several parts so that the opponent cannot deduce any information on the private value with a single element. The second is *hiding* and can be implemented directly at the hardware level with the use of circuits limiting leaks as with Dual Rail or by adding noise [142]. But also at the software level by synchronizing the observations in the time domain. This can be done by incorporating random delays or executing the instructions in random order. It is worth to notice that the IoT devices manufacturers must take into account available countermeasures which are often overlooked when they elaborate an IoT module.

3) *Open Challenges in ML-Based Side Channel Attacks*: There is still a lot of work to be done on this subject such as:

- *Selection and reduction of the number of traces*: Although the creation of Side channel attack with the Convolutional Neural Networks allowed to reduce the number of the traces necessary for an attacker to infer the encryption key, this latter can be further reduced. Consequently, the attack could be carried out more quickly because the attacker would require less time to recover the traces, he would then be less detectable and would consume fewer resources (e.g., energy, CPU).
- *The creation of a framework*: The idea would be to have a single algorithm possibly created with a machine learning method to infer the encryption key according to various possible scenarios (different cypher algorithm, different frames). It would also be interesting to test the solutions implemented in the above work against the detection means built up to date.

C. Lessons Learned

This paragraph discusses the use of machine learning to analyze the data produced by IoT networks in order to deduce sensitive information and to carry out attacks. The two attacks mentioned above already existed in the threat landscape, but the integration of ML made them simpler and increased their probability of success. We can confirm that data analysis with machine learning applies very well for passive attacks for several advantages.

- Most of these attacks involve listening to the network in order to process and infer information.
- The main methods used in these cases are classification algorithms which are relatively easy to set up for an attacker and available nowadays in many open source platforms dedicated to machine learning.
- Finally, their high energy and computation consumption is not a brake because these attacks are not necessarily executed on the target IoT network. Indeed, it is quite possible to recover the data, to analyze them thanks to

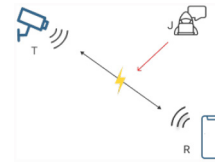


Fig. 4. Process of jamming attack.

one of the algorithms seen above on another environment initially and to use its result in a second time to carry out the attack.

However, much progress is still possible in these attacks such as:

- Developments directly associated to classification algorithms such as their learning speed or the number of data required for training.
- Improvements related to the attacks themselves as for the Traffic Analysis attacks which are tested only by taking into account that an IoT device belongs to a user which is rarely the case.

Although these attacks mainly use ML algorithms to analyze data and derive sensible information from it, we will see in the next section that these can also be used to deduce a behavior or an environment of a network.

VI. SMART ATTACKS BASED ON BEHAVIORAL DETECTION

One of the other motivations for using ML algorithms in attacks is to deduce the behavior and the environment of the network such as the communication or routing protocol adopted in order to deduce the optimal attack strategy. As in the previous section, we will discuss the ML-based attack, the target of the attack, the results and the open challenges in this field.

A. Jamming Attacks

1) *ML-Based Jamming Attacks*: Many wireless protocols have been widely proposed to interconnect and develop IoT networks in recent years. However, the inherent openness of wireless communication techniques has made them vulnerable to jamming attacks. This kind of attack consists of intentionally interfering with the communication medium to keep it occupied or to corrupt a signal transmission [143]. As we can see in Figure 4, the attacker (J) transmits a high range signal to disrupt communication between the transmitter (T) and the receiver (R). The goal is to prevent the exchange between the legitimate nodes of the network (T and R) by voluntarily occupying the channel or by causing a collision in order to force T to re-emit. The jamming attack is a type of denial-of-service (DoS) attack because it prevents proper network function and depletes essential resources such as node's batteries. When the primary purpose is to expend all the available energy of an IoT device, we are dealing here with a more specific attack, referred to as denial-of-sleep. Jamming attacks are mostly conducted at the physical and MAC layer, but sometimes cross-layer attacks are possible too. Depending on the strategy of the aggressor, the consequences can be felt on a single device or an entire network.

TABLE IX
JAMMING ATTACKS WITH ML

Ref	Date	Environment	Test Environment	ML Algorithm	Remarks
[149]	2014	Aware of protocol	Simulation	RL Delayed	Determine the optimal frame to jam in RTS-CTS handshake mechanism
[148]	2018	Aware of protocol	Simulation	DNN	Predict the length of the frame transmitted
[150], [169]	2016	unconscious of the protocol	Simulation	Jamming Bandit	Create their own online reinforcement learning algorithm: "Jamming Bandit".
[151]	2017	unconscious of the protocol	Simulation	ϵ -Bandit	Find new standards reward: change of power and the enduring time
[152]	2018	unconscious of the protocol	Simulation	Q-learning	Take into account the previous state of the network
[153]	2019	unconscious of the protocol	Simulation	OMP, MAB	Reduce the learning time in three interactions with the environment.
[156]	2019	unconscious of the protocol and dynamic environment	Simulation	Apprenticeship	A few interactions are required to converge in the optimal strategy.
[155]	2019	unconscious of the protocol and dynamic environment	Real testbed	Q-learning	Experience in a real testbed on a multiple-input and multiple-output (MIMO).
[160]	2018	unconscious of the protocol and dynamic environment	Simulation	Q-learning	Bypass an intelligent Deep reinforcement learning-based anti-jamming method.
[154]	2019	unconscious of the protocol and dynamic environment	Simulation	Q-learning	Find the channel to jam in multi channel environment
[157], [158]	2018	unconscious of the protocol and dynamic environment	Simulation	DNN	Studies based on cognitive radio.
[159]	2018	unconscious of the protocol and dynamic environment	Real testbed	K-means	Implement reactive jamming attack over CTC

We can classify jamming attacks into four categories according to their approaches mentioned in [144].

- *Constant jammer*: The goal is to continuously transmit on the wireless medium a radio signal composed of a random bit without following a MAC protocol. Its primary objective is to occupy the transmission channel.
- *Deceptive jammer*: This is a slight improvement from the previous strategy. Indeed, legitimate packets are sent by the attacker in place of random bits. Due to this modification, this attack is a little more difficult to detect, but it is still easily repairable because the frames are sent continuously.
- *Random jammer*: Here, the jammer alternates two states: sleep and active. It only jams when it is in the second phase by choosing its jamming strategy (constant or deceptive). The fact of changing its state allows it to conserve energy and to be less detectable.
- *Reactive/Intelligent jammer*: The jammer's aim is to be the least detectable by being as reactive as possible. It only jams when activity on a channel has been detected, which reduces its attack time and increases its efficiency.

However, attackers use the first three classes less and less. Indeed, from a defense point of view, they are straightforward to counter and detect thanks to modern detection methods, and the recent protocols implemented today on networks. From an offensive point of view, they, therefore, become more and more inefficient and energy-consuming. For these different reasons, proactive attacks are more and more selected. Notwithstanding, choosing the optimal time to jam a network without knowing the protocol and by consuming little energy can be a very complex problem. Machine learning can help

solve this problem by first discovering the network topology and protocol and then secondly by determining an optimal attack strategy. Machine learning based jamming attack is an open and constantly evolving research subject.

All the articles mentioned below have been classified in Table IX. The knowledge acquired by the attacker can vary the result of the attack, as seen in [145]. During our research, we have identified three different levels of knowledge of the attacker. We have therefore defined three sub-categories which are: 1) The attacker knows the protocol used in the network; 2) The jammer is oblivious to the configuration of the network, and 3) The assailant works in a dynamic environment.

- *The attacker is aware of the protocol*: The attackers are aware of the protocol and topology implemented in the network. The objective of these preliminary works was to prove that the use of machine learning could increase the performance of a jamming attack. Indeed work proves that it is possible by using a Deep neural network algorithm and by knowing the MAC protocol implemented in the network to predict the length of the frame transmitted and to jam the channel only during this period [146]. Consequently, by using machine learning, the attack time can be reduced as well as the attacker's battery consumption. In another investigation, the attacker is aware that the network to be jammed uses an 802.11-type protocol with the Request to Send-Clear to Send (RTS-CTS) handshake mechanism [147]. According to this knowledge, he develops a Markov decision protocol (MDP) model based on the messages exchanged between a transmitter and a receiver. Thanks to this MDP state transition structure, the aggressor tries to learn the best strategy to limit energy

consumption by using a delayed reinforcement learning algorithm. The goal here is to determine which exchange pattern (RTS / CTS / DATA / Acknowledged ACK) is the most efficient to jam in terms of energy consumption and the probability of success. After simulating their result on the OPNET platform, authors have deduced that for the case tested, the best strategy is to jam the CTS frame. They compare this approach with a classic jamming method (constant, random) and deduce that they get a better jammer success with the delayed reinforcement learning algorithm.

- *The jammer is unconscious of the protocol used:* The initial researches were done taking into account that the attackers have a large number of preliminary information about the strategy used by the transmitter and the receiver like, e.g., the protocol used, the topology of the network. However, in a real context, having such a kind of knowledge is very complicated and researches based on jammer attacks without any knowledge have been developed consequently. In the previous section, the attack strategies are defined on the basis of prior information. While the first works on this subject have recently emerged, they consider that it is not achievable to know in advance the communication protocol used by the network, in an electronic warfare context [147], [148]. An online reinforcement learning algorithm was introduced: 'Jamming Bandit', which allows finding an optimal attack strategy while having a reasonable computational complexity. For this, they invent a new multi-bandit problem-solving approach. The goal is to find the ideal modulation scheme, power level, and duration of the pulsation to jam the transmission between the transmitter and the receiver. They simulated and compared their solution with the most common method of solving a multi-armed bandit problem: the ϵ -greedy algorithm and prove that their alternative converges faster to find the optimal solution. After testing their algorithm on a single and several victims, they demonstrate the superiority of their algorithm in terms of convergence in both scenarios. However, some investigations highlight the need for ACK/NACK frames as a reward for solving the multi bandit problem [148]. Still, the latter are impossible to obtain in a specific context like in the military domain, for example. Researches relied on that of previous article proves that it is possible to base only on standard rewards for all types of environment (e.g., civilian, military) [149]. They find two new kinds of reward, which are the change of power and the enduring time. Indeed they noticed that during successful jamming, the power increases. Also, experiments have demonstrated that the proposed algorithm converges faster to the optimal solution than that used with ϵ -greedy, except when the discriminating parameter of the ϵ -greedy algorithm is optimal (M) [148]. The objective thus becomes finding the optimal discriminant parameter (M) in the first time, in order to apply the ϵ -greedy resolution algorithm in a second step [149]. Thanks to simulation, they compare their results with the 'Jamming bandit' algorithm and show that their method

converges faster towards an optimal jamming solution. Research is carried out even further, taking into account the fact that the behavior of a network to be jammed does not only depend on its current state but also on its previous states in [150]. Indeed, a transmitter-receiver pair can choose the encoding modulation according to the result of the previous transmission. Simulation results show that using a Deep Q-learning paired with recurrent neural networks while taking into account the old and current actions of the network, leads to better results during decision making (decision of jamming or not jamming the network with the right parameters). In order to be effective, a jammer must be as proactive as possible and have the shortest learning phase so as not to consume a lot of energy resources. Work aims to reduce the learning phase as shown in [151]. To address this problem, they combine the advantages of an orthogonal matching pursuit system (OMP) and a multi-agent system (MAB). They conduct the simulation in the MATLAB environment and reduce the learning time in three interactions.

- *The jammer is unaware of the protocol used and the environment is adaptive:* All the previously mentioned studies are mainly based on a static environment, like a pair of transceivers communicating through a single channel. Yet, in reality, two nodes can possess several channels (named multi-channel mechanism) to communicate in order to overcome interference and thus enhance the overall network performance. The searches try to improve the previous works by using multi-channel support in [152]. Here the main objective of the jammer is to find the right channel before carrying out its attack. They set up a low-complexity Q-learning algorithm and prove by simulation that finding the right channel to jam is achievable. Another investigation answers a fundamental question, which is that no mathematical model allows comparing the different types of jamming attacks in terms of efficiency and energy used [153]. They demonstrate a theorem based on three existing advanced jamming strategies, which are barrage, pilot, and ACK jamming. They conclude that with specific data (e.g., length of the frame emitted), it is preferable to choose a certain strategy. Since this information is most often unknown in advance, they use a Q-learning algorithm to find the optimal strategy and the channel. A real testbed was used to test a multiple-input and multiple-output (MIMO) network where the transmitter must choose the channel before each transmission. Reinforcement learning algorithms may take time to converge on the optimal solution. This is the case for searches [148] and [149]. Generally, this learning phase requires a lot of interactions. For example, 2×10^5 iterations are needed to find its ideal tactic in [148]. In addition, if the environment changes during this time, the jammer must restart its learning procedure. So this learning system (RL) is not effective in face of an adaptive environment. To provide a solution to this problem, Zhuansun *et al.* set up a new system: Apprenticeship learning [154]. Thanks to this method, few interactions are useful to converge towards

the perfect solution. This method is, therefore, advantageous in terms of efficiency and energy expended. Authors further reduce the gap by taking into account a cognitive transmitter in [155] and [156]. This equipment is able to automatically adapt its parameters according to its environment. Indeed through a deep learning algorithm, the transmitter can predict the optimal moment to realize a success transmission. The jammer also uses a deep learning algorithm to deduce the success of the transmission. This mechanism allows it to only disturb the channel if it is of interest (when the communication will be a success); therefore, it consumes less energy and makes it almost undetectable. In addition, to limit the time for collecting useful data throughout the learning phase, the authors exploited a Generative Adversarial Network (GAN) algorithm to produce a part of it. After a simulation, they manage to go from a successful transmission rate of 95.75% to 6.25% using this jamming attack. Another type of adaptive environment is tackled: Cross-Technology Communication (CTC) in [157]. This technique makes it possible to respond to the problem of inter-connectivity between the IoT devices of a different protocol. To achieve direct communication among heterogeneous devices, three methods can be used, such as the change of power level, the change of packet length, and the reordering of the packet. In this work, authors implement a reactive jamming system named JamCloack over a CTC protocol. This attack is composed of both a detection phase and a jamming attack phase. The first step observes and detects CTC activities by classifying the traffic, thanks to the K-means algorithm. They demonstrate their new attack in a real testbed and reduce the packet delivery ratio (PDR) by 80.8%. Finally, many countermeasures, in addition to detecting interference these days, react and adapt to the environment accordingly. An intelligent deep reinforcement learning-based anti-jamming method is bypassed in [158]. This process learns the jammer's strategy and obtains the optimal anti-jamming policy with little information about the attacker. An *et al.* circumnavigate this countermeasure thanks to the Q-learning algorithm taking into account three types of different rewards: a) the NACK when the communication is public, b) the change of power, and c) the detection of users switching channels [152]. They test their attack into three environments: a) a fixed user, b) a frequency hopping user and c) an anti-jamming user. The anti-jamming also employs a machine learning algorithm: Deep reinforcement Learning [159]. Simulations indicate that choosing the right channel can be managed, thus avoiding anti-jamming.

2) *Countermeasures Against Jamming Attacks*: In this part, we list some methods of countermeasures to combat jamming attacks detailed in [144].

- *Frequency Hopping Spread Spectrum (FHSS)*: This method allows alternating use of several channels distributed in a frequency band [160]. The pseudo-random sequence of channel usage is known in advance by the receiver and the transmitter. A sharing algorithm

determines it. This countermeasure is effective only if the attacker cannot deduce or does not know the pseudo-random characteristics of the algorithm used.

- *Direct-Sequence Spread Spectrum (DSSS)*: combines the signals with a pseudo-random signal of much higher frequency to reduce interference and increase resistance to jamming [160]. The receiver filters the noise to obtain original data. By adding noise, information are concealed, which makes DSSS more efficient than FHSS; indeed, it can be difficult for an attacker to restore the transmitted signal.
- *Hybrid FHSS/DSSS*: associates the advantages of FHSS and DSSS methods [161]. This system avoids interference by alerting several channels and by spreading its bandwidth. This technique is easy to implement and increases remarkably the resistance to jamming.
- *Ultra Wide Band Technology (UWB)*: is a radio modulation technique based on the transmission of very short pulses in a wide frequency band [162]. These short pulses, therefore, considerably reduce the effectiveness of jamming attacks since it becomes more difficult for the attacker to target the signal.
- *Polarization of antenna*: the orientation of the antenna and the radiated energy produced by them define the term polarization of the antenna. Two antennas must be configured on the same polarization to be able to communicate. When a node senses that a jamming attack is in progress, it can decide to change polarization and inform the other members of the network. Thus, the jammer must again find the polarization to be used before attacking again.

In addition to existing countermeasures, methods of detection in face of jamming attacks have been developed. The first is based on statistical approaches. Several statistics measurements that may be employed to detect jamming attacks and a case study for each are presented in [163]. The first metric mentioned is a natural measurement, the signal strength. Indeed a jamming attack affects the signal strength of a device. However, in practice, this method is binding because a node does not easily provide this metric. This second is the packet delivery ratio (PDR). As the signal strength, a jammer impacts the average of this metric. Indeed by corrupting a packet, the acknowledgement used to calculate the PDR will never be received. Thus when an attack takes place, the total average of the PDR drops. The last measure is the Carrier Sensing Time; it is the amount of time spent by a node to wait for a channel to become idle. An attacker can prevent a legitimate node from emitting by permanently occupying the channel. Consequently if a channel busy for a long time, the total average of carrier sensing time increases.

Nevertheless, the use of a statistical method to detect a jamming attack has some drawbacks like creating many false positives. That is why several proactive detection and countermeasures have been developed, such as 'JAM' (A Jammed-Area Mapping Service for Sensor Networks) or 'JAID' (An Algorithm for Data Fusion and Jamming Avoidance on WSNs) [38]. Moreover, to address the problem of adaptability to different environments (e.g., protocol, topology, number

of nodes), several machine learning-based detection methods have emerged in recent years. They utilize different algorithms like deep neural networks or reinforcement learning [158], [164], [165].

3) *Open Challenges on ML-Based Jamming Attacks*: In the near future, new lines of research in machine learning-based jamming attacks may be taken into account such as:

- *Target multiple victims*: Indeed, all the experiments mentioned above are carried out only on a single pair of receiver and transmitter. However, the strategy of a jamming attack can vary if the attacker targets several nodes.
- *Increase learning speed of algorithms*: The learning speed of the main solutions required many interactions and time to find the optimal attack strategy. During this time, the attacker is detectable and it is a crucial point from an attacker perspective to reduce the learning time.
- *Compare the consumption of battery*: One of the objectives of using ML algorithms in jamming attacks is to quickly find the optimal strategy in order to consume less battery for the attacker. Moreover, one of the primary purposes of a jamming attack is to cause excessive consumption of the victim battery to achieve a denial of sleep. However, no comparative study of battery consumption, either from an attacker or a victim point of view, was carried out in a real context during most of the previous experiments.
- *Adaptive environment*: Since recent research results have demonstrated the high potential of highly adaptive environment, based on cognitive radios and reprogrammable intelligent metasurface (RIM), investigating intelligent jamming attacks in an adaptive environment constitutes a future research direction.

B. Lessons Learned

An important remark that we can make based on these different works, is that reinforcement learning algorithms are often applied in the analysis of the behavior of a network and therefore of the autonomous implementation of an optimal attack strategy. Indeed, having no knowledge of the network and therefore of the data to be processed, it is impossible to use classification algorithms. As we have just seen through the studies of jamming attacks, using such algorithms has many advantages.

- Finding the optimal strategy, i.e., maximizing the impact on the network while minimizing the probability of being detected.
- Develop scalable attacks based on the target's environment that can be adaptive. Indeed, we could think that this type of attack could lead to a framework which would allow, depending on the configuration of the network (protocol used, number of IoT devices, etc.) to choose an optimal attack strategy autonomously.
- Bypass defence systems based on changing network configuration such as hopping channel.
- Even though in literature have been reported only jamming attacks based on reinforcement learning, the

behavioral analysis and the reinforcement learning can be successfully exploited for creating other attacks such as cloning and spoofing attacks.

Nevertheless, smarter attacks can be successfully realized by addressing some specific issues:

- Before the optimal attack tactic is achieved, the number of attacks activities need to be limited since it is more vulnerable to be detected.
- An attacker can enhance its impact by efficiently targeting several victims with a single attack.

VII. SMART ATTACKS BASED ON DATA GENERATION

It is also possible to use ML methods to generate data when creating attacks against IoT networks. This section describes the ML solutions to automatically generate data and its application in false injection attacks. Then we will analyze the different methods of defense against these last as well as the challenges still open.

A. False Data Injection Attack (FDIA)

1) *ML-Based False Data Injection Attack (FDIA)*: The False Data Injection Attack (FDIA) targeting the integrity of the data aims to insert erroneous data into a system to create false events or reports. An attacker must have access to the IoT device to inject malicious data into the networks. This attack by falsifying data can have severe consequences in specific fields, such as health or military surveillance. This was the case, for example, in 2015, when regional Ukrainian electricity distribution companies were victims of this type of attack affecting more than 225,000 customers, including critical infrastructure such as hospitals [167].

Moreover, this kind of attack can have a powerful impact on the IoT network and provokes a denial of service. Indeed the IoT devices have limited data storage, and if this is reached by false data, the node will delete the real incoming data, and all the whole system will be infected. Besides, the processing of additional data for the sensors results in unnecessary energy consumption.

The use of machine learning in the FDIA attack has made them less detectable by better targeting their victim. Numerous studies have been focused on the machine learning based false data injection attack in a smart grid environment. The smart grid is a complete infrastructure that ensures the distribution of electricity between distributors and consumers in an intelligent way by adjusting the flow according to needs. IoT devices are at the heart of the smart grid and play an essential role in processing, sorting, cleaning, analyzing data. In this context, an attacker is placed between the IoT device and the control center. This location allows it to listen to the measurements and distort them.

FDIA attack with the help of a Linear regression algorithm are constructed in [168]. This attack aims to generate suitable attack vectors based on eavesdropping measurements. An attack vector can be seen as the optimal choice of the node and the data to corrupt. With this process, they circumvent the existing defense techniques based on the support vector machine classifier. An improvement of this method is carried

in [169]. Indeed, in the first experimentation, the authors need to obtain all the exact measurements. However, this latter may be false due to a transmission error, for example, in a real environment. So, the assumption of previous article is therefore not realistic. Thereby, the authors try to answer to this problem by modifying the choice of the machine learning algorithm with robust linear regression in [169]. They compare their solution with the previous article and obtain better results against the detection methods.

The choice of the optimal strategy in the false data injection attack can be designed by a Markov decision problem where the attacker attempts to learn the ideal nodes and measurements to falsify data without being detected. A Q-learning algorithm with the nearest memory sequence is implemented in [170]. The advantages of this method are the attacker does not possess any knowledge.

In another research, the attacker has two goals: a) to be undetected by the system defender and b) to optimize the false data injection attack into a smart grid [171]. To achieve these objectives, the authors based this attack on generative adversarial network (GAN). The attacker knows the type of data required to inject into the measurement system and generates it thanks to the GAN algorithm. They simulate their system and show the effectiveness of the presented algorithm.

2) *Countermeasures Against False Injection Attacks*: As discussed previously, this type of attack can have serious consequences in critical infrastructures, which is why there are protection systems against these threats [172], [173].

- *Protection set of basic measurements*: One of the first ways of securing data and protecting it strategic [174]. Indeed, it has been proven that if the key measures are effectively protected (which constitutes only a small subset of all measures), systems can be effectively immunised.
- *PMU-Based protection*: Another effective method to protect data in a smart grid is to use Phasor measurements units (PMU). PMU is a measurement device equipped with the global positioning system (GPS) technology. In this way, these devices measure in a smart grid the bus voltage phasor directly with the time stamp of the positioning system global. It therefore becomes difficult for an attacker to change or produce the content of the data.
- *Blockchain*: A new method consists of using blockchain to protect the integrity of data. Indeed, this solution is employed to safeguard the healthcare images from false data injection attacks in [175].

3) *Open Challenges on ML-False Injection Attacks*: Considering the considerable damage that these attacks can do, we think that it may be interesting to continue working on them in order to better recognize the possible vulnerabilities; as we saw in the [167]. Several lines of research are possible such as:

- *Non linear data*: For most of the models taken into account, data are assumed to be linear, which does not correspond to a typical scenario of everyday life [168], [169]. Indeed, most of the time, the attacker will not be able to gain complete knowledge of the system to be attacked.

- *Datasets*: This kind of attack are based on data but few dataset exist. It could be interesting for scientific research to have available dataset to create false data injection attack or security system.

B. Lessons Learned

This section shows us that it is possible to generate data with ML algorithms and use them to create an attack. These are essentially unsupervised algorithms like the Generative Adversarial Network which are employed in this case. These algorithms make it possible to create very realistic data and, as shown in the work above, undetectable for certain defense systems or detection methods. However, there are still some challenges in creating an attack that uses ML to generate data that has not been taken into account as:

- Problems linked to the data: indeed in most of the solutions implemented, the attacker works with linear data that he has previously chosen and sorted. Nevertheless, in a real environment, this data is predominantly non-linear and its upstream processing also increases error rates. It is therefore important to continue to develop this type of attack by taking into account the constraints of the data generated in real environments.
- Also in this case, in order to make the attacks more harmful and to create more effective and impact attacks, it is important to reduce the cost and the complexity.

VIII. SMART ATTACKS BASED ON BEHAVIORAL DIVERSION

The last motivation to use ML in attack creation may be to derive the basic behavior of a system that is part of an IoT system. A new kind of attack although not specific to the IoT network can be classified in this category; these are adversarial attacks. We will detail it below along with its countermeasures and open research topics.

A. Adversarial Attacks

1) *ML-Based Adversarial Attacks*: The integration of machine learning into wireless networks has led to new threats. Indeed, many tasks performed by machine learning like spectrum sensing, signal detection, channel estimation have been integrated in new protocols or detection methods. The adversarial attack aims to corrupt a machine learning algorithm by adding small but intentionally selected perturbations to the original inputs [176]. We can classify this kind of attack into three categories according to the level of knowledge of the adversaries: white-box, gray-box, and black-box attacks. In the first approach, the aggressor is aware of the model and parameters used by machine learning approach. In the grey-box pattern, only the model of the algorithm is known. The black box model is the model that most closely resembles real life. Here, we are assuming that the attacker cannot gain access to crucial information about the machine learning algorithm is using and the parameters associated with it. He can guess only the output of the model.

We can also categorize this type of attack in three classes according to their strategies [177].

TABLE X
FALSE INJECTION DATA ATTACKS WITH ML

Ref	Date	ML Algorithm	Test Environment	Remarks
[171]	2018	Linear Regression	Simulation	Bypass the detection methods based on SVM
[172]	2019	RLR	Simulation	Improve the works of [171]
[173]	2018	Q-learning	Real testbed	The attacker needs little knowledge
[174]	2018	GAN	Simulation	Generate false data with GAN algorithm

TABLE XI
ADVERSARIAL ATTACKS WITH ML

Ref	Date	Target	ML Algorithms	Remarks
[186], [187]	2018	Cognitive radios	DNN	Causative attack by falsifying the data
[188]	2019	Cognitive radios	DNN	Causative attack during the retraining phase
[189]	2019	Cognitive radios	NB, SVM, PA-I, PA-II, MLP	Deceive a fusion center and cause a perturbation ratio of 45% to 80%
[190]	2020	Cognitive radios	DNN	Black-box model, create a universal adversarial attack
[191]	2018	NIDS	DNN, ZOO, GAN	Reduce the accuracy of the NIDS even if no information is known
[192]	2019	NIDS	GAN	Evade NIDS while ensuring that the functional behavior of the network traffic is preserved
[193]	2019	NIDS	GAN	DOS attack recognition result of 2% when the NIDS is under attack
[194]	2018	NIDS	SMOTE, DNN	Create a new data generating method named A-SMOKE
[195]	2019	NIDS	Q-learning	Deceive a NIDS with a reinforcement learning algorithm for the first time

- *Exploratory attack*: also called inference attack. It aims to guess the behavior of the machine learning the algorithm used, for example.
- *Evasion attack*: Here the attacker tries to mislead the machine learning algorithm to reach a wrong decision.
- *Causative attacks (or poisoning attacks)*: The purpose is to give false information to a machine learning algorithm. Two strategies can lead to the poisoning attack either by modifying the features and/or the labels of initial training data, or by injecting adversarial samples during the retraining step.

Szegedy *et al.*, Goodfellow *et al.*, and Papernot *et al.* have been exploring this type of attack and thus, methods to create adversarial attacks have been created [178]–[180]. We can find there, the fast gradient sign methods like FGSM, the jacobian-based saliency map attack (JSMA), the DeepFool and the C&W attack [181]. However, these methods are often applied in white-box models, which are not applicable during real scenarios where the attacker is in a black-box context. Therefore, attackers must use machine learning to mimic the behavior of the target or to generate a large number of false data to manipulate a classifier, for example, in a black-box model.

This type of attack can have far-reaching consequences, such as leading to many of the other assaults described above (e.g., jamming attack). Indeed by disrupting the spectrum sensing or channel classification in cognitive radio, the attacker may cause collision, for example. This kind of attack can produce serious damage such as the complete shutdown of the entire network. In addition to damaging a network, this type

of attack can bring down its security system. Indeed a lot of Network Intrusion Detection Systems (NIDS) are based on a machine learning algorithm. An attacker by falsifying data training of NIDS can bypass the security system and launch other attacks.

In the context of an IoT network, we can identify two potential victims to run this attack: the cognitive radio and the Network Intrusion Detection System (NIDS). We have classified the papers dealing with this topic according to these two targets. All articles listed below are categorized in Table XI.

- *Adversarial attacks on cognitive radios*: With the increase of IoT devices in recent years, many problems have arisen, such as bandwidth sharing. Moreover, a cognitive radio (CR) system is a system able to automatically adapt specific parameters according to the environment. This mechanism can be implemented into a receiver or transmitter device and meets certain needs like channel allocation, energy harvesting, and resource management. That is why the integration of cognitive radio networks into IoT is an open subject. Three steps make up the functioning of cognitive radio, the first consists in the sensing the environment, the second aims to make a decision thanks to the data obtained during the first phase and the third is the realization of the chosen action. More and more machine learning algorithms are used in the second phase and therefore become vulnerable to adversarial attacks [57]. Several works attempt to deceive a cognitive radio with a data poisoning attack named ‘over-the-air spectrum data poisoning attack’ [182], [183]. Here a deep learning algorithm is implemented in a transmitter

in order to predict the occupation (idle or busy) of a channel based on spectrum sensing data. The attacker uses the same type of algorithm to predict the success of the transmission based on the Acknowledgment data (ACK/NACK). This process resembles the one installed in [155], [156], but instead of directly jamming the transmission, here the attacker interferes with the channel in order to make it occupied and to mislead the transmitter. This type of attack is more complicated to detect because it does not intervene directly on transmission data but on the output of the classification algorithm. In normal times, cognitive radio infers that the channel is idle 98.96% of the time. Under this attack this deduction is reduced to 3.13%, it is the same for the success ratio transmission which drops from 96.94% to 75%. A causative attack called 'priority violation attack' are also experimented, but during the retraining phase of the machine learning algorithm in [184]. Cognitive radio is used by the transmitter (T) to predict the existence of a high-priority user. When the device deduces a high-priority transmission, it waits during several time slots. An attacker (A) tries to predict the success of communications of T thanks to a deep learning algorithm like as described in [182], [183]. If the transmission of T will be a success, A interferes with the channel and tries to behave like a high-priority user. Therefore, when T retrains its classifier, A provides wrong features in the retraining process. The normalized throughput is reduced from 79.62% to 74.23% and the success probability from 99.05% to 85.78%. This type of attack is not very effective compared to the previous one which uses almost the same process. Another works are also based on cooperative sensing spectrum, though, the decision is taken by the fusion center in [185]. Indeed the nodes report their sensing results to a fusion center that makes a centralized decision on the availability of channels. Luo *et al.* create a new attack nicknamed Learn-Evaluate-Beat (LEB) framework which aims to deceive the decision of fusion center and composed in three steps. The first permits to construct the own substitute model of the fusion center, six types of machine learning algorithm are tested like Naive Bayes classifier, Perceptron classifier, support vector machine (SVM), Passive Aggressive-I classifier (PA-I), Passive Aggressive-II classifier (PA-II) and Multi-layer Perceptron classifier (MLP). In the second phase, the attacker evaluates the accuracy of its system classifier to know if it can launch the attack. The third step is the falsification of the detection data with a minimum cost to change the decision of the fusion center. This framework causes a perturbation ratio of 45% to 80% and causes severe consequences like a denial of service. In addition to the choice of channel, the selection of modulation can be made thanks to cognitive radio [186]. Indeed, in this paper, the authors attack a deep learning-based modulation classifier with an adversarial attack. The system model is composed of a transmitter that chooses a modulation type to emit and a receiver that classifies the modulation of the signal. The authors demonstrate that it is possible to attack a

cognitive radio without knowing the architecture of the machine learning algorithm used. They place themselves in a black-box model and create a universal adversarial attack by basing on the transferability property. This latter explains that an adversarial attack designed to fool a DNN has a high probability of also working on another DNN. They create, therefore, a substitute deep learning algorithm and use to deceive the receive.

- *Adversarial attacks on NIDS*: Network intrusion detection systems have become a crucial component of network security. The traditional methods generally used in these NIDS were statistical approaches with basic rules. However, with the increase in network traffic, the statistical NIDS quickly became outdated. This is why many ML algorithms-based NIDS solving classification problems such as K-Nearest Neighbor, Support Vector Machine, or Decision Tree have been proposed in the literature. Classification algorithms have been utilized to monitor and analyze the traffic. Yang *et al.* form a new Adversarial Attacks against Deep neural network model applied in a NIDS in [187]. The authors are in the context of the black box and try to show how the addition of small disturbances in the original input can lead the model to an incorrect classification. They evaluate three different adversarial algorithms and test their performance. The aim is to generate false input, which varies slightly from the real ones in order to disrupt the classification algorithm in the NIDS. The first model is the attack based on a substitute model where the authors use the C&W algorithm to generate adversary examples target on the substitute model. The second is based on the zeroth-order optimization (ZOO) algorithm and the last on the generative adversarial network (GAN). Attacks based on ZOO and GAN have the best impact in the NIDS, and considerably decreases the accuracy of the latter. So deceive a NIDS is possible even if the attacker does not possess internal information. Several explorations also aim to deceive and evade the classifier by generating compromised data [188], [189]. In order to create data with minimal modification, they use generative adversarial network algorithms (GAN) based on the original malicious traffic. They test their adversary attacks against many classification algorithms, such as support vector machine or decision tree used in NIDS. The authors arrive at a DoS attack recognition result of 2% when the NIDS is under attack against 80% in normal time in [189]. Li *et al.* propose a poisoning strategy by stealing the learning model, which can threaten the security and the availability of NIDS [190]. First, they create a new data generating method named A-SMOKE based on Synthetic Minority Oversampling Technique (SMOKE). Then they imitate the targeted model by training substitute model using deep neural network and the augmented training data generated during the first step. Finally, the authors establish a poisoning method CBPC and combine this with the substitute model to establish a new attack. The aim is to modify the training data of the NIDS in order to evade detection. They prove the effectiveness of

this method with three real data sets and different algorithms used by the NIDS. Q-learning algorithm can also be used to create an adversarial attack on NIDS. Indeed it is possible to deceive a Botnet Attack detection model by generating adversarial traffic flows and adding perturbations in training data with reinforcement learning as proved in [191]. The attacker chooses from a set of five actions a strategy to modify the data flow and receive a positive reward by the NIDS if the new botnet flow was not detected. They attempt their algorithm in two kinds of botnet detection models; one is based on a decision tree algorithm and the second on a deep learning model. They obtain an evasion rate of 40% for the decision deep learning model. So they prove when using a reinforcement learning algorithm, it is possible to fool a NIDS.

2) *Countermeasures*: This type of attack is relatively recent, but many defenses have already emerged. An exhaustive list is given in [192]. We will briefly explain here the most used methods.

- *Adversarial training gradient*: The objective is to voluntarily inject adversarial examples into the training set to increase the robustness of the model. However, this defense technique is not efficient against a black-box attack.
- *Blocking the transferability*: This method aims to counteract the property of transferability and prevent an attacker from using a substitute classifier to generate an adversarial example. To this end, the input of the training set is more perturbed by adding a label class 'null'. Therefore, the confidence on the original label is lower, and the classifier rejects the adversarial examples by classifying them as NULL.
- *Defense-GAN*: This mechanism works in black-box and white-box attacks and defends deep neural networks against the perturbations. The goal is to 'denoise' the adversarial examples with the help of a generative adversarial network.

3) *Open Challenges on ML-Based Adversarial Attacks*: This kind of attack in full expansion can have several axes of improvement like:

- *Studies based on the optimal adversarial sample*: Few studies on the generation of optimal samples have been carried out. What are the good characteristics that make a adversarial sample able to deceive a cognitive radio or a NIDS?
- *Generative method stability*: Whether using adversarial generative network algorithms or reinforcement learning algorithms, the generation of adversarial samples remains long and unstable. Indeed the GAN algorithm has disadvantages to require an appropriate synchronization between the generator and the discriminator, but it is challenging to find a correct balance. Moreover, the rewards in reinforcement learning and the set of actions are defined by the attacker; therefore, inadequate knowledge of the environment can lead it to define wrong action or reward. So the stability of these algorithms and of the generated data are still based on susceptible elements.

B. Lessons Learned

In the previous sections we have demonstrated that it is possible to exploit ML algorithms for generating smart attacks by deriving the basic behavior of an IoT system. This can be the operation of the communication protocol, the defense system or even the IoT device for example. In this section, we considered a new type of attack: adversarial attacks which have seen their use increase over time with those of ML algorithms in IoT networks. Indeed, the integration of ML in IoT systems enable the possibility to resolve many challenges but also opened the door to various attack surfaces. In this section, two main targets on IoT networks are selected by this type of attack: cognitive radios where the goal is to alter the decision of the algorithm to reduce the performance of the system or the NIDS in order to decrease the probability of attack detection. This type of attack can cause real consequences and quickly decreases the performance of IoT networks. In addition, being at an early stage this type of attacks, there are not effective countermeasures compared to other attacks. It is therefore essential to take them into account when a new IoT network is deployed.

Using machine learning algorithms to deviate the basic behavior of a system most often appeals to two other motivations that we saw above: the deduction of a behavior of the system and the generation of false data. This is why, in the works cited above, it is essentially about algorithms that we find in the last two sections: reinforcement learning or unsupervised methods.

IX. DISCUSSION

In this section we summarize the key factors of smart attacks, their current status and we provide an overview on new research directions this type of attacks undertakes.

A. Lessons Learned

Through the various works mentioned just above, we have seen that the landscape of cyberattacks has expanded significantly in recent years. In fact, machine learning algorithms integrated into cyberattacks have enabled many advances.

- First of all, threats that were previously unachievable due to their complexity of implementation or lack of resources have emerged thanks to this technological advance.
- In addition, it has made existing attacks more effective and more robust against existing detection methods and countermeasures.
- We can also retain that it is now plausible for a potential attacker to more easily design more targeted and more cognitive assaults for each victim. Indeed, it would be possible to have attacks configured to disrupt the network only at specific times to maximize its effectiveness and simultaneously minimize its probability of being detected.
- On the contrary, it is also possible to think that this innovative approach could generate more generalized and automated attacks which would adapt more easily to changes in network behavior or to different protocols and types of targets. Indeed, it would be easy to think

that an attacker could launch an attack on various types of network protocols or IoT devices while not having to modify the code of the attack each time. After a brief learning phase, the attack could self-configure depending on the parameters it observes.

- Finally, the use of machine learning within attack creation can be justified by several essential reasons such as the analysis or generation of data, the deduction or the deceive the behavior of an IoT network.

Moreover, we believe that this type of attack will grow rapidly in the coming years. It is therefore essential for IoT devices manufacturers to take these threats into account as well as for those implementing new communication protocols and security systems. The reasons of this rapid expansion can be:

- The simplicity of accessing machine learning algorithms. Indeed, this implementation requires less and less knowledge due to the open-source machine learning frameworks which simplifies the creation of algorithms. Today, a person can implement machine learning algorithms without having great skills in the matter.
- Machines on the market are more and more powerful and are now sufficiently robust to support such an algorithm.
- Easy access to data or programs from the Internet. Indeed, storing data or having access to a machine learning program anywhere in the world is less and less complicated thanks to cloud computing or serverless platform. Attackers can therefore access a set of essential data for the proper functioning of their algorithm effortlessly and quickly nowadays.

Finally, we can note that this evolution in the cyberattack landscape can be seen as the response of protection systems that are becoming increasingly efficient, autonomous and robust.

B. Open Challenges and Future Directions

Although many studies confirm the progress and the advantages of smart attacks, there are still many challenges to be met. First of all, we can cite the challenges linked to machine learning algorithms.

- It is essential to improve learning methods in order to reduce the cost of the necessary resources and their training time while increasing their performance.
- The supervised algorithms used to create smart attacks today still require a lot of training data. However, during the listening and information collection phase, an attacker is likely to be detected. Thus, limiting the amount of required training data would allow to reduce the exposure time of an attacker.
- Among the numerous emerging challenges is the creation of accessible datasets for research teams on this subject. Indeed, collecting, sorting data can prove to be tedious and long. However, it is the basis of many works integrating machine learning methods, whether from an offensive or defensive point of view. Although there exist some data sources [193], [194], it would be good to diversify these

sources as they do not cover all areas of application of IoT systems or different attacks.

- From a defense perspective, it would also make sense to improve the robustness of machine learning algorithms in the face of adversarial attack.

There are also challenges related to the attacks themselves:

- Of the 52 works integrating machine learning algorithms into attack creation, only about 15 projects have been tested in real conditions and only less than a quarter on a large-scale network. It would therefore be interesting to test these different attacks in real applications to be able to study the different impacts that it would have (e.g., number of nodes impacted, resistance of communication protocols).
- These main works focus most of the time on the effectiveness of the attack which has been configured in a specific way. However, it could be interesting to evaluate other parameters such as energy consumption from an attacker's point of view or the optimal distance from its victim to carry out an attack.
- Using ML methods within attacks make it possible to respond to very specific motivations such as the analysis and deduction of the behavior of the IoT network in order to find the optimal strategy autonomously. Nevertheless, we have found that only one attack responds to this motivation in the literature: jamming attacks. We think it might be interesting to investigate other types of attacks such as cloning or identity theft in order to find an ideal tactic. Another challenge is therefore the implementation of other kinds of intelligent attacks in order to find and evaluate potential flaws related to the communications protocol and the deduction system.
- From a protection point of view, testing the various defensive methods recently implemented against these new types of attacks might also reveal some of their limitations.

Even though there are still open issues, it is important to make the various players in IoT networks aware of the even increasing smart attack and the damage they can cause.

X. CONCLUSION

The paper describes the literature review of ML methods used for creating attacks on IoT networks. We can notice that this evolution can be seen as the answer of the effective detection systems or countermeasures that are implemented in the IoT protocols in order to make them more robust in face of cyberattacks. The approach of attacks integrating ML algorithms allowed the improvement of already existing attacks, the creation of some attacks that were infeasible or the opening of new attack vectors as we have seen in Section IV. However, these smart attacks can still be improved as mentioned for each type in Sections V, VI, VII, and VIII. Indeed, for most of them the learning time can be reduced which would at the same time minimize their probability of being detected.

Moreover, the lack of datasets that allow this new type of attack to be produced may cause the scientific community to delay their development. However, this work makes it possible

to discover new vulnerabilities and therefore to anticipate the next attacks and improve the counter systems.

In conclusion, we believe that this survey can provide useful insights into cyberattack creation with machine learning methods and help readers interested in developing new solutions to prevent evolved and more powerful attacks in IoT networks.

REFERENCES

- [1] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *Proc. 10th Int. Conf. Cyber Conflict (CyCon)*, 2018, pp. 371–390.
- [2] "Do Criminals Dream of Electric Sheep?" Jul. 2019. [Online]. Available: https://www.europol.europa.eu/sites/default/files/documents/report_dMarionOswaldo_criminals_dream_of_electric_sheep.pdf (Accessed: Jul. 20, 2020).
- [3] C. R. A. Babuta and M. Oswald, "Machine Learning Algorithms and Police Decision-Making Legal, Ethical and Regulatory Challenges," Sep. 2018. [Online]. Available: https://rusi.org/sites/default/files/201809_whr_3-18_machine_learning_algorithms.pdf.pdf (Accessed: Jul. 20, 2020).
- [4] "Machine Learning Platform: Tensorflow." [Online]. Available: <https://www.tensorflow.org/?hl=fr> (Accessed: Jul. 20, 2020).
- [5] Y. Arfat, S. Usman, R. Mehmood, and I. Katib, "Big data tools, technologies, and applications: A survey," in *Smart Infrastructure and Applications*. Cham, Switzerland: Springer, 2020, pp. 453–490.
- [6] "Cisco Edge to Enterprise IoT Analytics for Electric Utilities Solution Overview." Cisco. 2018. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/big-data/solution-overview-c22-740248.html> (Accessed: May 11, 2020).
- [7] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-Things-based smart environments: State of the art, taxonomy, and open research challenges," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 10–16, Oct. 2016.
- [8] I. Mat, M. R. M. Kassim, A. N. Harun, and I. M. Yusoff, "Smart agriculture using Internet of Things," in *Proc. IEEE Conf. Open Syst. (ICOS)*, 2018, pp. 54–59.
- [9] M. Zhang, T. Yu, and G. F. Zhai, "Smart transport system based on 'the Internet of Things,'" in *Applied Mechanics and Materials*, vol. 48. Dürnten, Switzerland: Trans Tech Publ., 2011, pp. 1073–1076.
- [10] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
- [11] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
- [12] V. Gotarane and S. Raskar, "IoT practices in military applications," in *Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, 2019, pp. 891–894.
- [13] F. Al-Turjman and M. Abujubbeh, "IoT-enabled smart grid via SM: An overview," *Future Gener. Comput. Syst.*, vol. 96, pp. 579–590, Jul. 2019.
- [14] M. Antonakakis et al., "Understanding the Mirai botnet," in *Proc. 26th USENIX Security Symposium (USENIX Security)*, 2017, pp. 1093–1110.
- [15] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," in *Proc. 70th Annu. Conf. Protect. Relay Eng. (CPRE)*, 2017, pp. 1–8.
- [16] B. Alexander, S. Haseeb, and A. Baranchuk, "Are implanted electronic devices hackable?" *Trends Cardiovasc. Med.*, vol. 29, no. 8, pp. 476–480, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1050173818302597>
- [17] "Muddy Waters LLC MW is Short St. Jude Medical (STJ:US)." 2016. [Online]. Available: <http://d.muddywatersresearch.com/research/stj/mw-is-short-stj/> (Accessed: Jan. 15, 2021).
- [18] M. Dibaei et al., "Attacks and defences on intelligent connected vehicles: A survey," *Digit. Commun. Netw.*, vol. 6, no. 4, pp. 399–421, 2020.
- [19] J. Seymour and P. Tully, "Weaponizing data science for social engineering: Automated E2E spear phishing on twitter," in *Proc. Black Hat USA*, vol. 37, 2016, pp. 1–39.
- [20] D. Kirat, J. Jang, and M. Stoecklin, "Deeplocker—Concealing targeted attacks with AI locksmithing," in *Proc. Blackhat USA*, 2018. [Online]. Available: <https://www.blackhat.com/us-18/briefings/schedule/#deeplocker%E2%80%94concealing-targeted-attacks-with-ai-locksmithing-11549> (Assessed: Nov. 20, 2021).
- [21] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.
- [22] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 4, pp. 586–602, Oct.–Dec. 2017.
- [23] S. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *J. Netw. Comput. Appl.*, vol. 64, Jul. 2020, Art. no. 102630.
- [24] T. C. King, N. Aggarwal, M. Taddeo, and L. Floridi, "Artificial intelligence crime: An interdisciplinary analysis of foreseeable threats and solutions," *Sci. Eng. Ethics*, vol. 26, no. 1, pp. 89–120, 2020.
- [25] M. Caldwell, J. Andrews, T. Tanay, and L. Griffin, "AI-enabled future crime," *Crime Sci.*, vol. 9, no. 1, pp. 1–13, 2020.
- [26] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.
- [27] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IoT) technologies, applications and challenges," in *Proc. IEEE Smart Energy Grid Eng. (SEGE)*, 2016, pp. 381–385.
- [28] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
- [29] K. K. Patel and S. M. Patel, "Internet of Things-IOT: Definition, characteristics, architecture, enabling technologies, application & future challenges," *Int. J. Eng. Sci. Comput.*, vol. 6, no. 5, pp. 6122–6131, May 2016.
- [30] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 32–37.
- [31] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.
- [32] O. Yousuf and R. N. Mir, "A survey on the Internet of Things security: State-of-art, architecture, issues and countermeasures," *Inf. Comput. Security*, vol. 27, no. 2, pp. 292–323, 2019.
- [33] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019.
- [34] E. Bou-Harb and N. Neshenko, *Cyber Threat Intelligence for the Internet of Things*. Cham, Switzerland: Springer, Feb. 2020.
- [35] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2015, pp. 180–187.
- [36] F. Alkhudhayr, S. Alfarraj, B. Aljameeli, and S. Elkhidri, "Information security: A review of information security issues and techniques," in *Proc. 2nd Int. Conf. Comput. Appl. Inf. Security (ICCAIS)*, 2019, pp. 1–6.
- [37] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.
- [38] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 4, pp. 42–56, 4th Quart., 2009.
- [39] J.-F. Raymond, *Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems*. Heidelberg, Germany: Springer, 2001, pp. 10–29. [Online]. Available: https://doi.org/10.1007/3-540-44702-4_2
- [40] G. Li, Z. Yan, and Y. Fu, "A study and simulation research of blackhole attack on mobile adhoc network," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2018, pp. 1–6.
- [41] M. Kaur and A. Singh, "Detection and mitigation of sinkhole attack in wireless sensor network," in *Proc. Int. Conf. Micro-Electron. Telecommun. Eng. (ICMETE)*, 2016, pp. 217–221.
- [42] G. J. Persial, M. Prabhu, and R. Shanmugalakshmi, "Side channel attack-survey," *Int. J. Adv. Sci. Res. Rev.*, vol. 1, no. 4, pp. 54–57, 2011.
- [43] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020.
- [44] M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: A survey," *J. Supercomput.*, vol. 76, pp. 5320–5363, Jul. 2019.

- [45] A. Jain and S. Jain, "A survey on miscellaneous attacks and countermeasures for RPL routing protocol in IoT," in *Proc. IEMIS*, vol. 3, 2019, pp. 611–620.
- [46] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A review," in *Proc. Int. Conf. Comput. Sci. Electron. Eng.*, vol. 3, 2012, pp. 648–651.
- [47] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–29, 2019.
- [48] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, Feb. 2017.
- [49] I. U. Din, M. Guizani, B.-S. Kim, S. Hassan, and M. K. Khan, "Trust management techniques for the Internet of Things: A survey," *IEEE Access*, vol. 7, pp. 29763–29787, 2018.
- [50] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 812–837, 1st Quart., 2018.
- [51] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021.
- [52] I. K. Dutta, B. Ghosh, and M. Bayoumi, "Lightweight cryptography for Internet of Insecure Things: A survey," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2019, pp. 475–481.
- [53] A. Shah and M. Engineer, "A survey of lightweight cryptographic algorithms for IoT-based applications," in *Smart Innovations in Communication and Computational Sciences*. Singapore: Springer, 2019, pp. 283–293.
- [54] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
- [55] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.
- [56] N. Abbas, Y. Nasser, and K. El Ahmad, "Recent advances on artificial intelligence and learning techniques in cognitive radio networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–20, 2015.
- [57] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, 3rd Quart., 2012.
- [58] A. Khan, M. H. Rehmani, and A. Rachedi, "Cognitive-radio-based Internet of Things: Applications, architectures, spectrum related functionalities, and future research directions," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 17–25, Jun. 2017.
- [59] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
- [60] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, Nov. 2017.
- [61] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.
- [62] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.
- [63] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.
- [64] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.
- [65] M. S. Mahdavinjad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 161–175, 2018.
- [66] R. P. Lippmann *et al.*, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," in *Proc. DARPA Inf. Survivability Conf. Expo.*, vol. 2, 2000, pp. 12–26.
- [67] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sep. 2018.
- [68] S. Zeadally and M. Tsikerdakis, "Securing Internet of Things (IoT) with machine learning," *Int. J. Commun. Syst.*, vol. 33, no. 1, 2020, Art. no. e4169.
- [69] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.
- [70] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," *Inf. Security Tech. Rep.*, vol. 14, no. 1, pp. 16–29, 2009.
- [71] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Comput. Security*, vol. 81, pp. 123–147, Mar. 2019.
- [72] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2018.
- [73] L. Wang and R. Jones, "Big data analytics for network intrusion detection: A survey," *Int. J. Netw. Commun.*, vol. 7, no. 1, pp. 24–31, 2017.
- [74] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.
- [75] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3496–3509, 4th Quart., 2018.
- [76] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.
- [77] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.
- [78] L. Pajola, L. Pasa, and M. Conti, "Threat is in the air: Machine learning for wireless network applications," in *Proc. ACM Workshop Wireless Security Mach. Learn.*, 2019, pp. 16–21. [Online]. Available: <https://doi.org/10.1145/3324921.3328783>
- [79] P. Sen, M. Hajra, and M. Ghosh, *Supervised Classification Algorithms in Machine Learning: A Survey and Review*. Singapore: Springer, 2020, pp. 99–111.
- [80] N. Bhatia and V. Bachani, "Survey of nearest neighbor techniques," 2010, *arXiv:1007.0085*.
- [81] K. Chomboon, P. Chujai, P. Teerarassamsee, K. Kerdprasop, and N. Kerdprasop, "An empirical study of distance metrics for k-nearest neighbor algorithm," in *Proc. Int. Conf. Ind. Appl. Eng.*, 2015, pp. 280–285.
- [82] P. Deshmukh and D. Kumar, "Decision tree classifier: A detailed survey," *Int. J. Inf. Decis. Sci.*, vol. 12, no. 3, pp. 246–269, 2020.
- [83] S. Singh and P. Gupta, "Comparative study ID3, CART and C4.5 decision tree algorithm: A survey," *Int. J. Adv. Inf. Sci. Technol.*, vol. 27, no. 27, pp. 97–103, 2014.
- [84] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.
- [85] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.
- [86] C. Cortes and V. Vapnik, "Support vector machine," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [87] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. Hoboken, NJ, USA: Wiley, 2012, vol. 821.
- [88] M. E. Celebi and K. Aydin, *Unsupervised Learning Algorithms*. Cham, Switzerland: Springer, 2016.
- [89] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, 2003.
- [90] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
- [91] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2014, pp. 2672–2680.
- [92] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
- [93] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," 2014, *arXiv:1402.6028*.

- [94] D. Chapman and L. P. Kaelbling, "Input generalization in delayed reinforcement learning: An algorithm and performance comparisons," in *Proc. IJCAI*, vol. 91, 1991, pp. 726–731.
- [95] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
- [96] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, nos. 5–6, pp. 183–197, 1991.
- [97] Y. Reich and S. Barai, "Evaluating machine learning models for engineering problems," *Artif. Intell. Eng.*, vol. 13, no. 3, pp. 257–272, 1999.
- [98] T. Fushiki, "Estimation of prediction error by using k-fold cross-validation," *Stat. Comput.*, vol. 21, no. 2, pp. 137–146, 2011.
- [99] "Machine Learning Platform: Openai-Gym." [Online]. Available: <https://gym.openai.com/> (Accessed: Jul. 20, 2020).
- [100] M. Brundage *et al.*, "The malicious use of artificial intelligence: Forecasting prevention and mitigation," 2018, *arXiv:1802.07228*.
- [101] M. Akdere, C. Bilgin, O. Gerdaneri, I. Korpeoglu, Z. Ulusoy, and U. Çetintemel, "A comparison of epidemic algorithms in wireless sensor networks," *Comput. Commun.*, vol. 29, pp. 2450–2457, Aug. 2006.
- [102] A. Tabassum, A. Erbad, and M. Guizani, "A survey on recent approaches in intrusion detection system in IoTs," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2019, pp. 1190–1197.
- [103] D. Stiawan *et al.*, "Investigating brute force attack patterns in IoT network," *J. Electr. Comput. Eng.*, vol. 2019, Apr. 2019, Art. no. 4568368.
- [104] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *Proc. Comput. Security Found. Workshop VII*, 1994, pp. 187–191.
- [105] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the Internet of Things: A comprehensive survey," *Ad Hoc Netw.*, vol. 93, Oct. 2019, Art. no. 101913. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870519300812>
- [106] V. Ishakian, V. Muthusamy, and A. Slominski, "Serving deep learning models in a serverless platform," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, 2018, pp. 257–262.
- [107] V. Toldov, L. Clavier, V. Loscri, and N. Mitton, "A thompson sampling approach to channel exploration-exploitation problem in multihop cognitive radio networks," in *Proc. IEEE 27th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2016, pp. 1–6.
- [108] J. Ziegeldorf, O. Morchon, and K. Wehrle, "Privacy in the Internet of Things: Threats and challenges," *Security Commun. Netw.*, vol. 7, no. 12, pp. 2728–2742, 2014.
- [109] S. Wong, "The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards," Bethesda, MD, USA, SANS Inst., White Paper, 2003.
- [110] B. Cocos, K. Levitt, M. Bishop, and J. Rowe, "Is anybody home? Inferring activity from smart home network traffic," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2016, pp. 245–251.
- [111] A. Sivanathan *et al.*, "Characterizing and classifying IoT traffic in smart cities and campuses," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 559–564.
- [112] M. Skowron, A. Janicki, and W. Mazurczyk, "Traffic fingerprinting attacks on Internet of Things using machine learning," *IEEE Access*, vol. 8, pp. 20386–20400, 2020.
- [113] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic," Aug. 2017, *arXiv:1708.05044*.
- [114] N. Msadek, R. Soua, and T. Engel, "IoT device fingerprinting: Machine learning based encrypted traffic analysis," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–8.
- [115] A. Acar *et al.*, "Peek-a-boo: I see your smart home activities, even encrypted!" in *Proc. 13th ACM Conf. Security Privacy Wireless Mobile Netw.*, Jul. 2020, pp. 207–218. [Online]. Available: <http://dx.doi.org/10.1145/3395351.3399421>
- [116] S. Dong, Z. Li, D. Tang, J. Chen, M. Sun, and K. Zhang, "Your smart home can't keep a secret: Towards automated fingerprinting of IoT traffic with neural networks," 2019, *arXiv:1909.00104*.
- [117] J. Ren, D. Dubois, D. Choffnes, A. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach," in *Proc. IMC*, Oct. 2019, pp. 267–279.
- [118] R. Trimananda, J. Varmarken, A. Markopoulou, and B. Demsky, "PingPong: Packet-level signatures for smart home device events," 2020, *arXiv:1907.11797*.
- [119] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," in *Proc. Int. Conf. Electron. Commun. Comput. Eng. (ICECCE)*, 2014, pp. 83–93.
- [120] E. Ronen, A. Shamir, A. Weingarten, and C. O'Flynn, "IoT goes nuclear: Creating a ZigBee chain reaction," *IEEE Security Privacy*, vol. 16, no. 1, pp. 54–62, Jan./Feb. 2018.
- [121] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems (CHES)*, B. S. Kaliski, ç. K. Koç, and C. Paar, Eds. Heidelberg, Germany: Springer, 2003, pp. 13–28.
- [122] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.*, 1999, pp. 388–397.
- [123] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptogr. Hardw. Embedded Syst.*, 2004, pp. 16–29.
- [124] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *Proc. Int. Workshop Cryptogr. Hardw. Embedded Syst.*, 2008, pp. 426–442.
- [125] H. Patel and R. Baldwin, "Random forest profiling attack on advanced encryption standard," *Int. J. Appl. Cryptogr.*, vol. 3, no. 2, pp. 181–194, 2014.
- [126] L. Lerman, G. Bontempi, and O. Markowitch, "Power analysis attack: An approach based on machine learning," *Int. J. Appl. Cryptogr.*, vol. 3, no. 2, pp. 97–115, 2014.
- [127] H. Tsague and B. Twala, *Practical Techniques for Securing the Internet of Things (IoT) Against Side Channel Attacks*. Cham, Switzerland: Springer, 2018, pp. 439–481.
- [128] L. Lerman, R. Poussier, G. Bontempi, O. Markowitch, and F.-X. Standaert, "Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis)," in *Proc. Int. Workshop Construct. Side-Channel Anal. Secure Design*, 2015, pp. 20–33.
- [129] S. Picek *et al.*, "Side-channel analysis and machine learning: A practical perspective," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2017, pp. 4095–4102.
- [130] H. Maghrebi, T. Portigliatti, and E. Prouff, *Breaking Cryptographic Implementations Using Deep Learning Techniques*. Cham, Switzerland: Springer, Dec. 2016, pp. 3–26.
- [131] Z. Martinasek, P. Dzurenda, and L. Malina, "Profiling power analysis attack based on MLP in DPA contest V4.2," in *Proc. 39th Int. Conf. Telecommun. Signal Process. (TSP)*, 2016, pp. 223–226.
- [132] Z. Martinasek, O. Zapletal, K. Vrba, and K. Trasy, "Power analysis attack based on the MLP in DPA contest V4," in *Proc. 38th Int. Conf. Telecommun. Signal Process. (TSP)*, 2015, pp. 154–158.
- [133] Z. Martinasek, L. Malina, and K. Trasy, *Profiling Power Analysis Attack Based on Multi-Layer Perceptron Network*. Cham, Switzerland: Springer, vol. 343, 2015, pp. 317–339.
- [134] J.-S. Coron, E. Prouff, and M. Rivain, "Side channel cryptanalysis of a higher order masking scheme," in *Cryptographic Hardware and Embedded Systems (CHES)*, P. Paillier and I. Verbauwhede, Eds. Heidelberg, Germany: Springer, 2007, pp. 28–44.
- [135] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.
- [136] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," in *Proc. Int. Conf. Cryptogr. Hardw. Embedded Syst.*, 2017, pp. 45–68.
- [137] Y. Zotkin, F. Olivier, and E. Bourbao, "Deep learning vs template attacks in front of fundamental targets: Experimental study," IACR, Lyon, France, Rep. 2018/1213, 2018.
- [138] J. Xu, Y. Tang, Y. Wang, and X. Wang, "A practical side-channel attack of a lorawan module using deep learning," in *Proc. IEEE 13th Int. Conf. Anti-Counterfeiting Security Identification (ASID)*, 2019, pp. 17–21.
- [139] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, *Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database*, vol. 22, ANSSI, Paris, France, 2018. [Online]. Available: <https://eprint.iacr.org/2018/053.pdf>
- [140] C. Pfeifer and P. Haddad, "Spread: A new layer for profiled deep-learning side-channel attacks," IACR, Lyon, France, Rep. 2018/880, 2018.
- [141] B. Timon, "Non-profiled deep learning-based side-channel attacks," IACR, Lyon, France, Rep. 2018/196, 2018.
- [142] Z. Chen and Y. Zhou, "Dual-rail random switching logic: A countermeasure to reduce side channel leakage," in *Cryptographic Hardware and Embedded Systems*, L. Goubin and M. Matsui, Eds. Heidelberg, Germany: Springer, 2006, pp. 242–254.
- [143] S. R. Ratna and R. Ravi, "Survey on jamming wireless networks: Attacks and prevention strategies," *Int. J. Comput. Inf. Eng.*, vol. 9, no. 2, pp. 642–648, 2015.
- [144] S. Jaitly, H. Malhotra, and B. Bhushan, "Security vulnerabilities and countermeasures against jamming attacks in wireless sensor networks:

- A survey,” in *Proc. Int. Conf. Comput. Commun. Electron. (Comptelix)*, 2017, pp. 559–564.
- [145] A. Gallais, T.-H. Hedli, V. Loscri, and N. Mitton, “Denial-of-sleep attacks against IoT networks,” in *Proc. 6th Int. Conf. Control Decis. Inf. Technol. (CoDIT)*, 2019, pp. 1025–1030.
- [146] G. Kim, Y. Kim, J. Park, and H. Lim, “Frame-selective wireless attack using deep-learning-based length prediction,” in *Proc. 15th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, 2018, pp. 1–2.
- [147] S. Amuru and R. M. Buehrer, “Optimal jamming using delayed learning,” in *Proc. IEEE Military Commun. Conf.*, 2014, pp. 1528–1533.
- [148] S. Amuru, C. Tekin, M. V. der Schaar, and R. M. Buehrer, “Jamming bandits—A novel learning method for optimal jamming,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2792–2808, Apr. 2016.
- [149] S. Zhuansun, J. Yang, H. Liu, and K. Huang, “A novel jamming strategy-greedy bandit,” in *Proc. IEEE 9th Int. Conf. Commun. Softw. Netw. (ICCSN)*, 2017, pp. 1142–1146.
- [150] N. Thurston, G. Vanhoy, and T. Bose, “Intelligent Jamming Using Deep Q-Learning,” 2018. [Online]. Available: <http://hdl.handle.net/10150/631658>
- [151] S. Zhuansun, J.-A. Yang, and H. Liu, “An algorithm for jamming strategy using OMP and MAB,” *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 85, 2019.
- [152] L. An and G.-H. Yang, “Data-based optimal denial-of-service attack scheduling against robust control based on Q-learning,” *Int. J. Robust Nonlinear Control*, vol. 29, no. 15, pp. 5178–5194, 2019.
- [153] L. Zhang, F. Restuccia, T. Melodia, and S. M. Pudleski, “Jam sessions: Analysis and experimental evaluation of advanced jamming attacks in MIMO networks,” in *Proc. Mobihoc*, Jul. 2019, pp. 61–70.
- [154] S. Zhuansun, J.-A. Yang, and H. Liu, “Apprenticeship learning in cognitive jamming,” *Optimal Control Appl. Methods*, vol. 40, no. 4, pp. 647–658, 2019.
- [155] T. Erpek, Y. E. Sagduyu, and Y. Shi, “Deep learning for launching and mitigating wireless jamming attacks,” 2018, *arXiv:1807.02567*.
- [156] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, “Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2018, pp. 1–6.
- [157] G. Chen and W. Dong, “JamCloak: Reactive jamming attack over cross-technology communication links,” in *Proc. IEEE 26th Int. Conf. Netw. Protocols (ICNP)*, 2018, pp. 34–43.
- [158] Y. Li *et al.*, “On the performance of deep reinforcement learning-based anti-jamming method confronting intelligent jammer,” *Appl. Sci.*, vol. 9, no. 7, p. 1361, 2019.
- [159] Y. Li, “Deep reinforcement learning: An overview,” 2017, *arXiv:1701.07274*.
- [160] R. Pickholtz, D. Schilling, and L. Milstein, “Theory of spread-spectrum communications—A tutorial,” *IEEE Trans. Commun.*, vol. 30, no. 5, pp. 855–884, May 1982.
- [161] M. M. Olama, X. Ma, T. P. Kuruganti, S. F. Smith, and S. M. Djouadi, “Hybrid DS/FFH spread-spectrum: A robust, secure transmission technique for communication in harsh environments,” in *Proc. Military Commun. Conf.*, 2011, pp. 2136–2141.
- [162] Y. Rahayu, T. A. Rahman, R. Ngah, and P. Hall, “Ultra wideband technology and its applications,” in *Proc. 5th IFIP Int. Conf. Wireless Opt. Commun. Netw. (WOCN)*, 2008, pp. 1–5.
- [163] W. Xu, W. Trappe, Y. Zhang, and T. Wood, “The feasibility of launching and detecting jamming attacks in wireless networks,” in *Proc. MobiHoc*, 2005, pp. 46–57. [Online]. Available: <https://doi.org/10.1145/1062689.1062697>
- [164] S. Gegel, C. Goztepe, and G. K. Kurt, “Jammer detection based on artificial neural networks: A measurement study,” in *Proc. ACM Workshop Wireless Security Mach. Learn.*, 2019, pp. 43–48. [Online]. Available: <https://doi.org/10.1145/3324921.3328788>
- [165] M. A. Aref, S. K. Jayaweera, and S. Machuzak, “Multi-agent reinforcement learning based cognitive anti-jamming,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2017, pp. 1–6.
- [166] S. Amuru, C. Tekin, M. van der Schaar, and R. M. Buehrer, “A systematic learning method for optimal jamming,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 2822–2827.
- [167] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, “The 2015 Ukraine Blackout: Implications for false data injection attacks,” *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3317–3318, Jul. 2017.
- [168] R. Nawaz, M. A. Shahid, I. M. Qureshi, and M. H. Mehmood, “Machine learning based false data injection in smart grid,” in *Proc. 1st Int. Conf. Power Energy Smart Grid (ICPESG)*, 2018, pp. 1–6.
- [169] J. Tian, B. Wang, T. Li, F. Shang, K. Cao, and J. Li, “Stealthy and sparse false data injection attacks based on machine learning,” in *Proc. Int. Symp. Cyberspace Safety Security*, 2019, pp. 337–347.
- [170] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, “Evaluation of reinforcement learning-based false data injection attack to automatic voltage control,” *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2158–2169, Mar. 2019.
- [171] S. Ahmadian, H. Malki, and Z. Han, “Cyber attacks on smart energy grids using generative adversarial networks,” in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2018, pp. 942–946.
- [172] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, “A review of false data injection attacks against modern power systems,” *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017.
- [173] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, “False data injection on state estimation in power systems—Attacks, impacts, and defense: A survey,” *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 411–423, Apr. 2017.
- [174] T. T. Kim and H. V. Poor, “Strategic protection against data injection attacks on power grids,” *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 326–333, Jun. 2011.
- [175] M. Ahmed, “False image injection prevention using iChain,” *Appl. Sci.*, vol. 9, no. 20, pp. 1–11, 2019.
- [176] X. Yuan, P. He, Q. Zhu, R. Bhat, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” Dec. 2017, *arXiv:1712.07107*.
- [177] Y. Shi and Y. Sagduyu, “Evasion and causative attacks with adversarial deep learning,” in *Proc. MILCOM*, Oct. 2017, pp. 243–248.
- [178] C. Szegedy *et al.*, “Intriguing properties of neural networks,” 2013, *arXiv:1312.6199*.
- [179] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” Dec. 2014, *arXiv:1412.6572*.
- [180] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *Proc. EuroSP*, Mar. 2016, pp. 372–387.
- [181] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 39–57.
- [182] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. Li, “Spectrum data poisoning with adversarial deep learning,” in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2018, pp. 407–412.
- [183] Y. Sagduyu, Y. Shi, and T. Erpek, “Adversarial deep learning for over-the-air spectrum poisoning attacks,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 306–319, Feb. 2021.
- [184] Y. E. Sagduyu, Y. Shi, and T. Erpek, “IoT network security from the perspective of adversarial deep learning,” in *Proc. 16th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, 2019, pp. 1–9.
- [185] Z. Luo, S. Zhao, Z. Lu, J. Xu, and Y. Sagduyu, “When attackers meet AI: Learning-empowered attacks in cooperative spectrum sensing,” Mar. 2019, *arXiv:1905.01430*.
- [186] B. Kim, Y. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, “Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels,” Feb. 2020, *arXiv:2002.02400*.
- [187] K. Yang, J. Liu, C. Zhang, and Y. Fang, “Adversarial examples against the deep learning based network intrusion detection systems,” in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2018, pp. 559–564.
- [188] M. Usama, M. Asim, S. Latif, J. Qadir, and A. Al-Fuqaha, “Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems,” in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2019, pp. 78–83.
- [189] Z. Lin, Y. Shi, and Z. Xue, “IDSGAN: Generative adversarial networks for attack generation against intrusion detection,” 2018, *arXiv:1809.02077*.
- [190] P. Li, W. Zhao, Q. Liu, X. Liu, and L. Yu, *Poisoning Machine Learning Based Wireless IDSs via Stealing Learning Model*. Shanghai, China: Springer, Jun. 2018, pp. 261–273.
- [191] D. Wu, B. Fang, J. Wang, Q. Liu, and X. Cui, “Evading machine learning botnet detection models via deep reinforcement learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.
- [192] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, “Adversarial attacks and defences: A survey,” 2018, *arXiv:1810.00069*.
- [193] Y. Al-Hadhrani and F. K. Hussain, “Real time dataset generation framework for intrusion detection systems in IoT,” *Future Gener. Comput. Syst.*, vol. 108, pp. 414–423, Jul. 2020.
- [194] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, “Ton_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems,” *IEEE Access*, vol. 8, pp. 165130–165150, 2020.