Review

Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity

Murat Kuzlu[1] · Corinne Fair[2] · Ozgur Guler[3]

Received: 29 September 2020 / Accepted: 30 November 2020

Abstract
In recent years, the use of the Internet of Things (IoT) has increased exponentially, and cybersecurity concerns have increased along with it. On the cutting edge of cybersecurity is Artificial Intelligence (AI), which is used for the development of complex algorithms to protect networks and systems, including IoT systems. However, cyber-attackers have figured out how to exploit AI and have even begun to use adversarial AI in order to carry out cybersecurity attacks. This review paper compiles information from several other surveys and research papers regarding IoT, AI, and attacks with and against AI and explores the relationship between these three topics with the purpose of comprehensively presenting and summarizing relevant literature in these fields.

Keywords Artificial Intelligence · Internet of Things (IoT) · Cybersecurity

1 Introduction

Since around 2008, when the Internet of Things (IoT) was born [1], its growth has been booming, and now IoT is a part of daily life and has a place in many homes and businesses. IoT is hard to define as it has been evolving and changing since its conception, but it can be best understood as a network of digital and analog machines and computing devices provided with unique identifiers (UIDs) that have the ability to exchange data without human intervention [2]. In most cases, this manifests as a human interfacing with a central hub device or application, often a mobile app, that then goes on to send data and instructions to one or multiple fringe IoT devices [3]. The fringe devices are able to complete functions if required and send data back to the hub device or application, which the human can then view.

The IoT concept has given the world a higher level of accessibility, integrity, availability, scalability, confidentiality, and interoperability in terms of device connectivity [4]. However, IoTs are vulnerable to cyberattacks due to a combination of their multiple attack surfaces and their newness and thus lack of security standardizations and requirements [5]. There are a large variety of cyberattacks that attackers can leverage against IoTs, depending on what aspect of the system they are targeting and what they hope to gain from the attack. As such, there is a large volume of research into cybersecurity surrounding IoT. This includes Artificial Intelligence (AI) approaches to protecting IoT systems from attackers, usually in terms of detecting unusual behavior that may indicate an attack is occurring [6]. However, in the case of IoT, cyber-attackers always have the upper hand as they only need to find one vulnerability while cybersecurity experts must protect multiple targets. This has led to increased use of AI by cyber-attackers as well, in order to thwart the complicated algorithms that detect anomalous activity and pass by unnoticed [7]. AI has received much attention with the growth of IoT technologies.

✉ Murat Kuzlu, mkuzlu@odu.edu | [1]Batten College of Engineering and Technology, Old Dominion University, Norfolk, VA, USA. [2]Computer Science, Christopher Newport University, Newport News, VA, USA. [3]eKare, Inc, Fairfax, VA, USA.

Published online: 24 February 2021

With this growth, AI technologies, such as decision trees, linear regression, machine learning, support vector machines, and neural networks, have been used in IoT cybersecurity applications to able to identify threats and potential attacks.

Authors in [8] provide a comprehensive review of the security risks related to IoT application and possible countermeasures as well as compare IoT technologies in terms of integrity, anonymity, confidentiality, privacy, access control, authentication, authorization, resilience, and self-organization. The authors propose deep learning models using CICIDS2017 datasets for DDoS attack detection for the cybersecurity in IoT (Internet of Things), which provide high accuracy, i.e., 97.16% [9]. In [10], the authors evaluate the Artificial Neural Networks (ANN) in a gateway device to able to detect anomalies in the data sent from the edge devices. The results show that the proposed approach can improve the security of IoT systems. The authors in [11] propose an AI-based control approach for detection and estimation as well as compensation of cyber attacks in industrial IoT systems. In [12], The authors provide a robust pervasive detection for IoT Environments and develop a variety of adversarial attacks and defense mechanisms against them as well as validate their approach through datasets including MNIST, CIFAR-10, and SVHN. In [13], the authors analyze the recent evolution of AI decision-making in cyber physical systems and find that such evolution is virtually autonomous due to the increasing integration of IoT devices in cyber physical systems, and the value of AI decision-making due to its speed and efficiency in handling large loads of data is likely going to make this evolution inevitable. The authors of [14] discuss new approaches to risk analytics using AI and machine learning, particularly in IoT networks present in industry settings. Finally, [15] discusses methods of capturing and assessing cybersecurity risks to IoT devices for the purpose of standardizing such practices so that risk in IoT systems may be more efficiently identified and protected against.

This review paper covers a variety of topics regarding cybersecurity, the Internet of Things (IoT), Artificial Intelligence (AI), and how they all relate to each other in three survey-style sections and provides a comprehensive review of cyberattacks against IoT devices as well as provides recommended AI-based methods of protecting against these attacks. The ultimate goal of this paper is to create a resource for others who are researching these prevalent topics by presenting summaries of and making connections between relevant works covering different aspects of these subjects.

2 Methods of attacking IoT devices

Due to the lax security in many IoT devices, cyberattackers have found many ways to attack IoT devices from many different attack surfaces. Attack surfaces can vary from the IoT device itself, both its hardware and software, the network on which the IoT device is connected to, and the application with which the device interfaces; these are the three most commonly used attack surfaces as together they make up the main parts of an IoT system. Figure 1 illustrates a basic breakdown of a common IoT system; most of the attacks discussed in this paper occur at the network gateway and/or cloud data server connections, as these connections are generally where IoT security is most lacking.

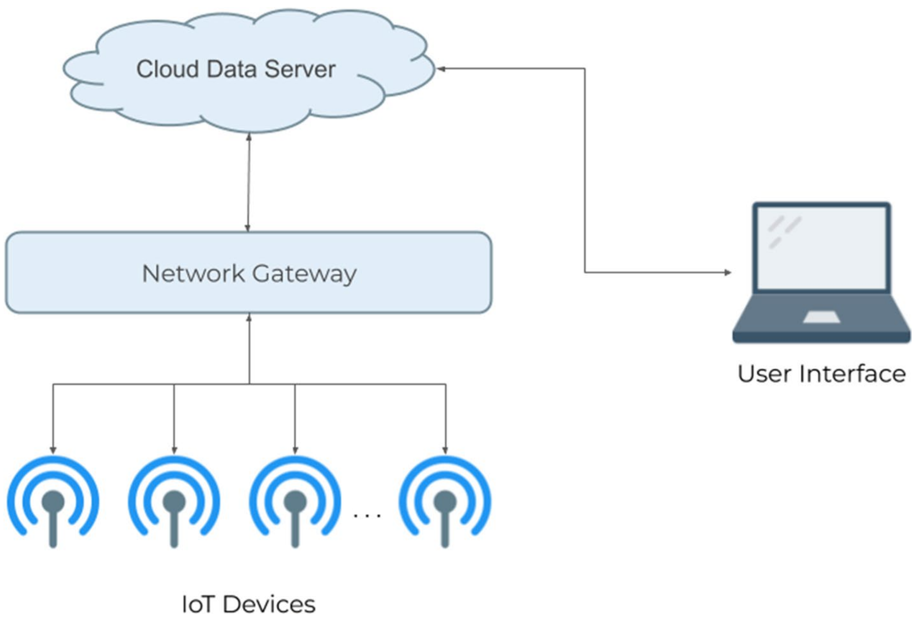

Fig. 1 A high-level breakdown of typical IoT structure

2.1 Initial reconnaissance

Before IoT attackers even attempt cyberattacks on an IoT device, they will often study the device to identify vulnerabilities. This is often done by buying a copy of the IoT device they are targeting from the market. They then reverse engineer the device to create a test attack to see what outputs can be obtained and what avenues exist to attack the device. Examples of this include opening up the device and analyzing the internal hardware—such as the flash memory—in order to learn about the software, and tampering with the microcontroller to identify sensitive information or cause unintended behavior [16]. In order to counter reverse engineering, it is important for IoT devices to have hardware-based security. The application processor, which consists of sensors, actuators, power supply, and connectivity, should be placed in a tamper-resistant environment [16]. Device authentication can also be done with hardware-based security, such that the device can prove to the server it is connected to that it is not fake.

2.2 Physical attacks

An often low-tech type category of attacks includes physical attacks, in which the hardware of the target device is used to the benefit of the attacker in some way. There are several different types of physical attacks. These include attacks such as outage attacks, where the network that the devices are connected to are shut off to disrupt their functions; physical damage, where devices or their components are damaged to prevent proper functionality; malicious code injection, an example of which includes an attacker plugging a USB containing a virus into the target device; and object jamming, in which signal jammers are used to block or manipulate the signals put out by the devices [17]. Permanent denial of service (PDoS) attacks, which are discussed later in this paper, can be carried out as a physical attack; if an IoT device is connected to a high voltage power source, for example, its power system may become overloaded and would then require replacement [18].

2.3 Man-in-the-Middle

One of the most popular attacks on IoTs is Man-in-the-Middle (MITM) attack. With regards to computers in general, an MITM attack intercepts communication between two nodes and allows the attacker to take the role of a proxy. Attackers can perform MITM attacks between many different connections such as a computer and a router, two cell phones, and, most commonly, a server and a client. Figure 2 shows a basic example of an MITM attack between a client and a server. In regards to IoT, the attacker usually performs MITM attacks between an IoT device and the application with which it interfaces. IoT devices, in particular, tend to be more vulnerable to MITM attacks as they lack the standard implementations to fight the attacks. There are two common modes of MITM attacks: cloud polling and direct connection. In cloud polling, the smart home device is in constant communication with the cloud, usually to look for firmware updates. Attackers can redirect network traffic using Address Resolution Protocol (ARP) poisoning or by altering Domain Name System (DNS) settings or intercept HTTPS traffic by using self-signed certificates or tools such as (Secure Sockets Layer) SSL strip

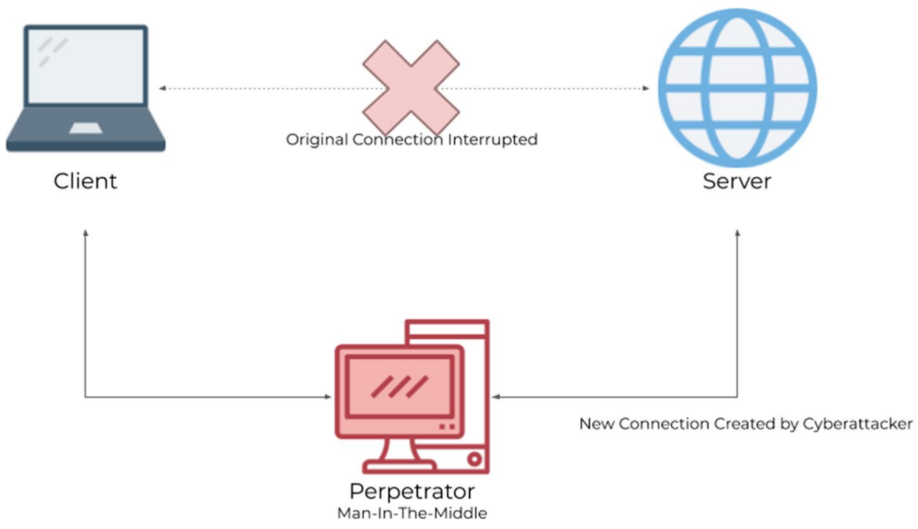
Original Connection Interrupted
Client
Server
New Connection Created by Cyberattacker
Perpetrator
Man-In-The-Middle

Fig. 2 A simple representation of a Man-in-the-Middle attack

[19]. Many IoT devices do not verify the authenticity or the trust level of certificates, making the self-signed certificate method particularly effective. In the case of direct connections, devices communicate with a hub or application in the same network. By doing this, mobile apps can locate new devices by probing every IP address on the local network for a specific port. An attacker can do the same thing to discover devices on the network [19]. An example of an MITM IoT attack is that of a smart refrigerator that could display the user's Google calendar. It seems like a harmless feature, but attackers found that the system did not validate SSL certificates, which allowed them to perform an MITM attack and steal the user's Google credentials [19].

2.3.1 Bluetooth Man-in-the-Middle

A common form of MITM attack leveraged against IoT devices is via Bluetooth connection. Many IoT devices run Bluetooth Low Energy (BLE), which is designed with IoT devices in mind to be smaller, cheaper, and more power-efficient [20]. However, BLE is vulnerable to MITM attacks. BLE uses AES-CCM encryption; AES encryption is considered secure, but the way that the encryption keys are exchanged is often insecure. The level of security relies on the pairing method used to exchange temporary keys between the devices. BLE specifically uses three-phase pairing processes: first, the initiating device sends a pairing request, and the devices exchange pairing capabilities over an insecure channel; second, the devices exchange temporary keys and verify that they are using the same temporary key, which is then used to generate a short-term key (some newer devices use a long-term key exchanged using Elliptic Curve Diffie-Hellman public-key cryptography, which is significantly more secure than the standard BLE protocol); third, the created key is exchanged over a secure connection and can be used to encrypt data [20]. Figure 3 represents this three-phase pairing process.

The temporary key is determined according to the pairing method, which is determined on the OS level of the device. There are three common pairing methods popular with IoT devices. One, called Just Works, always sets the temporary key to 0, which is obviously very insecure. However, it remains one of if not the most popular pairing methods used with BLE devices [20]. The second, Passkey, uses six-digit number combinations, which the user must manually enter into a device, which is fairly secure, though there are methods of bypassing this [20]. Finally, the Out-of-Band pairing method exchanges temporary keys using methods such as Near Field Communication. The security level of this method is determined by the security capabilities of the exchange method. If the exchange channel is protected from MITM attacks, the BLE connection can also be considered protected. Unfortunately, the Out-of-Band method is not yet common in IoT devices [20]. Another important feature of BLE devices is the Generic Attribute Profile (GATT), which is used to communicate between devices using a standardized data schema. The GATT describes devices' roles, general behaviors, and other metadata. Any BLE-supported app within the range of an IoT device can read its GATT schema, which provides the app with necessary information [20]. In order for attackers to perform MITM attacks in BLE networks, the attacker must use two connected BLE devices himself: one device acting as the IoT device to connect to the target mobile app, and a fake mobile app to connect to the target IoT device. Some other tools for BLE MITM attacks exist, such as GATTacker, a

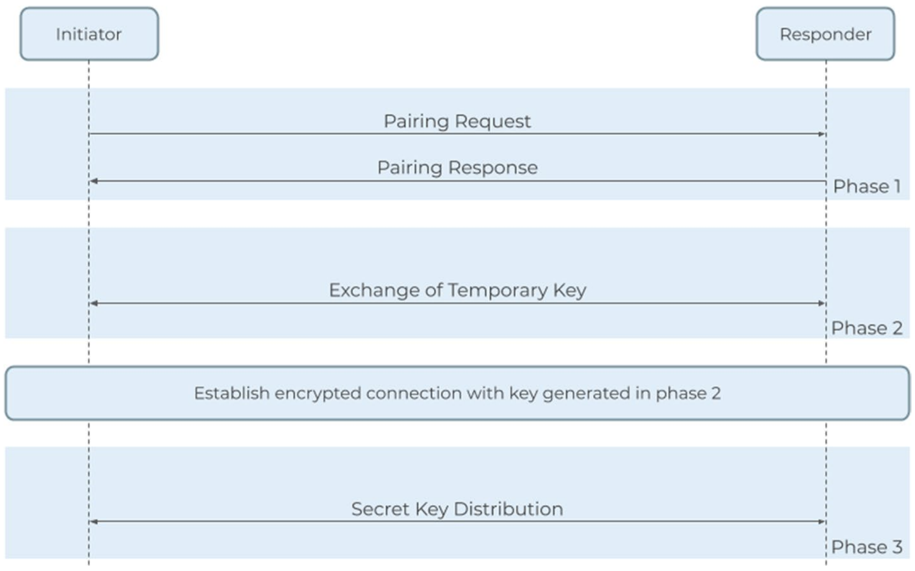

Fig. 3 A diagram illustrating the basic BLE pairing process

Node.js package that scans and copies BLE signals and then runs a cloned version of the IoT device, and BtleJuice, which allows MITM attacks on Bluetooth Smart devices which have improved security over BLE [20].

2.3.2 False data injection attacks

Once an attacker has access to some or all of the devices on an IoT network via an MITM attack, one example of an attack they could carry out next is a False Data Injection (FDI) attack. FDI attacks are when an attacker alters measurements from IoT sensors by a small amount so as to avoid suspicion and then outputs the faulty data [21]. FDI attacks can be perpetrated in a number of ways, but in practice doing so via MITM attacks is the most practical. FDI attacks are often leveraged against sensors that send data to an algorithm that attempts to make predictions based on the data it has received or otherwise uses data to make conclusions. These algorithms, sometimes referred to as predictive maintenance systems, are commonly used in monitoring the state of a mechanical machine and predicting when it will need to be maintained or tuned [21]. These predictive maintenance algorithms and similar would also be a staple feature of smart cities, FDI attacks against which could be disastrous. An example of an FDI attack on a predictive maintenance system is sensors on an airplane engine that predict when the engine will need critical maintenance. When attackers are able to access even a small portion of the sensors, they are able to create a small amount of noise that goes undetected by faulty data detection mechanisms but is just enough to skew the algorithm's predictions [21]. In testing, it would even be enough to delay critical maintenance to the system, potentially causing catastrophic failure while in use, which could cause a costly unplanned delay or loss of life.

2.4 Botnets

Another kind of common attack on IoT devices is recruiting many devices to create botnets and launch Distributed Denial of Service (DDoS) attacks. A denial of service (DoS) attack is characterized by an orchestrated effort to prevent legitimate use of a service; a DDoS attack uses attacks from multiple entities to achieve this goal. DDoS attacks aim to overwhelm the infrastructure of the target service and disrupt normal data flow. DDoS attacks generally go through a few phases: recruitment, in which the attacker scans for vulnerable machines to be used in the DDoS attack against the target; exploitation and infection, in which the vulnerable machines are exploited, and malicious code is injected; communication, in which the attacker assesses the infected machines, sees which are online and decides when to schedule attacks or upgrade the machines; and attack, in which the attacker commands the infected machines to send malicious packets to the target [22]. One of the most popular ways to gain infected machines and conduct DDoS attacks is through IoT devices due to their high availability and generally poor security and maintenance. Figure 4 shows a common command structure, in which the attacker's master computer sends commands to one or more infected command and control centers, who each control a series of zombie devices that can then attack the target.

One of the most famous malware, the Mirai worm, has been used to perpetrate some of the largest DDoS attacks ever known and is designed to infect and control IoT devices such as DVRs, CCTV cameras, and home routers. The infected devices become part of a large-scale botnet and can perpetrate several types of DDoS attacks. Mirai was built to handle multiple different CPU architectures that are popular to use in IoT devices, such as x86, ARM, Sparc, PowerPC, Motorola,

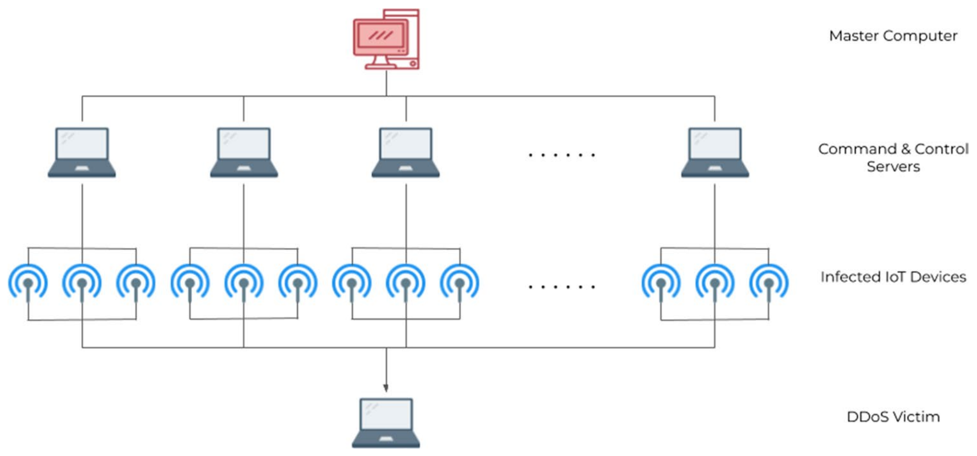

Fig. 4 A graphical representation of a common botnet hierarchy

etc., in order to capture as many devices as possible [23]. In order to be covert, the virus is quite small and actually does not reside in the device's hard disk. It stays in memory, which means that once the device is rebooted, the virus is lost. However, devices that have been infected once are susceptible to reinfection due to having already been discovered as being vulnerable, and reinfection can take as little as a few minutes [23]. Today, many well-known IoT-targeting botnet viruses are derived from Mirai's source code, including Okiru, Satori, and Reaper [23].

2.5 Denial of service attacks

IoT devices may often carry out DoS attacks, but they themselves are susceptible to them as well. IoT devices are particularly susceptible to permanent denial of service (PDoS) attacks that render a device or system completely inoperable. This can be done by overloading the battery or power systems or, more popularly, firmware attacks. In a firmware attack, the attacker may use vulnerabilities to replace a device's basic software (usually its operating system) with a corrupted or defective version of the software, rendering it useless [18]. This process, when done legitimately, is known as flashing, and its illegitimate counterpart is known as "phlashing". When a device is phlashed, the owner of the device has no choice but to flash the device with a clean copy of the OS and any content that might've been put on the device. In a particularly powerful attack, the corrupted software could overwork the hardware of the device such that recovery is impossible without replacing parts of the device [18]. The attacks to the device's power system, though less popular, are possibly even more devastating. One example of this type of attack is a USB device with malware loaded on it that, when plugged into a computer, overuses the device's power to the point that the hardware of the device is rendered completely ruined and needs to be replaced [18].

One example of PDoS malware is known as BrickerBot. BrickerBot uses brute force dictionary attacks to gain access to IoT devices and, once logged in to the device, runs a series of commands that result in permanent damage to the device. These commands include misconfiguring the device's storage and kernel parameters, hindering internet connection, sabotaging device performance, and wiping all files on the device [24]. This attack is devastating enough that it often requires reinstallation of hardware or complete replacement of the device. If the hardware survives the attack, the software certainly didn't and would need reflashing, which would lose everything that might have been on it. Interestingly enough, BrickerBot was designed to target the same devices the Mirai botnet targets and would employ as bots, and uses the same or a similar dictionary to make its brute force attacks. As it turns out, BrickerBot was actually intended to render useless those devices that Mirai would have been able to recruit in an effort to fight back against the botnet [24].

Due to the structure of IoT systems, there are multiple attack surfaces, but the most popular way of attacking IoT systems is through their connections as these tend to be the weakest links. In the future, it is advisable that IoT developers ensure that their products have strong protections against such attacks, and the introduction of IoT security standards would prevent users from unknowingly purchasing products that are insecure. Alternatively, keeping the network that the IoT system resides on secure will help prevent many popular attacks, and keeping the system largely separated from other critical systems or having backup measures will help mitigate the damage done should an attack be carried out.

3 Artificial Intelligence in cybersecurity

In order to dynamically protect systems from cyber threats, many cybersecurity experts are turning to Artificial Intelligence (AI). AI is most commonly used for intrusion detection in cybersecurity by analyzing traffic patterns and looking for an activity that is characteristic of an attack.

3.1 Machine learning

There are two main kinds of machine learning: supervised and unsupervised learning. Supervised learning is when humans manually label training data as malicious or legitimate and then input that data into the algorithm to create a model that has "classes" of data that it compares the traffic it is analyzing. Unsupervised learning forgoes training data and manual labeling, and instead the algorithm groups together similar pieces of data into classes and then classifies them according to the data coherence within one class and the data modularity between classes [25]. One popular machine learning algorithm for cybersecurity is naïve Bayes, which seeks to classify data based on the Bayesian theorem wherein anomalous activities are all assumed to originate from independent events instead of one attack. Naïve Bayes is a supervised learning algorithm, and once it is trained and has generated its classes will analyze each activity to determine the probability that it is anomalous [25]. Machine learning algorithms can also be used to create the other models discussed in this section

3.2 Decision trees

A decision tree is a type of AI that creates a set of rules based on its training data samples. It uses iterative division to find a description (often simply "attack" or "normal") that best categorizes the traffic it is analyzing. An example of this approach in cybersecurity is detecting DoS attacks by analyzing the flow rate, size, and duration of traffic. For example, if the flow rate is low, but the duration of the traffic is long, it is likely to be an attack and will, therefore, be classified as such [25]. Decision trees can also be used to detect command injection attacks in robotic vehicles by categorizing values from CPU consumption, network flow, and volume of data written [25] as shown in Fig. 5. This technique is popular as it is intuitive in that what the AI does and doesn't consider anomalous traffic is known to the developer. Additionally, once an effective series of rules is found, the AI can analyze traffic in real-time, providing an almost immediate alert if unusual activity is detected.

Another approach to decision trees is the Rule-Learning technique, which searches for a set of attack characteristics in each iteration while maximizing some score that denotes the quality of the classification (i.e., the number of incorrectly classified data samples) [25]. The main difference between traditional decision trees and the rule-learning techniques is that traditional decision trees look for characteristics that will lead to a classification, whereas the rule-learning technique finds a complete set of rules that can describe a class. This can be an advantage as it can factor in human advice when generating rules, which creates an optimized set of rules [25].

3.3 K-nearest neighbors

The k-nearest neighbor (k-NN) technique learns from data samples to create classes by analyzing the Euclidean distance between a new piece of data and already classified pieces of data to decide what class the new piece should be put in, to put it simply [25]. For example, the new piece of data when k, the number of nearest neighbors, equals three (3) would be classified into class two (2), but when k equals nine (9), the new piece would be classified in class 1 as shown in Fig. 6. The k-NN technique is attractive for intrusion detection systems as it can quickly learn from new traffic patterns to notice previously unseen, even zero-day attacks. Cybersecurity experts are also researching applications of k-NN for real-time detection of cyberattacks [25]. The technique has been employed to detect attacks such as false data injection attacks and performs well when data can be represented through a model that allows the measurement of their distance to other data, i.e., through a Gaussian distribution or a vector.

3.4 Support vector machines

Support vector machines (SVMs) are an extension of linear regression models that locates a plane that separates data into two classes [25]. This plane can be linear, non-linear, polynomial, Gaussian, sigmoid, etc., depending on the function used in the algorithm. SVMs can also separate data into more than two classes by using more than one plane. In cybersecurity, this technique is used to analyze Internet traffic patterns and separate them into their component classes

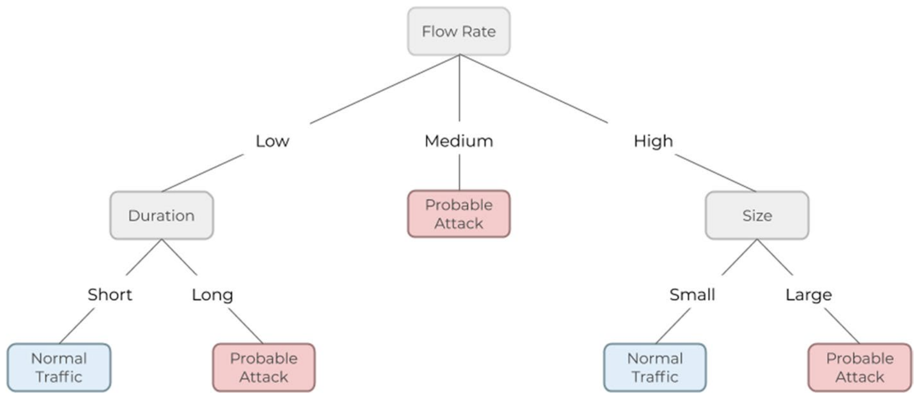

Fig. 5 An example of a decision tree for classifying network traffic

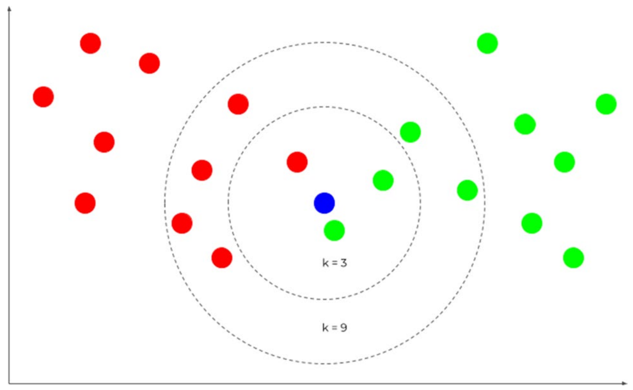

Fig. 6 How k-NN technique can classify a data point differently given different k values

such as HTTP, FTP, SMTP, and so on [25]. As SVM is a supervised machine learning technique, it is often used in applications where attacks can be simulated, such as using network traffic generated from penetration testing as training data.

3.5 Artificial neural networks

Artificial neural networks (ANNs) are a technique derived from the way that neurons interact with each other in the brain in order to pass and interpret information. In ANNs, a neuron is a mathematical equation that reads data and outputs a target value, which is then passed along to the next neuron based on its value. The ANN algorithm then iterates until the output value is acceptably close to the target value, which allows the neurons to learn and correct their weights by measuring the error between the expected value and the previous output value. Once this process is finished, the algorithm presents a mathematical equation that outputs a value that can be used to classify the data [25].

A large benefit of ANNs is that they are able to adjust their mathematical models when presented with new information, whereas other mathematical models may become obsolete as new types of traffic and attacks become common [25]. This also means that ANNs are adept at catching previously unseen and zero-day attacks as they take new information into heavier consideration than static mathematical models can. Because of this, ANNs make solid intrusion detection systems and have performed well with attacks such as DoS [25].

At present, using AI in cybersecurity is a small but rapidly growing field. It is also expensive and resource intensive, so using AI to protect a small system may not be feasible. However, businesses that have large networks may benefit from these solutions, especially if they are considering or have already introduced IoT devices into their network. AI cybersecurity would also be beneficial in the massive systems one would find in a smart city, and the AI would be able to give very quick response times that are important in systems like traffic management. In the future, AI cybersecurity could also be integrated into smaller systems such as self-driving cars or smart homes. Additionally, many AI cybersecurity measures detect or thwart attacks in progress rather than preventing attacks in the first place, meaning that other preventative security measured should also be in place.

4 AI to attack IoT

Not all AI is used for the purposes of cybersecurity; cybercriminals have begun using malicious AI to aid attacks, often to thwart the intrusion detection algorithms in the case of IoT, or attacking beneficial AI in such a way that the AI works against its own system.

4.1 Automation of vulnerability detection

Machine learning can be used to discover vulnerabilities in a system. While this can be useful for those trying to secure a system to intelligently search for vulnerabilities that need to be patched, attackers also use this technology to locate and exploit vulnerabilities in their target system. As technology soars in usage, especially technologies with low-security standards such as IoT devices, the number of vulnerabilities that attackers are able to exploit has soared as well, including zero-day vulnerabilities. In order to identify vulnerabilities quickly, attackers often use AI to discover vulnerabilities and exploit them much more quickly than developers can fix them. Developers are able to use these detection tools as well, but it should be noted that developers are at a disadvantage when it comes to securing a system or device; they must find and correct every single vulnerability that could potentially exist, while attackers need only find one, making automatic detection a valuable tool for attackers.

4.1.1 Fuzzing

Fuzzing, at its core, is a testing method that generates random inputs (i.e., numbers, chars, metadata, binary, and especially "known-to-be-dangerous" values such as zero, negative or very large numbers, SQL requests, special characters) that causes the target software to crash [26]. It can be divided into dumb fuzzing and smart fuzzing. Dumb fuzzing simply generates defects by randomly changing the input variables; this is very fast as changing the input variable is simple, but it is not very good at finding defects as code coverage is narrow [26]. Smart fuzzing, on the other hand, generates input values suitable for the target software based on the software's format and error generation. This software analysis is a big advantage for smart fuzzing as it allows the fuzzing algorithm to know where errors can occur; however, developing an efficient smart fuzzing algorithm takes expert knowledge and tuning [26].

4.1.2 Symbolic execution

Symbolic execution is a technique similar to fuzzing that searches for vulnerabilities by setting input variables to a symbol instead of a real value [26]. This technique is often split into offline and online symbolic execution. Offline symbolic execution chooses only one path to explore at a time to create new input variables by resolving the path predicate [26]. This means that each time one wishes to explore a new path, the algorithm must be run from the beginning, which is a disadvantage due to the large amount of overhead due to code re-execution. Online symbolic execution replicates states and generates path predicates at every branch statement [26]. This method does not incur much overhead, but it does require a large amount of storage to store all the status information and simultaneous processing of all the states it creates, leading to significant resource consumption.

4.2 Input attacks

When an attacker alters the input of an AI system in such a way that causes the AI to malfunction or give an incorrect output, it is known as an input attack. Input attacks are carried out by adding an attack pattern to the input, which can be anything from putting tape on a physical stop sign to confuse self-driving cars to adding small amounts of noise to an image that is imperceptible to the human eye but will confuse an AI [27]. Notably, the actual algorithm and security of the AI does not need to be compromised in order to carry out an input attack—only the input that the attacker wants to compromise the output of must be altered. In the case of tape on a stop sign, the attacker may not need to use technology at all. However, more sophisticated attacks are completely hidden from the human eye, wherein the attacker may alter a tiny part of the image in a very precise manner that is designed to misdirect the algorithm. That being said, input attacks are often categorized based on where they rest on two axes: perceivability and format.

The perceivability of an input attack is the measure of how noticeable the attack is to the human eye, while the format is the measure of how digital versus physical the attack is [27]. On one end of the perceivability axis is perceivable attacks. Altering targets, such as by deforming, removing part of, or changing its colors, and adding to the target, such as affixing physical tape or adding digital marks, are types of perceivable attacks [27]. While perceivable attacks are perceivable by humans, humans may not notice slight changes like tape on a stop sign or consider them important. A human driver still sees a stop sign with tape or scratches as a stop sign, even though a self-driving car may not. This lends itself to the effectiveness of perceivable attacks, allowing them to, in many cases, hide in plain sight. Conversely, imperceivable attacks are invisible to the human eye. This can include things such as "digital dust," which is a small amount of noise added to the entire image that is not visible to the human eye but significant enough to an AI to change its output or an imperceptible pattern on a 3D printed object that can be picked up by AI [27]. Imperceivable attacks can also be made through audio, such as playing audio at ranges outside of the human hearing range that would be picked up by a microphone [27]. Imperceivable attacks are generally more of a security risk, as there is almost no chance that a human would notice the attack before the AI algorithm outputs an incorrect response.

The format of an attack is usually either digital or physical, without many attacks that are a combination of both [27]. In many cases of physical attacks, the attack pattern must be more obvious rather than imperceivable as physical objects must be digitized to be processed and, in that process, may lose some finer detail [27]. Some attacks are still difficult to perceive even with the detail loss, however, as with the case of 3D printed objects with a pattern that blends into the structure of the object such that it is imperceptible to humans [27]. Opposite of physical attacks are digital attacks, which attack digital inputs such as images, videos, audio recordings, and files. As these inputs are already digitized, there is no process wherein detail is lost, and as such attackers can make very exact attacks, allowing them to be more imperceptible to the human eye than physical attacks [27]. Digital attacks are not necessarily imperceptible. However—photoshopping glasses with a strange pattern over a celebrity, for example, may cause the AI to identify the image as a different person, but still a person nonetheless. An example of input attacks specific to IoT smart cars and, more broadly, smart cities. As mentioned earlier, simply placing pieces of tape in a specific way on a stop sign is enough for an algorithm to not recognize the stop sign or even classify it as a green light—this is harmful for passengers in the car if the car does not heed the stop sign, and at a larger scale could alter traffic pattern detectors in smart cities. Additionally, noise-based input attacks could cause smart assistants to malfunction and carry out unintended commands.

4.3 Data poisoning/false data injection

Data poisoning attacks and input attacks are very similar, but while the goal of input attacks is simply to alter the output of the affected input, the goal of data poisoning is to alter inputs over a long enough period of time that the AI that analyzes data has shifted and is inherently flawed; because of this, data poisoning is usually carried out while the AI is still being trained before it is actually deployed [27]. In many cases, the AI learns to fail on specific inputs that the attacker chooses; for example, if a military uses AI to detect aircraft, the enemy military may poison the AI so that it does not recognize certain types of aircraft like drones [27]. Data poisoning can also be used on AIs that are constantly learning and analyzing data in order to make and adjust predictions, such as in predictive maintenance systems. There are three main methods attackers can use to poison an AI.

4.3.1 Dataset poisoning

Poisoning the dataset of an AI is perhaps the most direct method of data poisoning—as AI gain all of their knowledge from the training datasets they are provided, any flaws within those datasets will subsequently flaw the AI's knowledge. A basic example of this is shown in Fig. 7: a significant portion of the data is corrupted in the second dataset, leading the resultant machine learning model to be flawed. Dataset poisoning is done by including incorrect or mislabeled information in the target dataset [27]. As AI learn by recognizing patterns in datasets, poisoned datasets break patterns or may introduce new incorrect patterns, causing the AI to misidentify inputs or identify them incorrectly [27]. Many datasets are very large, so finding poisoned data within datasets can be difficult. Continuing the example of traffic patterns, an attacker could change dataset labels in such a way that the AI no longer recognizes stop signs or add data and labels that cause the AI to classify a red light as a green light.

4.3.2 Algorithm poisoning

Algorithm poisoning attacks take advantage of weaknesses that may be in the learning algorithm of the AI. This method of attack is very prominent in federated learning, which is a method of training machine learning while protecting data privacy of an individual. Federated learning, rather than collecting potentially sensitive data from users and combining it into one dataset, trains small models directly on users' devices and then combines these models to form the final model. The users' data never leaves their devices, and so is more secure; however, if an attacker is one of the users that the algorithm is using the data of, they are free to manipulate their own data in order to poison the model [27]. The poisoned algorithm, when combined with the rest of the algorithms, has the potential to poison the final model. They could degrade the model or even install a backdoor in this manner.

One example of federated learning is Google's Gboard, which used federated learning to learn about text patterns in order to train predictive keyboards [28]. Although Google has extensive data vetting measures, in a less careful approach, users could potentially type nonsensical messages to confuse the predictive text or, more sinisterly, inject code into the algorithm to give themselves a backdoor. Similarly, some cutting-edge IoT devices are beginning to employ federated learning in order to learn from each other. One example of this is using machine learning to predict air pressure changes as it flows through gradually clogging filters, allowing the IoT sensor to predict when the filter will need to be changed [29]. This learning process would take a long enough time to make the study infeasible with just a few filters, but with federated learning the process is able to be sped up significantly. However, users could easily manipulate the process with

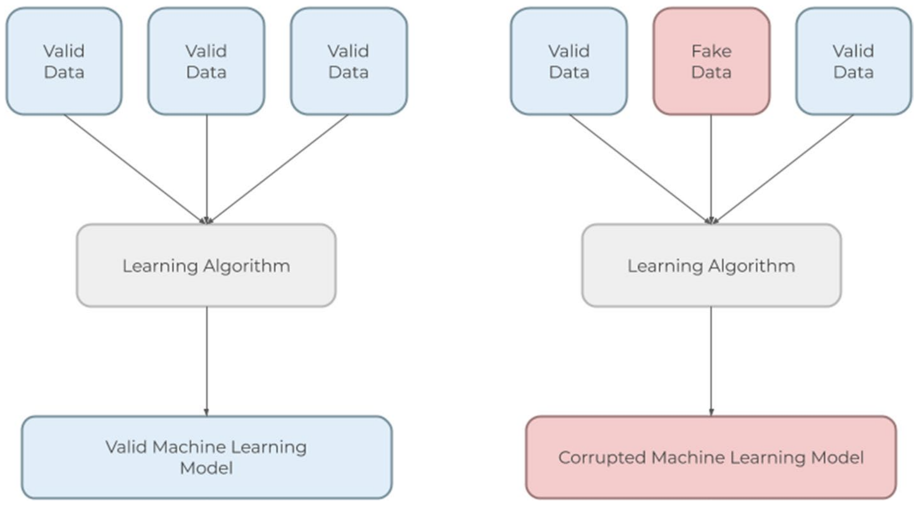

Fig. 7 A visual representation of dataset poisoning

their own filters in order to poison the algorithm. Although this is a relatively innocent example of algorithm poisoning, as federated learning increases in IoT, so will the potentially harmful applications of federated learning.

4.3.3 Model poisoning

Finally, some attackers simply replace a legitimate model with an already poisoned model prepared ahead of time; all the attacker has to do is get into the system which stores the model and replace the file [27]. Alternatively, the equations and data within the trained model file could be altered. This method is potentially dangerous as even if a model trained model is double-checked and data is verified to be not poisoned, the attacker can still alter the model at various points in its distribution, such as while the model is still in company's network awaiting placement on an IoT device or on an individual IoT device once it has been distributed [27].

Many of the attacks as described above can be mitigated or prevented by properly sanitizing inputs and checking for unusual data. However, some attacks are subtle and can bypass the notice of humans and even other AI, especially when the attacks are created by malevolent AI systems. These attacks and how to defend against effectively them are at the forefront of current research as the popularity of these attacks grow, but at present many attacks do not use AI for the same reason that many security systems do not: AI is resource intensive and a good algorithm requires high-level knowledge to build, making it inaccessible and infeasible to many attackers.

5 Summary of attacks and their defenses

The various attacks discussed in this paper are listed in Table 1, and are paired with one or more ways of protecting an IoT system from the attack. While comprehensively protecting an IoT system can be a challenging task due to the number of attack surfaces present, many of the methods listed will defend against many types of attacks; for example, as many of the attacks listed are carried out by first conducting MITM attacks, protecting the network on which an IoT system resides will protect the system from many common attacks.

6 Conclusion

Due to the nature of IoT systems to have many attack surfaces, there exists a variety of attacks against these systems, and more are being discovered as IoT grows in popularity. It is necessary to protect systems against these attacks as effectively as possible. As the number and speed of attacks grow, experts are turning to AI as a means of protecting these systems intelligently and in real-time. Of course, attackers find ways to thwart these AI and may even use AI to attack systems. This paper explores popular techniques to attempt to disrupt or compromise IoT and explains at a surface level how these attacks are carried out. Where applicable, examples are also provided in order to clarify these explanations. Next, several

Table 1 IoT attack and methods of protecting against the attack

IoT attack	Methods of protecting against the attack
Physical attacks	Use tamper-resistant hardware, hardware-based security trust anchors [16], employ kill commands, self-destruction [17]
Man-in-the-Middle	Run regular software updates, use proper firewall configuration, strong encryption, avoid insecure WiFi [19]
Bluetooth MITM	Make devices non-discoverable, run regular software updates, block unknown devices, use two-factor authentication [17], use strong pairing methods, such as Elliptic Curve Diffie-Hellman public-key cryptography or the Out-of-Band method [21]
False data injection	Run regular software updates, use firewalls with proper configuration, avoid insecure WiFi [19], employ anomaly detection and look for unusual output [21]
Botnets	Run regular antivirus scans, avoid suspicious email attachments or download links, run regular updates [30]
Mirai botnet	Run regular updates, change the default login credentials of IoT devices [31]
Denial of service attacks	Employ a DoS protection service, use an antivirus and firewall [32]
BrickerBot	Run regular updates, minimize network/internet exposure to IoT devices, use firewalls, configure authentication mechanisms [24]
Fuzzing and symbolic execution	Properly sanitize any inputs, limit allowed inputs [26]
Dataset poisoning	Employ outlier detection such as data sanitization and anomaly detection, employ micromodels [33]
Algorithm poisoning	Employ a validation dataset to remove local models that have a large negative impact on error rate or result in large loss [34]
Model poisoning	Protect the system that the model resides on [27]

AI algorithms are introduced, and their applications in cybersecurity are investigated. In many cases, these models are not yet common in commercial applications but rather are still undergoing research and development or are still difficult to implement and thus rare. Nonetheless, the models discussed are promising and may become common attack detection systems within just a couple of years. Methods of attacking AI and using AI to attack are also discussed, with the frame of IoT systems. The growth of IoT systems will see these types of attacks become more and more of a threat, especially as massive networks such as smart cities begin experimentation; both as massive networks are harder to protect with a multitude of attack surfaces, and as daily life and safety revolve around AI which needs to be more or less failure-proof. This is followed by a chart reiterating the threats covered in this paper, paired with common or recommended methods of protecting against each attack. Having covered all these topics, this paper aims to provide a useful tool with which researchers and cybersecurity professionals may study IoT in the context of cybersecurity and AI in order to secure IoT systems. Additionally, it also aims to emphasize the implications of up and coming technology and the impacts that each of these fields will have on the others. It is important to consider all the potential consequences of a technological development both before and after it is made public, as cyberattackers are constantly looking to use new technologies to their benefit, whether this means diverting the technology from its original purpose or using the technology as a tool to perpetuate other attacks. This paper discusses how IoT and AI have been taken advantage of for criminal purposes or have had weaknesses exploited as an example of this, which will help readers understand current risks and help cultivate an understanding such that these weaknesses are accounted for in the future in order to prevent cyberattacks.

Acknowledgements This work was supported in part by the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation and workforce development in Virginia, USA. For more information about CCI, visit cyberinitiative.org.

Authors' contributions MK, and CF conceived and designed the work as well as contributed to the acquisition, analysis, and interpretation of data. All authors discussed the results and wrote the final manuscript. All authors read and approved the final manuscript.

Competing interests The authors declare that they no competing interests.

References

1. Evans D. The Internet of Things: how the next evolution of the internet is changing everything. Cisco Internet Business Solutions Group: Cisco; 2011.
2. Rouse M. What is IoT (Internet of Things) and how does it work? IoT Agenda, TechTarget. http://www.internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT. Accessed 11 Feb 2020.
3. Linthicum D. App nirvana: when the internet of things meets the API economy. https://techbeacon.com/app-dev-testing/app-nirvana-when-internet-things-meets-api-economy. Accessed 15 Nov 2019.
4. Lu Y, Xu LD. Internet of Things (IoT) cybersecurity research: a review of current research topics. IEEE Internet Things J. 2019;6(2):2103–15.
5. Vorakulpipat C, Rattanalerdnusorn E, Thaenkaew P, Hai HD. Recent challenges, trends, and concerns related to IoT security: aan evolutionary study. In: 2018 20th international conference on advanced communication technology (ICACT), Chuncheon-si Gangwon-do, Korea (South); 2018. p. 405–10.
6. Lakhani A. The role of artificial intelligence in IoT and OT security. https://www.csoonline.com/article/3317836/the-role-of-artificial-intelligence-in-iot-and-ot-security.html. Accessed 11 Feb 2020.
7. Pendse A. Transforming cybersecurity with AI and ML: view. https://ciso.economictimes.indiatimes.com/news/transforming-cybersecurity-with-ai-and-ml/67899197. Accessed 12 Feb 2020.
8. Meneghello F, Calore M, Zucchetto D, Polese M, Zanella A. IoT: internet of threats? A survey of practical security vulnerabilities in real IoT devices. IEEE Internet Things J. 2019;6(5):8182–201.
9. Roopak M, Yun Tian G, Chambers J. Models deep learning, for cyber security in IoT networks. In: IEEE 9th annual computing and communication workshop and conference (CCWC), Las Vegas, NV, USA. 2019;2019:0452–7.
10. Cañedo J, Skjellum A. Using machine learning to secure IoT systems. In: 2016 14th annual conference on privacy, security and trust (PST), Auckland; 2016. p. 219–22, https://doi.org/10.1109/PST.2016.7906930.
11. Farivar F, Haghighi MS, Jolfaei A, Alazab M. Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT. IEEE Trans Ind Inf. 2020;16(4):2716–25. https://doi.org/10.1109/TII.2019.2956474.
12. Wang S, Qiao Z. Robust pervasive detection for adversarial samples of artificial intelligence in IoT environments. IEEE Access. 2019;7:88693–704. https://doi.org/10.1109/ACCESS.2019.2919695.
13. Radanliev P, De Roure D, Van Kleek M, Santos O, Ani U. Artificial intelligence in cyber physical systems. AI & society. 2020; p. 1–14.
14. Radanliev P, De Roure D, Page K, Nurse JR, Mantilla Montalvo R, Santos O, Maddox LT, Burnap P. Cyber risk at the edge: current and future trends on cyber risk analytics and artificial intelligence in the industrial internet of things and industry 4.0 supply chains. Cybersecurity. 2020;3:1–21.
15. Radanliev P, De Roure DC, Nurse JR, Montalvo RM, Cannady S, Santos O, Burnap P, Maple C. Future developments in standardisation of cyber risk in the Internet of Things (IoT). SN Appl Sci. 2020;2(2):169.
16. Woo S. The right security for IoT: physical attacks and how to counter them. In: Minj VP, editor. Profit From IoT. http://www.iot.electronicsforu.com/headlines/the-right-security-for-iot-physical-attacks-and-how-to-counter-them/. Accessed 13 June 2019.
17. Akram H, Dimitri K, Mohammed M. A comprehensive iot attacks survey based on a building-blocked reference mode. Int J Adv Comput Sci Appl. 2018. https://doi.org/10.14569/IJACSA.2018.090349.
18. Herberger C. DDoS fire & forget: PDoS—a permanent denial of service. Radware Blog, Radware Ltd. http://www.blog.radware.com/security/2015/10/ddos-fire-forget-pdos-a-permanent-denial-of-service/. Accessed 12 Sept 2016.
19. Cekerevac Z, Dvorak Z, Prigoda L, Čekerevac P. Internet of things and the man-in-the-middle attacks–security and economic risks. Mest J. 2017;5:15–25. https://doi.org/10.12709/mest.05.05.02.03.
20. Melamed T. An active man-in-the-middle attack on bluetooth smart devices. WIT Press, International Journal of Safety and Security Engineering. http://www.witpress.com/elibrary/sse-volumes/8/2/2120. Accessed 1 Feb 2018.
21. Mode G, Calyam P, Hoque K. False data injection attacks in Internet of Things and deep learning enabled predictive analytics; 2019.
22. De Donno M, Dragoni N, Giaretta A, Spognardi A. Analysis of DDoS-capable IoT malwares. In: 2017 federated conference on computer science and information systems (FedCSIS), Prague; 2017. p. 807–16. https://doi.org/10.15439/2017F288.
23. Mirai Botnet DDoS Attack. Corero, Corero. http://www.corero.com/resource-hub/mirai-botnet-ddos-attack/. Accessed 9 Dec 2019.
24. BrickerBot Malware emerges, permanently bricks iot devices. Trend Micro, Trend Micro Incorporated. http://www.trendmicro.com/vinfo/us/security/news/internet-of-things/brickerbot-malware-permanently-bricks-iot-devices. Accessed 19 Apr 2017.
25. Zeadally S, Adi E, Baig Z, Khan IA. Harnessing artificial intelligence capabilities to improve cybersecurity. IEEE Access. 2020;8:23817–37.
26. Jurn J, Kim T, Kim H. An automated vulnerability detection and remediation method for software security. Sustainability. 2018;10:1652. https://doi.org/10.3390/su10051652.
27. Comiter M. Attacking artificial intelligence. Belfer Center for Science and International Affairs, Belfer Center for Science and International Affairs. http://www.belfercenter.org/sites/default/files/2019-08/AttackingAI/AttackingAI.pdf. Accessed 25 Aug 2019.
28. McMahan B, Daniel R. Federated learning: collaborative machine learning without centralized training data. Google AI Blog, Google. http://www.ai.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed 6 Apr 2017.
29. Rojek M. Federated learning for IoT. Medium, becoming human: artificial intelligence magazine. http://www.becominghuman.ai/theres-a-better-way-of-doing-ai-in-The-iot-era-feabbbc1b589. Accessed 16 Apr 2019.

30. Porter E. What is a botnet? And how to protect yourself in 2020. SafetyDetectives, Safety Detectives. http://www.safetydetectives.com/blog/what-is-a-botnet-and-how-to-protect-yourself-in/#review-2. Accessed 28 Dec 2019.
31. Hendrickson J. What is the mirai botnet, and how can i protect my devices? How to geek, LifeSavvy media. http://www.howtogeek.com/408036/what-is-the-mirai-botnet-and-how-can-i-protect-my-devices/. Accessed 22 Mar 2019.
32. Understanding denial of service attacks. Cybersecurity and infrastructure security agency CISA. http://www.us-cert.cisa.gov/ncas/tips/ST04-015. Accessed 20 Nov 2019.
33. Moisejevs I. Poisoning attacks on machine learning. Towards data science, medium. http://www.towardsdatascience.com/poisoning-attacks-on-machine-learning-1ff247c254db. Accessed 15 July 2019.
34. Fang M et al. Local model poisoning attacks to Byzantine-Robust federated learning. In: Usenix security symposium. arXiv:1911.11815. Accessed 6 Apr 2020.