Chapter 4

Book Chapter

Enhancing IOT Security: Leveraging Artificial Intelligence

Chithra ES [*1], Arathi PH [†2], Pranitha P [‡3], and Geetha R [§4]

[1]Assistant Professor, Department of Computer Application, Acharya Institute of Graduate Studies, Bangalore
[2]Assistant Professor, Department of Computer Application, Acharya Institute of Graduate Studies, Bangalore
[3]Assistant Professor, Department of Computer Application, Acharya Institute of Graduate Studies, Bangalore
[4]Assistant Professor, Department of Computer Application, Acharya Institute of Graduate Studies, Bangalore

Abstract

In recent years, the adoption of the Internet of Things (IoT) has experienced a rapid surge, accompanied by a corresponding rise in cybersecurity concerns. At the forefront of cybersecurity advancements lies Artificial Intelligence (AI), utilized for crafting sophisticated algorithms aimed at fortifying networks and systems, including those within the IoT realm. Nonetheless, cyber adversaries have identified methods to exploit AI, going as far as employing adversarial AI techniques to orchestrate cybersecurity breaches. This review paper consolidates insights from numerous surveys and scholarly works pertaining to IoT, AI, and AI-driven attacks, delving into the intricate interplay among these domains. The primary

[*]Email: chithra.es@gmail.com Corresponding Author
[†]Email: arathyhariharan@gmail.com
[‡]Email: pranitha2210@gmail.com
[§]Email: geethar63504@gmail.com

aim is to comprehensively synthesize and summarize pertinent literature in these areas, shedding light on the evolving landscape of IoT, AI, and cybersecurity, both in terms of defensive strategies and offensive tactics employed by malicious actors.

Keywords: Internet of Things (IoT). Cybersecurity. Artificial Intelligence (AI). Algorithms. Networks. Systems Adversarial AI.

1 Introduction

The internet of things, or IoT, is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. IoT devices are typically embedded with technology such as sensors and software and can include mechanical and digital machines and consumer objects.(Kiran, 2019). Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, deliver enhanced customer service, improve decision-making, and increase the value of the business. With IoT, data is transferable over a network without requiring human-to-human or human-to-computer interactions. A thing in the Internet of Things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low, or any other natural or man-made object that can be assigned an Internet Protocol address and can transfer data over a network. The goal is to link commonplace physical objects into a networked ecosystem of digital data that is reachable from anywhere at any time. With sensing, processing, and actuation built in, ”things” in the Internet of Things function autonomously to provide intelligent and cutting-edge services. (Mukhtar et al., 2023).

An IoT ecosystem consists of web-enabled smart devices that use embedded systems – such as processors, sensors and communication hardware – to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway, which acts as a central hub where IoT devices can send data. Before the data is shared, it can also be sent to an edge device where that data is analyzed locally. Analyzing data locally reduces the volume of data sent to the cloud, which minimizes bandwidth consumption. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices – for example, to set them up, give them instructions or access the data. The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.(see figure 1).

IoT helps people live and work smarter. Consumers, for example, can use IoT-embedded devices -such as cars, smartwatches or thermostats - to improve their lives.

DOI:10.48001/978-81-966500-6-3-4 | ISBN: 978-81-966500-6-3 | Copyright ©2024 QTanalytics®

For example, when a person arrives home, their car could communicate with the garage to open the door; their thermostat could adjust to a preset temperature; and their lighting could be set to a lower intensity and color. In addition to offering smart devices to automate homes, IoT is essential to business. It provides organizations with a real-time look into how their systems really work, delivering insights into everything from the performance of machines to supply chain and logistics operations. IoT enables machines to complete tedious tasks without human intervention. Companies can automate processes, reduce labor costs, cut down on waste and improve service delivery. IoT helps make it less expensive to manufacture and deliver goods, and offers transparency into customer transactions.

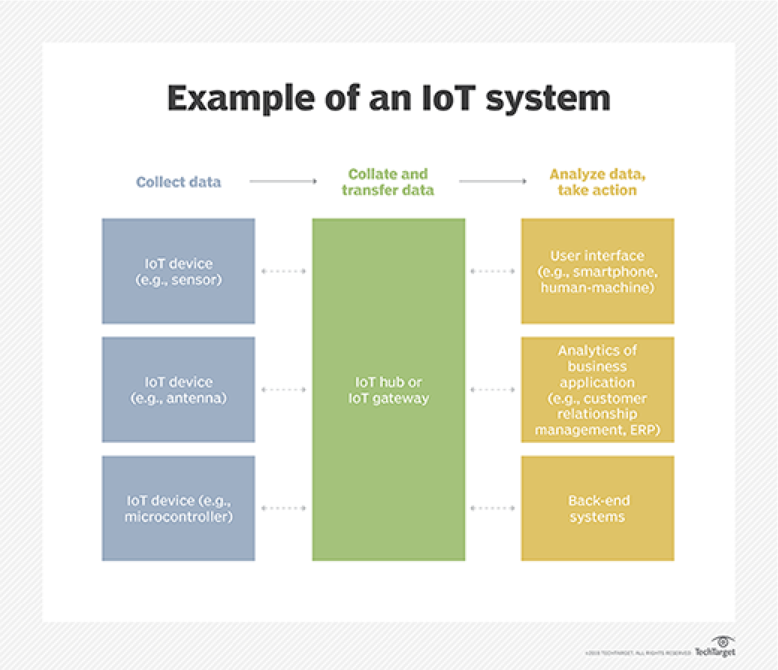
Example of an IoT system
Collect data
Collate and transfer data
Analyze data, take action
IoT device (e.g., sensor)
IoT device (e.g., antenna)
IoT device (e.g., microcontroller)
IoT hub or IoT gateway
User interface (e.g., smartphone, human-machine)
Analytics of business application (e.g., customer relationship management, ERP)
Back-end systems
TechTarget

Figure 1. Example of an IOT system

2 Literature review

2.1 Methods of attacking IoT devices

Attacking IoT devices involves various methods and techniques, each targeting different vulnerabilities within the device or its ecosystem. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks aim to overwhelm IoT devices or networks with excessive traffic, rendering them unresponsive. Man-in-the-middle (MitM) attacks intercept communication between IoT devices or between a device and a server, allowing attackers to eavesdrop on sensitive data, modify messages, or inject malicious content. (Džaferović et al., 2019). Physical attacks involve gaining access to an IoT device to tamper with its hardware or extract sensitive information.

Exploiting default credentials is a common attack method, as many IoT devices come with weak, well-known usernames and passwords. Firmware attacks target vulnerabilities in the device's firmware, allowing attackers to inject malicious updates or tamper with the firmware for persistent control.(Mukhtar et al., 2023; Sasi et al., 2023). Eavesdropping and sniffing involve unauthorized listening to communications and capturing data packets to extract sensitive information. IoT devices often run software with vulnerabilities that can be exploited through methods like buffer overflow attacks, SQL injection, and remote code execution exploits.(Noman & Abu-Sharkh, 2023).

2.2 Bluetooth man in the middle

A common MITM attack on IoT devices occurs over a Bluetooth connection. Many IoT devices use Bluetooth Low Energy (BLE), which is designed to make IoT devices smaller, cheaper, and more energy efficient. However, BLE is vulnerable to MITM attacks. BLE uses AES-CCM encryption. Although AES encryption is considered secure, the method of exchanging encryption keys is often insecure.(Melamed, 2018). The level of security depends on the pairing method used to exchange temporary keys between devices. BLE uses a three-stage pairing process. 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. (Cäsar et al., 2022).

This temporary key is used to generate short-term keys (some newer devices use long-term keys exchanged using elliptic curve Diffie-Hellman public key cryptography, which is much more secure than the standard BLE protocol). Third, the created keys are exchanged over a secure connection and can be used to encrypt data. Figure shows this three-stage pairing process.(see figure 2).

DOI:10.48001/978-81-966500-6-3-4 | ISBN: 978-81-966500-6-3 | Copyright ©2024 QTanalytics®

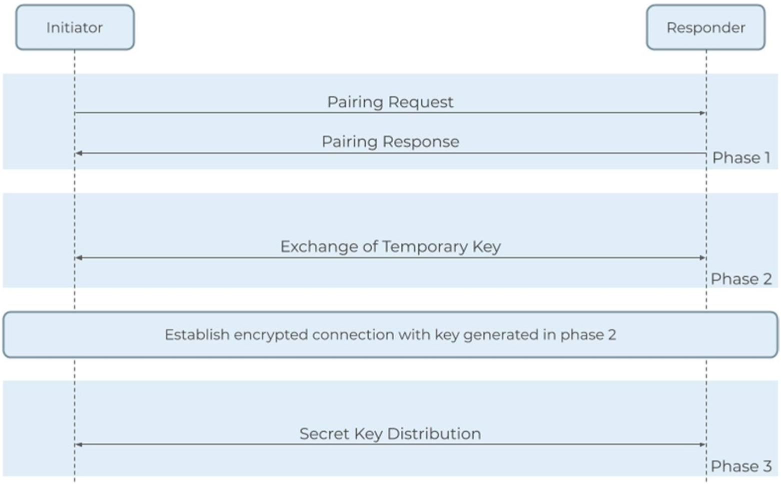

Figure 2. A diagram illustrating the basic BLE pairing process

The temporary key is determined according to the pairing method set at the device's operating system level. There are three common pairing methods for IoT devices. One of them, called Just Works, always sets the temporary key to 0, which is obviously very insecure. However, it remains one of the most popular, if not the most common, ways to pair BLE devices. The second method, passkey, uses a six-digit combination that the user must manually enter into the device. This is fairly secure, but there are ways to circumvent it. Finally, the out-of-band pairing method exchanges ephemeral keys using methods such as near-field communication. The security level of this method depends on the security capabilities of the exchange method. If the exchange channel is protected against MITM attacks, the BLE connection is considered protected as well.

Unfortunately, out-of-band methods are not yet widely used in IoT devices. Another important feature of BLE devices is the Generic Attribute Profile (GATT), which is used for device-to-device communication using a standardized data schema. GATT describes the device's role, general behavior, and other metadata. Any BLE-enabled app within range of the IoT device can read the GATT schema, which provides the app with the necessary information. For an attacker to perform an MITM attack on a BLE network, the attacker must use two connected BLE devices himself: one device acting as an IoT device that connects to the target mobile app, and the other is a fake mobile app that connects to the target IoT device. There are other tools for BLE MITM attacks, such as GATTacker.

DOI:10.48001/978-81-966500-6-3-4 | ISBN: 978-81-966500-6-3 | Copyright ©2024 QTanalytics®

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 .(Kuzlu, Fair, & Guler, 2021).

2.3 False data injection attacks

The financial sector has seen several financial frauds recently, which has caused professionals and auditors responsible for maintaining accuracy and openness in day-to-day auditing operations to become concerned.(Mehta et al., 2022; Mittal, Kaur, & Gupta, 2021). One such attack leading to fraud is the man-in-the-middle attack. The man-in-the-middle attack occurs when an outsider listens in on a conversation between two persons they may trust, takes sensitive data (passwords, PINs, etc.), and uses it improperly.(Sivasankari & Kamalakkannan, 2022). Once an attacker has gained access to some or all devices on an IoT network through a MITM attack, a possible next step could be, for example, a False Data Injection (FDI) attack. In an FDI attack, the attacker slightly alters the measurements of IoT sensors to output false data in order to avoid suspicion . FDI attacks can be carried out in a variety of ways, but in practice, MITM attacks are the most realistic. FDI attacks are often used against sensors that send data to algorithms that try to make predictions based on the data received or use the data to draw conclusions. These algorithms, also known as predictive maintenance systems, are often used to monitor the condition of machines and predict when maintenance or optimization will be required .

In a study (Aboelwafa et al., 2020) introduce a unique Autoencoder (AE)-based FDI attack detection technique. We take advantage of the temporal and spatial correlation of sensor data, which can be used to detect manipulated data.

Similar predictive maintenance algorithms will also become an integral part of smart cities, where an FDI attack could have devastating consequences. An example of an FDI attack against a predictive maintenance system would be the sensors on an aircraft engine that predict when the engine will need critical maintenance. If an attacker could gain access to even a small portion of the sensors, they could generate small amounts of noise that would go undetected by flawed data detection mechanisms, but would be enough to skew the algorithm’s predictions. Indeed, in testing, this could be enough to delay critical maintenance of the system, potentially causing catastrophic failures during operation, resulting in costly unplanned delays and 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. (Kuzlu, Fair, & Guler, 2021). 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. 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.(Alahmadi et al., 2023). 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 has been illustrated.(see figure 3).

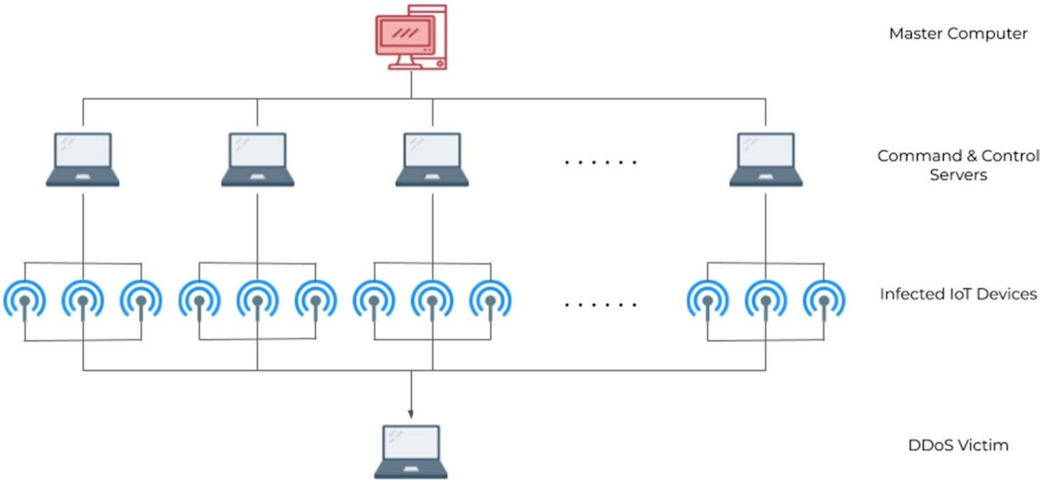

Figure 3. A graphical representation of a common botnet hierarchy

The Mirai worm, one of the most famous malware, 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.(Hallman et al., 2017). 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 etc., in order to capture as many devices as possible. 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. Today, many well-known IoT-targeting botnet viruses are derived from Mirai's source code, including Okiru, Satori, and Reaper.

IoT devices often perform DoS attacks, but are also vulnerable to them themselves. IoT devices are particularly vulnerable to Persistent Denial of Service (PDoS) attacks that render a device or system completely inoperable. This can occur by overloading the battery or power system, or more commonly, by a firmware attack. In a firmware attack, an attacker can exploit a vulnerability to replace the device's underlying software (usually the operating system) with a corrupted or flawed version of that software, thereby rendering the device unusable . When this process is performed legally, it is called flashing; when it is performed illegally, it is called ”flashing.” If a device falls victim to a flashing attack, its owner has no choice but to transfer a clean copy of the operating system and any content that may be stored on the device to the device. In particularly powerful attacks, the corrupted software may overload the device's hardware, making recovery impossible unless parts of the device are replaced . Attacks against the device's power grid are less common, but can be more destructive. An example of this type of attack is a malware-loaded USB device. When the USB device is plugged into a computer, it overloads the device's power supply to the extent that the device's hardware is completely damaged and must be replaced. An example of PDoS malware is BrickerBot. BrickerBot uses a brute force dictionary attack to gain access to IoT devices and, after logging into the device, executes a series of commands to permanently damage the device.

These commands include misconfiguring the device's memory and kernel parameters, disrupting the Internet connection, disrupting the device's performance, deleting all files on the device, etc. . This attack is so destructive that it often requires the reinstallation of the hardware or a complete replacement of the device. If the hardware survives the attack, the software is not reliably flashed and will have to be re-flashed, losing anything that may have been there. Interestingly, BrickerBot is designed to attack the same devices that the Mirai botnet attacks and uses as bots, and it uses the same or similar dictionaries for its brute force attacks. After all, BrickerBot was actually intended to disable devices that Mirai may have employed to protect itself from botnets. Although the structure of IoT systems presents multiple attack surfaces, the most common way to attack IoT systems is through their connections, as connections are usually the weakest link. Going forward, IoT developers are encouraged to ensure that their products are strongly protected against such attacks, and adopting IoT security standards can prevent users from unknowingly purchasing insecure products. Alternatively, securing the network on which IoT systems reside can help prevent many common attacks and isolate the system from the majority.

Using other critical systems or taking backup measures can help mitigate the damage caused if an attack does occur.

3 Artificial Intelligence in Cybersecurity

To dynamically protect systems from cyber threats, many cybersecurity experts are turning to artificial intelligence (AI). AI is most commonly used to detect attacks in cybersecurity by analyzing traffic patterns and looking for activity characteristic of attacks.(Kaur, Gabrijelčič, & Klobučar, 2023).

3.1 Machine Learning

There are two main types of machine learning: supervised and unsupervised. In supervised learning, you manually label training data as malicious or legitimate and feed that data into an algorithm to build a model with ”classes” of data to compare to the traffic you are analyzing. In unsupervised learning, the training data and manual labeling are omitted. Instead, the algorithm groups similar data into classes and classifies them according to the consistency of data within classes and the modularity of data between classes . A common machine learning algorithm in cybersecurity is Naive Bayes, which attempts to classify data based on Bayes' theorem, which assumes that all anomalous activity comes from independent events rather than a single attack. Naïve Bayes is a supervised learning algorithm that, once trained and generating classes, analyzes all activity to determine the probability that it is anomalous. 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 training data samples. It uses iterative refinement to find a description (often simply ”attack” or ”normal”) that best classifies the traffic being analyzed. One example of this approach in cybersecurity is detecting DoS attacks by analyzing the traffic flow rate, size, and duration. It is a popular data mining technique for creating classification schemes based on several covariates or creating prediction algorithms for a target variable.(Song & Lu, 2015). For example, if the flow rate is low but the traffic duration is long, it is likely an attack and is therefore classified as an attack. Decision trees can also be used to detect instruction injection attacks in robotic vehicles by classifying values from CPU consumption, network flows, and the amount of data written.(see figure 4). This technique is popular because it is intuitive, since developers know what the AI considers anomalous traffic and what it does not. Once an effective set of rules is found, the AI analyzes the traffic in real time and issues an alert almost immediately if anomalous activity is detected.

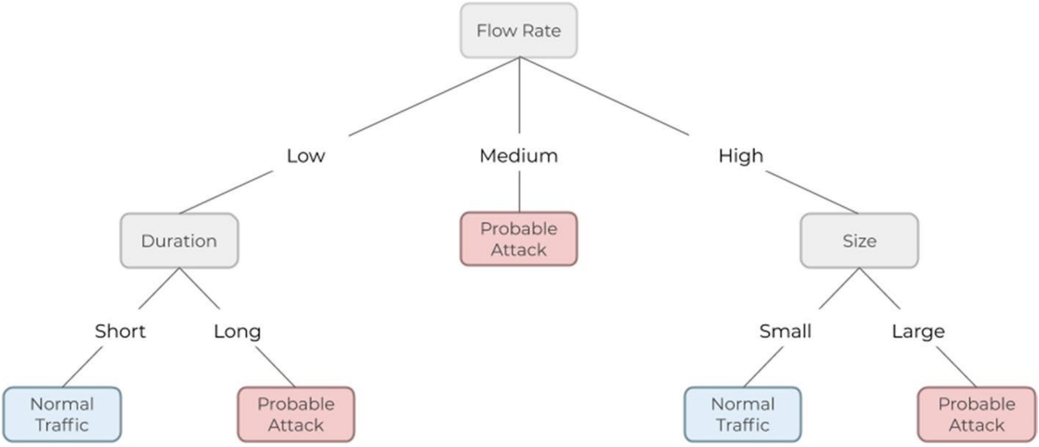

Figure 4. An example of a decision tree for classifying network traffic

Another approach to decision trees is the rule learning technique, which searches for a set of attack features at each iteration while maximizing a certain value that indicates the quality of the classification (i.e., the number of misclassified data samples). The main difference between traditional decision trees and rule learning techniques is that traditional decision trees look for features that lead to a classification, while rule learning techniques find a complete set of rules that can describe a class. This can be an advantage because it allows human advice to be taken into account during rule generation, resulting in an optimized rule set.

3.2 K-Nearest Neighbors

K-Nearest Neighbor (k-NN) techniques learn to create classes from data samples by analyzing the Euclidean distance between new data items and already classified data items to determine which class the new data item should be classified into. (Cohen & Widdows, 2014). Briefly, For example, if k, the number of nearest neighbors, is 3, the new data item will be classified into class 2, but if k is 9, the new data item will be classified into class 1. K-NN technique is attractive for intrusion detection systems because it can rapidly learn from new traffic patterns to detect never-before-seen attacks, even zero-day attacks. (see figure 5). Cybersecurity experts are also exploring the application of k-NN for real-time detection of cyber attacks. This technique has been used to detect attacks such as false data injection attacks, and works well when data can be represented by a model that can measure its distance from other data, e.g. by H. Gaussian distribution or vectors.

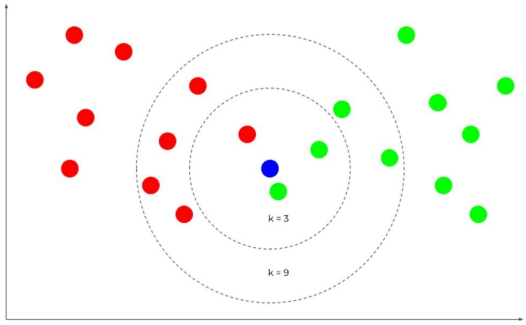

Figure 5. k-NN technique can classify a data point differ- ently given different k values

3.3 Support Vector Machines

Support Vector Machines (SVMs) are an extension of linear regression models that identify a plane that splits data into two classes. This plane can be linear, nonlinear, polynomial, Gaussian, sigmoid, etc., depending on the function used in the algorithm. SVMs can also split data into more than two classes using multiple layers.Many issues can be solved with the SVM, including text categorization, image identification, audio recognition, face detection, faulty card detection, junk mail classification, credit rating analysis, and the classification of diseases like diabetes and cancer.(Montesinos López, Montesinos López, & Crossa, 2022). In cybersecurity, this technology is used to analyze internet traffic patterns and break them down into component classes, such as HTTP, FTP, SMTP, and so on . As SVM is a supervised machine learning technique, it is often used in applica- tions where attacks can be simulated, such as using network traffic generated from penetration testing as training data.

3.4 Artificial Neural Networks

Artificial neural networks (ANNs) are a technology that originates from the way neurons in the brain interact with each other to pass and interpret information. In an ANN, neurons are mathematical formulas that read data and output a target value, which is then passed to the next neuron based on its value. The ANN algorithm is then iterated until the output value is within an acceptable range of the target value, allowing 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 complete, the algorithm presents a mathematical formula that outputs a value that can be used to classify data . The main advantage of ANNs is that they can adapt their mathematical models when presented with new information, while other mathematical models may become obsolete as new types of traffic or attacks become common. This also means that ANNs are better at blocking previously undetected zero-day attacks because they take new information into account than static mathematical models. For this reason, ANNs are robust attack detection systems and have achieved good results against attacks such as DoS.

Currently, the use of AI in cybersecurity is a small but rapidly growing field. It is also expensive and resource intensive, so using AI to protect small systems may not be practical. However, companies with large networks can benefit from these solutions, especially if they are considering introducing or already have IoT devices in their networks. AI cybersecurity is also beneficial for huge systems in smart cities, where AI can provide extremely fast response times that are important in systems such as traffic management. In the future, AI cybersecurity may also be integrated into smaller systems such as self-driving cars and smart homes. In addition, many of the AI cybersecurity measures do not prevent attacks in the first place, but rather detect or thwart attacks in progress, so other preventative security measures must also be taken.

4 AI attacking IoT

Not all AI is used for cybersecurity purposes. Cybercriminals are beginning to use malicious AI to assist in attacks, often by thwarting attack detection algorithms in the case of IoT, or attacking useful AI in such a way that the AI works against its own system.

4.1 Automating Vulnerability Detection

Machine learning can be used to discover vulnerabilities in systems. While this is useful when trying to protect systems by intelligently scanning for vulnerabilities that need to be patched, attackers also use this technology to find and exploit vulnerabilities in target systems. As the use of technology increases, especially less secure technologies such as IoT devices, the number of vulnerabilities attackers can exploit is also exponentially increasing, including zero-day vulnerabilities. To quickly identify vulnerabilities, attackers often use AI to discover and exploit vulnerabilities much faster than developers can fix them. Developers can use these detection tools to 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.

4.2 Fuzzing

Fuzzy sets are used in place of sensitive attribute values in fuzzy approaches. For instance, categories like ”Old” or ”Young” are disclosed rather than the precise age. Likewise, ”High,” ”Medium,” or ”Low” can be used in place of the precise annual income. In this manner, the input data is converted into fuzzy sets while protecting the anonymity of the individual.(Gautam & Mittal, 2022). At its core, fuzzing is a testing method that generates random inputs (numbers, characters, metadata, binary values, especially ”known dangerous” values such as zero, negative, or very large numbers, SQL queries, and special characters) to lead target software. It is divided into dumb fuzzing and smart fuzzing. Dumb fuzzing simply generates errors by randomly changing input variables. This is very fast because it is easy to change the input variables, but it is not very good at finding bugs because of low code coverage. On the other hand, smart fuzzing generates input values that are appropriate for the target software based on the form of the software and the circumstances under which the error occurs.

This software analysis is a great advantage for smart fuzzing, as it allows the fuzzing algorithm to identify where the error may occur. However, developing an efficient smart fuzzing algorithm requires expertise and fine-tuning. Symbolic Execution Symbolic execution is a fuzzing-like technique that looks for vulnerabilities by setting input variables to symbols instead of their actual values. The technique is often divided into offline and online symbolic execution. In offline symbolic execution, only one path at a time is selected for exploration to create new input variables by resolving path predicates. This means that the algorithm must be run from scratch every time a new path is explored, which is a drawback as it incurs a large overhead of re-executing the code. In online symbolic execution, the state is replicated and path predicates are generated at each branch instruction. While this approach does not incur significant overhead, storing all state information and processing all states simultaneously requires a large amount of memory and is resource-intensive.

4.3 Input Attacks

When an attacker modifies the input of an AI system to make it stop working properly or produce erroneous outputs, it is called an input attack. An input attack is performed by adding attack patterns to the input. This could be anything from putting tape on a physical stop sign to confuse a self-driving car to adding small amounts of noise to an image that is imperceptible to the human eye but confuses the AI.

Notably, the actual algorithms and security of the AI do not need to be compromised to perform an input attack. All that needs to be modified is the input whose output the attacker wants to compromise. In the case of tape on a stop sign, the attacker may not need to use any technology. More advanced attacks, however, are completely hidden from the human eye. An attacker may alter small parts of an image in a very precise way in order to mislead an algorithm. However, input attacks are often categorized along two axes: perceptibility and format. The perceptibility of an input attack is a measure of how noticeable the attack is to the human eye, while format is a measure of whether the attack is digital or physical. At one end of the perceptibility axis are perceptible attacks. Modifying a target by distorting it, removing parts, changing its color, etc., or adding elements to the target by sticking physical tape or adding digital markers, etc., are types of perceptible attacks.

Salient attacks are noticeable to humans, but they may not notice small changes, like tape on a stop sign, or consider them important. A stop sign with tape or scratches on it will be recognized as a stop sign by a human driver, but an autonomous car may not. This contributes to the effectiveness of salient attacks, often allowing the attack to be hidden in plain sight. In contrast, imperceptible attacks are invisible to the human eye. These include things like ”digital dust,” a small amount of noise added throughout an image that is invisible to the human eye but significant enough for the AI to alter the output, or imperceptible patterns on a single 3D-printed object that is not visible to the AI. Imperceptible attacks can also occur via voice B by playing audio signals outside the range of human hearing that would be picked up by a microphone. Unobtrusive attacks generally pose a greater security risk, as there is little chance that a human would notice the attack before the AI algorithm returns an inaccurate answer.(see figure 6).

The form of the attack is usually either digital or physical, and many attacks are a combination of both. In many cases of physical attacks, the attack pattern must be obvious rather than unobtrusive, since physical objects must be digitized in order to be processed, and in the process, some finer details may be lost. Some attacks remain difficult to detect. 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.

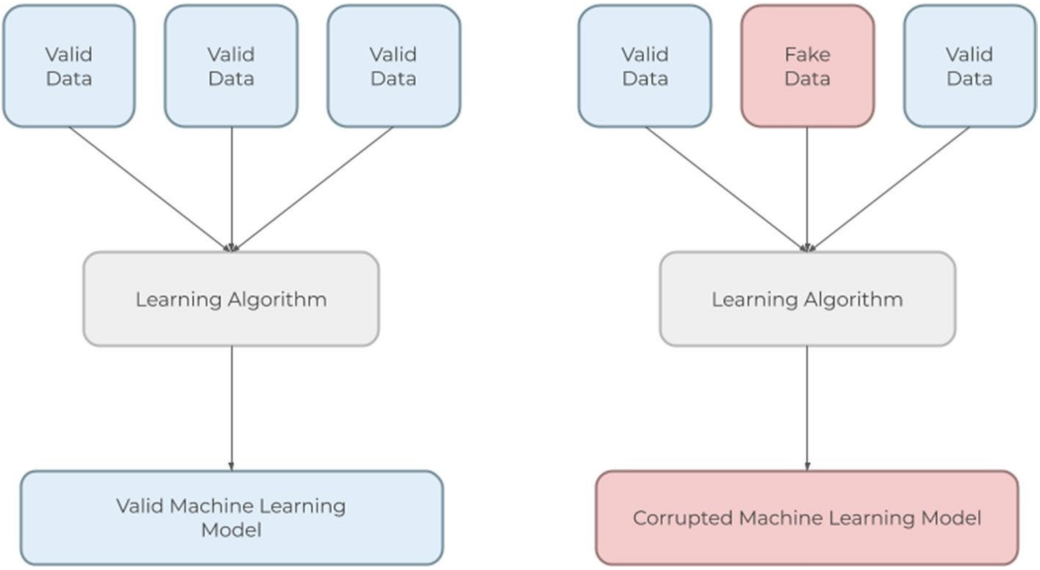

Figure 6. A visual representation of dataset poisoning

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. 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 to train predictive keyboards. 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.(Kuzlu, Fair, & Guler, 2021). 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. 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 can be sped up significantly. However, users could easily manipulate the process with their filters 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.

4.4 Model poisoning

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. 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.

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 Conclusion

The intersection of AI and IoT presents both opportunities and challenges in cybersecurity. AI algorithms hold promise as advanced tools for intrusion detection and real-time threat mitigation in IoT systems. While these technologies are still in development and face implementation challenges, their potential to enhance security is significant. However, the same AI capabilities can be turned against IoT systems by attackers, posing serious threats as these systems expand, especially in complex environments like smart cities.

The discussion underscores the critical need for robust cybersecurity strategies that integrate AI-driven defenses while anticipating AI-based attacks. IoT systems, with their expansive attack surfaces, require continuous vigilance and adaptive defenses to mitigate evolving threats effectively. As technologies evolve, understanding and preemptively addressing vulnerabilities in both AI and IoT frameworks will be crucial to safeguarding against potential exploits.

6 Suggestions

1. Advanced AI Integration: Invest in further research and development to advance AI algorithms tailored for cybersecurity applications in IoT. Focus on enhancing the

accuracy and real-time capabilities of AI-driven intrusion detection systems.

2. Multi-layered Defense Strategies: Implement comprehensive cybersecurity frameworks that incorporate AI for anomaly detection, predictive analytics, and automated response mechanisms. Augment these with traditional security measures to create a resilient defense against sophisticated attacks.

3. Continuous Monitoring and Adaptation: Establish continuous monitoring protocols to detect and respond to emerging threats promptly. Leverage AI to analyze vast amounts of data in real-time, enabling proactive threat mitigation and reducing response times.

4. Education and Awareness: Promote awareness among cybersecurity professionals and stakeholders about the evolving landscape of AI and IoT vulnerabilities. Foster a culture of proactive defense strategies and collaboration to address potential threats effectively.

5. Regulatory and Ethical Considerations: Advocate for regulatory frameworks that address the ethical implications of AI use in cybersecurity, ensuring responsible deployment and minimizing misuse by malicious actors.

6. Collaborative Research Initiatives: Encourage interdisciplinary research initiatives that bring together experts in AI, cybersecurity, and IoT to innovate and develop holistic security solutions. Foster collaboration between academia, industry, and government agencies to stay ahead of emerging threats.

By prioritizing these suggestions, stakeholders can proactively address the complex cybersecurity challenges posed by the convergence of AI and IoT. This proactive approach will be crucial in safeguarding critical infrastructure, personal data, and ensuring the trustworthiness of interconnected IoT ecosystems in the digital age.

References

Aboelwafa, M. M., Seddik, K. G., Eldefrawy, M. H., Gadallah, Y., & Gidlund, M. (2020). A Machine-Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT. IEEE Internet of Things Journal, 7(9), 8462–8471. https://doi.org/10.1109/JIOT.2020.2991693

Alahmadi, A. A., Aljabri, M., Alhaidari, F., Alharthi, D. J., Rayani, G. E., Marghalani, L. A., Alotaibi, O. B., & Bajandouh, S. A. (2023). DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions. Electronics (Switzerland), 12(14). https://doi.org/10.3390/electronics12143103

Cäsar, M., Pawelke, T., Steffan, J., & Terhorst, G. (2022). A survey on Bluetooth Low Energy security and privacy. Computer Networks, 205. https://doi.org/10.1016/j.comnet.2021.108712

Cohen, T., & Widdows, D. (2014). Geometric Representations in Biomedical Informatics: Applications in Automated Text Analysis. Methods in Biomedical Informatics: A Pragmatic Approach, 99–139. https://doi.org/10.1016/B978-0-12-401678-1.00005-1

Džaferović, E., Sokol, A., Almisreb, A. A., & Mohd Norzeli, S. (2019). DoS and DDoS vulnerability of IoT: A review. Sustainable Engineering and Innovation, 1(1), 43–48. https://doi.org/10.37868/sei.v1i1.36

Gautam, S., & Mittal, P. (2022). Systematic Analysis of Predictive Modeling Methods in Stock Markets. International Research Journal of Computer Science, 9(11), 377–385. https://doi.org/10.26562/irjcs.2022.v0911.01

Hallman, R., Bryan, J., Palavicini, G., Divita, J., & Romero-Mariona, J. (2017). IoDDoS -The internet of distributed denial of sevice attacks A case study of the mirai malware and IoT-Based botnets. IoTBDS 2017 - Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security, 47–58. https://doi.org/10.5220/0006246600470058

Kaur, R., Gabrijelčič, D., & Klobučar, T. (2023). Artificial intelligence for cybersecurity: Literature review and future research directions. Information Fusion, 97. https://doi.org/10.1016/j.inffus.2023.101804

Kiran. (2019). Internet of Things. In D. R. Kiran (Ed.), Production planning and control (pp. 495–513). Butterworth-Heinemann. https://doi.org/10.1016/B978-0-12-818364-9.00035-4

Kuzlu, M., Fair, C., & Guler, O. (2021). Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity. Discover Internet of Things. https://doi.org/10.1007/s43926-020-00001-4

Mehta, K., Mittal, P., Gupta, P. K., & Tandon, J. K. (2022). Analyzing the Impact of Forensic Accounting in the Detection of Financial Fraud: The Mediating Role of Artificial Intelligence. Advances in Intelligent Systems and Computing, 585–592. https://doi.org/10.1007/978-981-16-2597-8_50

Melamed, T. (2018). An active man-in-The-middle attack on bluetooth smart devices. International Journal of Safety and Security Engineering, 8(2), 200–211. https://doi.org/10.2495/SAFE-V8-N2-200-211

Mittal, P., Kaur, A., & Gupta, P. K. (2021). THE MEDIATING ROLE of BIG DATA to INFLUENCE PRACTITIONERS to USE FORENSIC ACCOUNTING for FRAUD DETECTION. European Journal of Business Science and Technology, 7(1), 47–58. https://doi.org/10.11118/ejobsat.2021.009

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Support Vector Machines and Support Vector Regression. In Multivariate statistical machine learning methods for genomic prediction (pp. 337–378). Springer, Cham. https://doi.org/10.1007/978-3-030-89010-0_9

Mukhtar, B. I., Elsayed, M. S., Jurcut, A. D., & Azer, M. A. (2023). IoT Vulnerabilities and Attacks: SILEX Malware Case Study. Symmetry, 15(11). https://doi.org/10.3390/sym15111978

Noman, H. A., & Abu-Sharkh, O. M. (2023). Code Injection Attacks in Wireless-Based Internet of Things (IoT): A Comprehensive Review and Practical Implementations. Sensors, 23(13). https://doi.org/10.3390/s23136067

Sasi, T., Lashkari, A. H., Lu, R., Xiong, P., & Iqbal, S. (2023). A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges. Journal of Information and Intelligence. https://doi.org/10.1016/j.jiixd.2023.12.001

Sivasankari, N., & Kamalakkannan, S. (2022). Detection and prevention of man-in-the-middle attack in iot network using regression modeling. Advances in Engineering Software, 169. https://doi.org/10.1016/j.advengsoft.2022.103126

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. Shanghai Archives of Psychiatry, 27(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044