Article

Not peer-reviewed version

Comprehensive Study of IoT Vulnerabilities and Countermeasures

Ian Coston *, Eadan Plotnizky *, Mehrdad Nojoumian *

Posted Date: 15 October 2024

doi: 10.20944/preprints202410.1215.v1

Keywords: Internet of Things; Internet of Battlefield Things; Internet of Medical Things; Internet of Agricultural Things; Artificial Intelligence; Machine Learning; Deep Learning; AI-Driven Security; Cloud Computing; Cloud Security; Cybersecurity; IoT Security; IoT Vulnerabilities; Hardware Security; Software Security; Network Security; Wireless Sensor Networks; Wireless Communication Protocols; IoT Protocols; Intrusion Detection System; Vulnerability Analysis; Attack Mitigation; Penetration Testing

Article

Comprehensive Study of IoT Vulnerabilities and Countermeasures

Ian Coston, Eadan Plotnizky and Mehrdad Nojoumian *

Department of Electrical Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA; ICoston2016@fau.edu (I.C.); EPlotnizky2020@fau.edu (E.P.); MNojoumian@fau.edu (M.N.)

* Correspondence: ICoston2016@fau.edu; Tel.: +1-561-297-3411 (I.C.)

Abstract: This comprehensive study provides an in-depth examination of the 'Internet of' technologies, focusing specifically on the Internet of Things (IoT), which is defined as the networked interconnection of multiple devices through various wireless protocols that facilitate data transfer and improve operational intelligence. The applications of IoT are widespread, including urban infrastructure, domestic settings, transportation systems, military operations, and agricultural practices. The study elucidates the complexities of cloud computing and artificial intelligence (AI), systematically categorizing the vulnerabilities inherent in hardware, software, cloud, network, and sensor networks, and underscores the omnipresent security risks in networks and IoT devices, highlighting the need for robust mitigation strategies. The proposed trajectory of this study is the development of a comprehensive AI architecture that can discern and counteract a wide spectrum of vulnerabilities. This AI system, embedded in a Nvidia Jetson Orin Nano that is attached to IoT devices, will be supported by infrastructure hosted within a cloud environment, will continuously monitor the IoT device for anomalies, conduct self-initiated penetration tests to pinpoint weaknesses, and implement appropriate countermeasures to mitigate the identified vulnerabilities. The study also advocates the need for the exploration of postquantum cryptographic solutions in further research in this field to safeguard data on IoT devices, a proactive approach that is crucial in light of the potential vulnerability of contemporary cryptography to quantum computing breakthroughs.

Keywords: Internet of Things; Internet of Battlefield Things; Internet of Medical Things; Internet of Agricultural Things; Artificial Intelligence; Machine Learning; Deep Learning; AI-Driven Security; Cloud Computing; Cloud Security; Cybersecurity; IoT Security; IoT Vulnerabilities; Hardware Security; Software Security; Network Security; Wireless Sensor Networks; Wireless Communication Protocols; IoT Protocols; Intrusion Detection System; Vulnerability Analysis; Attack Mitigation; Penetration Testing

1. Introduction

The advent of the Internet of Things (IoT) has revolutionized the landscape of technology by enabling the interconnection of billions of devices. From smart homes and medical systems to battlefield and agricultural applications, IoT plays a crucial role in modern infrastructures, facilitating automation, data exchange, and intelligent decision-making. Despite its transformative impact, the rapid growth of IoT has introduced significant security challenges. IoT devices, by their very nature, are highly susceptible to vulnerabilities across various layers, including hardware, software, network, and cloud infrastructures. These vulnerabilities pose a substantial risk to the confidentiality, integrity, and availability of IoT systems. This study aims to provide a comprehensive examination of the security issues inherent in IoT environments, systematically categorizing vulnerabilities in hardware, software, network, and sensor networks, and emphasizing the need for robust mitigation strategies. As IoT devices become more ubiquitous, the ability to secure them from a wide range of threats is critical for maintaining trust in these technologies, particularly in sensitive applications such as the Internet of Medical Things (IoMT), the Internet of Battlefield Things (IoBT) and the Internet of

 doi:10.20944/preprints202410.1215.v1

Agricultural Things (IoAT). A key contribution of this research is the development of an AI-driven architecture designed to identify and mitigate IoT vulnerabilities. Using artificial intelligence, this architecture aims to detect anomalies in IoT systems and autonomously respond to potential threats. The AI system, embedded in an Nvidia Jetson Orin Nano and integrated within a cloud-hosted infrastructure, will continuously monitor IoT devices, performing self-initiated penetration tests to classify vulnerabilities and implement appropriate countermeasures. In addition to addressing present security challenges, this study recognizes the looming threat posed by advancements in quantum computing. As quantum capabilities evolve, traditional cryptographic systems may become increasingly vulnerable to attacks. Thus, this research advocates for the early adoption of postquantum cryptographic techniques, which will be crucial for protecting IoT data from future quantum-enabled threats. This dual approach -integrating AI for real-time threat detection and mitigation, coupled with quantum-resistant cryptography - provides a forward-looking framework to secure IoT environments against both current and emerging risks. Through this study, we seek to advance the security of IoT systems by integrating AI and exploring cryptographic solutions that future-proof these technologies against emerging threats.

2. Preliminary Information

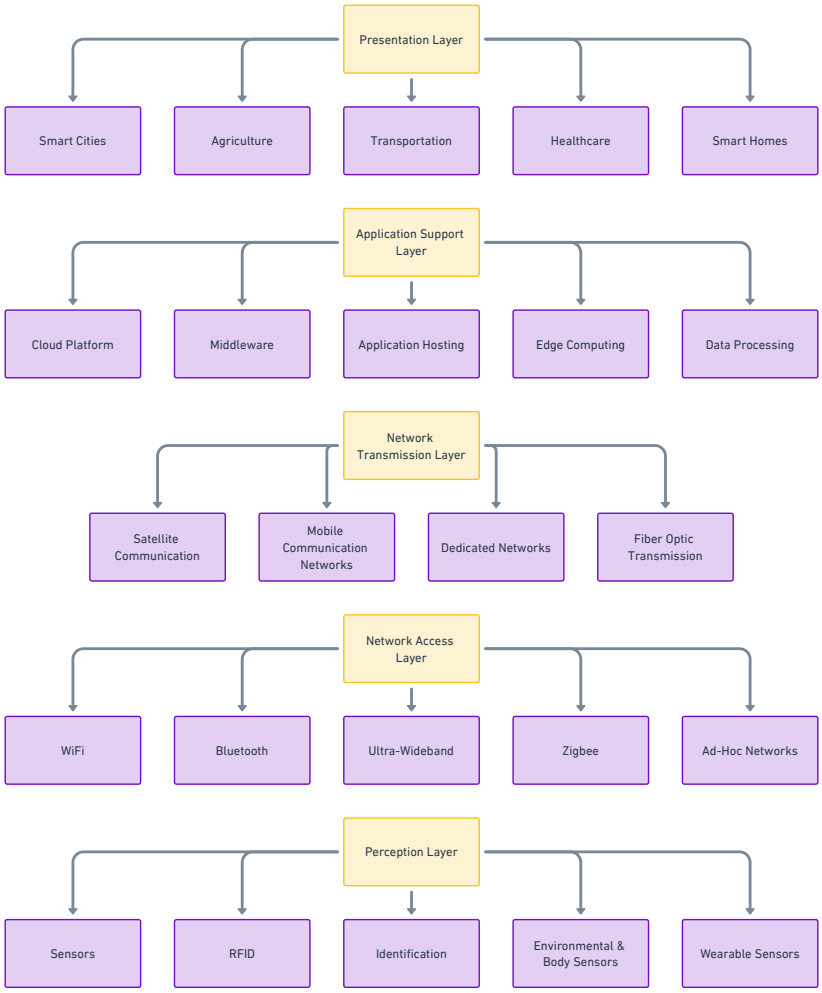

Figure 1. Five Layer Architecture. The image has been inspired by [1].

The 'Internet of' series of technologies has been an idea that has been developing over time for many different avenues and purposes. In today's society, most of the new evolution of technologies are typically part of this thought process, whether its Internet of People (Human-Machine Interaction),

 doi:10.20944/preprints202410.1215.v1

Internet of Agents (Machine Learning and Artificial Intelligence), Internet of Content (Cloud) or Internet of Things (Machine-to-Machine Interaction) [2]. While the authors in [2] looked at the overall 'Internet of' series, the one we will focus on will be the Internet of Things. The idea of Internet of Things is essentially when multiple devices, or otherwise known as "Things," are connected together in some manner through the use of a wireless network of some sort. This network can be any wireless protocol such as Bluetooth, WiFi, 5G, Ultra-wideband, radio frequency identification, and many more. Now, all these devices communicate with each other, send data to each other, and are typically used to make things smarter [2–4]. Some of the uses of IoT are seen in technology found in cities, homes, transportation, military, agriculture, and so many more. The idea is to make life easier for people who use technology by making the technology smarter. However, all IoT use cases generally fall within the general multilayer architecture ranging from three to six layers, with the four-layer model being the most popular. The five-layer architecture, as shown in Figure 1, is one that we will focus on in this investigation, as it goes a bit deeper compared to the four-layer model[1].

The five-layer architecture consists of the perception layer, which is the layer that gathers the data. This can consist of sensors and other things that are used to collect information, such as barcodes [1]. We then move up a layer to the network access layer, which is essentially the means of communication. This is how the device sends out signals and communicates with other devices. This can be seen as wireless protocols such as ultra-wideband, WiFi, Bluetooth, etc. and helps to create the sensor network [1]. We then have the 'Network Transmission Layer' above that, which is how the network communicates with each other as a whole. This can be seen as communication using satellites, mobile communication networks, or even a specific dedicated network [1]. The next layer would be the Application Support Layer which consists of middleware, cloud platforms, application host, etc. [1]. And finally, we have the "Presentation Layer" which is what IoT devices are used solely for whether it is used for a smart city, agriculture, transportation, and many more [1]. In other words, the presentation layer is the output layer for IoT devices where they act upon the world or environment.

2.1. Internet of Things for Smart Cities & Transportation

As the Internet of Things devices become more prevalent and advanced, they are slowly being added to cities and vehicles in various aspects to allow for a smoother city or vehicular experience overall. IoT is used everywhere within cities, from security to maintenance, which facilitates operating, as well as even vehicle mobility [5]. Some of the most common uses of the Internet of Things in cities can be seen in Figure 2. As we can see, there are many avenues for IoT within smart cities that are all vary different and specific in their overall purpose.

 doi:10.20944/preprints202410.1215.v1

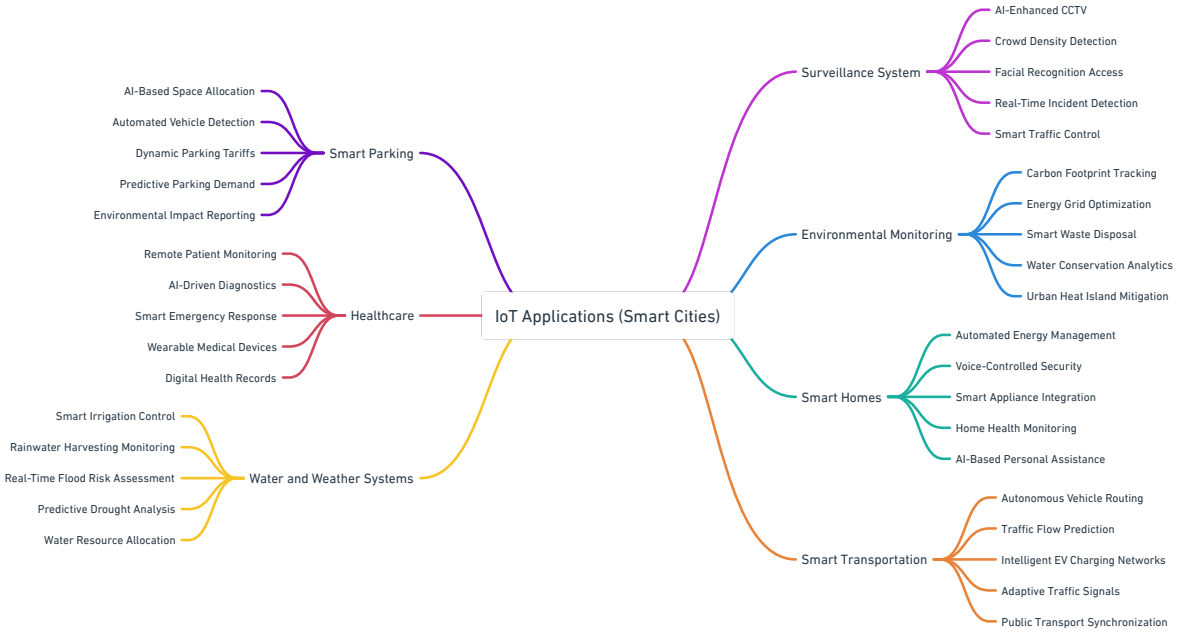

Figure 2. IoT Applications within Smart Cities. The image has been inspired by [6].

For example, you may be driving through the city in a Tesla vehicle that communicates with the surrounding smart vehicles and with the traffic lights to ensure that your car and everyone else have a smooth driving experience. You may see cameras around the city that are all connected to a CCTV system that is used to prevent crime from occurring in the city. As you drive, you see a store that you want to go shopping in. You park in a parking lot and use a smart meter to monitor the time you stay there and to ticket you if you stay too long without paying. When you walk into the store, you see cashierless registers and cameras everywhere that track the placement of items and objects found in the store. This is the experience that a smart city wants citizens to have. All of these are used to make the city run much more smoothly and to drastically improve the lives of its citizens. With all this in mind, IoT is also being used more in-depth in transportation [5]. And, in particular, we are seeing IoT being added to transportation through the use of vehicle-to-vehicle (V2V) communication protocols as well as vehicle-to-infrastructure (V2I) communication protocols. These protocols essentially allow each vehicle with IoT sensors to communicate with each other but also with the surrounding buildings within the city. Electric scooters commonly seen around cities, as well as bus systems, are all controlled by IoT devices. All of this work together with the city to prevent citizens from being stuck in a specific location in the city and allow for easier mobility. However, another way in which they are often used, is with smart meters and self-driving capabilities. Meters use V2I technology to communicate with the vehicle to get an accurate time on how long the vehicle is placed within the spot. This allows everything to be more accurate when it comes to meter tracking. And for the self-driving cars that are on the roads, they are often using IoT technology to communicate with other cars and the infrastructure to know things such as the speed limit, the color of lights, street signs, monitor other cars, people and objects, and so much more. Once again, the purpose is to essentially make life easier for the user.

2.2. Internet of Things Smart Home System

Internet of Things devices are becoming a major presence in the home environment, and the sole purpose is to make life easier for the user and people living within that environment. It first started as a small gadget or 'thing' that could be added to a home ecosystem and connected to other 'things' with wireless signals, but as times progress, more and more native home devices come with the 'internet of things' factor already built in [7]. The home IoT ecosystem is varied and not the same everywhere, but one of the ways it can be seen is as follows. The user may push the button on their car where sends a

 doi:10.20944/preprints202410.1215.v1

signal to the garage to open. From there, they tell an Alexa device that they are home, and then Alexa would follow a routine that the user set within the app. The routine would turn on all of the house lights, set the AC thermostat to 68 degrees Fahrenheit, set the air purifiers to a medium setting, turn on the living room TV to their favorite channel, etc. And then, when the user gets ready for bed, he tells Alexa 'good night.' Alexa would then close all window blinds, set the air purifiers to high, turn off all lights, set the AC to a lower temperature, and activate the robot vacuum for night cleaning. This is just a sample of what home automation would look like. Everyone sets it differently and has a different use, but the overall idea of it is that it makes someone's life easier.

2.3. Internet of Medical Things

Internet of Things devices find extensive use in the medical field, also known as the Internet of Medical Things (IoMT). IoMT is found everywhere today with wireless medical devices, widely used in hospitals, nursing homes, and homes (for personal health related use). Integrating IoT technology in the medical field has revolutionized healthcare delivery, offering numerous benefits and practical applications for healthcare services and patients. Adopting IoMT protocols offers numerous practical applications for medical device companies, health services, patients, and caregivers.

For example, healthcare providers can collect real-time data on patient health conditions, such as heart rate, blood pressure, temperature, SPO2 (blood oxygen saturation levels), and respiratory rate. This remote monitoring capability allows for spot checks, continuous patient monitoring, and early detection of abnormalities in the patients health. Patients can actively participate in their care using mobile applications that connect to IoMT devices, providing instant access to their health status and facilitating the self-management of chronic diseases, saving valuable time for health services to focus on other tasks. IoMT facilitates the development of innovative solutions, such as the CardiacSense watch based on IoMT [8]. The CardiacSense watch incorporates sensors and interfaces such as pulse rate, ECG recording, continuous detection of A-fib, unlimited event reports, manually added measurement values, general arrhythmia detection, threshold configuration, detailed event report, monthly reports, and sleep time tracking [9].

2.4. Internet of Agricultural Things

We can see implementation everywhere with the ongoing advances of the Internet of Things. One of the hottest implementations of IoT is in agriculture. From basic smart home gardening systems to smart farms, the IoAT offers intelligent solutions to different agricultural applications, such as precision farming, livestock monitoring, greenhouse monitoring, and agricultural drones. Smart farms often use a variety of sensors to collect real-time data on the farms status to monitor both livestock and crops: sensors to monitor oil richness, temperature, humidity, gas, air pressure, water pressure, and crop disease for field monitoring; and temperature, heart rate, and digestion for livestock [10]. One of the many innovations designed and implemented is SeeTree, an "Intelligence Platform for Trees" that allows growers to monitor the productivity and health of individual trees. It can scan and analyze hundreds of millions of trees using rich information sources, including drones, satellites, IoT sensors, climate data, and more [11,12].

2.5. Internet of Battlefield Things

Internet of Things devices are used worldwide, but a place where they are becoming increasingly prevalent is in the battlefield environment. This is called the Internet of Battlefield Things, otherwise known as IoBT for short. These devices have the sole purpose of helping the soldier while on the battlefield, whether that is by monitoring and tracking the soldier and enemies, or whether they are actually using the IoBT device for warfare purposes [13,14]. For example, in a given field environment, the soldier will have a heart rate sensor, a salinity sensor, an optical sensor, and movement tracking, all of which are used to monitor the body and movement of the soldier. In the air, they may have an unmanned aerial vehicle (UAV) that uses video monitoring of the entire environment layout to

 doi:10.20944/preprints202410.1215.v1

track soldiers and enemies. And lastly, the vehicles in the field that the soldiers use will have tracking technology, as well as mine detection, sensors on the weaponry, etc. As can be seen in Figure 3, the Internet of Battlefield Things is an area of IoT that is extremely broad but very specific at the same time. Any IoT technology can be considered an IoBT device as long as it is used in the battlefield environment.

Figure 3. Internet of Battlefield Things Layout. This is a generative AI production.

2.6. *Commonly Used Protocols:*

Along with the evolution of IoT devices, many protocols were developed to satisfy the needs of different IoT applications. There are many different situations and needs that come with IoT devices. Some devices need protocols that are energy efficient and short-range, some need different data rates, and some require long-range and high-speed data transport, depending on the use case. This section will give a brief introduction to some of the more common protocols as well as some of the least common protocols that you can find within IoT devices.

ZigBee

ZigBee is a wireless networking protocol based on the IEEE 802.15.4 technical standard [15]. It has a low data rate, low power consumption, low cost, and encrypted communication using the Advanced Encryption Standard (AES) with a 128-bit key. ZigBee is a great technology for Internet of Things devices that need long battery life and a low data transfer rate. The ZigBee data rate is 250 kbps at 2.4GHz (global), 40 kbps at 915MHz (Americas), and 20 kbps at 868MHz (Europe). Some applications of ZigBee can be found in Smart homes, Smart buildings, the Internet of Medical Things (IoMT), and more [16,17].

Dash7

Dash7 (D7AP) is an open-source subGHz wireless sensor and actuator network protocol (WSAN) that complies with the ISO/IEC 18000-7 standard. It has a medium range of up to 2 Km, low power consumption (multi-year battery life), low latency, and is encrypted using AES with a 128-bit key. D7AP operates in unlicensed ISM bands of 433.92, 868 and 915 MHz [18,19]. A few use cases of D7AP can be found in agriculture IoT, IoMT and smart cities [20].

 doi:10.20944/preprints202410.1215.v1

WiFi

WiFi, or IEEE 802.11, is a standard of wireless LAN technology widely used in business and home environments to obtain fast and reliable Internet access. WiFi provides a common platform to connect various devices, from smart home applications to industrial sensors. The range, reliability, and security strengths of WiFi are ideal for many IoT applications [21]. WiFI encryption has evolved in recent years, and each new generation provides stronger security against attacks. WEP was the original WiFi security protocol, which used the RC4 algorithm with two sides of data communication. However, WEP is easily cracked, and it is no longer considered secure. WPA was an improved version of WEP and addressed some of the security vulnerabilities in WEP. WPA2 is the successor to WPA and is considered to be more secure. WPA2 uses the Advanced Encryption Standard (AES) cipher [22]. WPA3 is the latest WiFi security protocol, offering the strongest security to date. WPA3 uses the Simultaneous Authentication of Equals (SAE) protocol, designed to be more secure than the four-way handshake used in WPA2 [23]. IEEE 802.11 is a living standard and new generations are being developed regularly to meet the growing demands of wireless networking. Some of the most used IEEE 802.11 standards are IEEE 802.11g, which operates in the 2.4 GHz band and supports data rates up to 54 Mbps; IEEE 802.11n, which operates in the 2.4 GHz and 5 GHz bands and supports data rates up to 600 Mbps.; IEEE 802.11ac, which operates in the 5 GHz band and supports data rates up to 6.93 Gbps; IEEE 802.11ax, which operates in the 2.4 GHz and 5 GHz bands and supports data rates up to 9.6 Gbps [24,25].

Cellular

Cellular networks have been around for a while. With new advancements and technologies being developed around them, new cellular networks were discussed, tested, and created to provide the best performances that will match the needs of the latest technology standard. LTE-Advanced(4G) and 5G technologies are the most recent and widely used cellular standards [26]. LTE-A (Long-Term Evolution Advanced) is a wireless cellular technology that significantly improved speed, capacity, and coverage over previous generations of cellular technology. It has a peak downlink data rate of 1 Gbps and an uplink data rate of 500 Mbps, a peak downlink spectrum efficiency of 30 bps / Hz and an uplink spectrum efficiency of 15 bps/Hz, and a bandwidth of 100 MHz [27]. LTE-A is currently being overtaken by 5G [28]. 5G is the fifth generation of wireless cellular technology. It has improvements over previous generations of cellular networks, such as higher data rates, lower quality of service (QoS) latency, low interference, and increased capacity. Some of the 5G requirements include a maximum downlink data rate of 20 Gbps and an uplink data rate of 10 Gbps, a maximum downlink spectrum efficiency of 30 bps/Hz and uplink spectrum efficiency of 15 bps/Hz, user plane latency of 4 ms for eMBB and 1 ms for URLLC, control plane latency of 10-20ms, and bandwidth of 100 MHz - 1 GHz [29]. 5G is the key for advanced IoT applications, such as smart factories, smart hospitals, smart transportation, smart agriculture, smart homes and cities, etc [29,30].

6LoWPAN

6LoWPAN is a networking technology that allows IPv6 packets to be efficiently transmitted over low-power wireless networks, such as those based on the IEEE 802.15.4 standard. It supports various mesh network topologies and can fragment and reassemble packets as needed. 6LoWPAN implementations are small enough to fit 32K flash memory parts. 6LoWPAN enables low-power mesh and sensor networks to take advantage of the benefits of IP networking [31,32]. It has frequency bands of 2.4 GHz, 868 MHz, and 915 MHz (the same as ZigBee) and data rates between 50 and 250 kbit/s [33].

Bluetooth

Bluetooth is a technology standard used for short-range wireless communication between mobile devices. Bluetooth operates on 79 different frequencies to transmit data from 2.402 GHz to 2.48 GHz

 doi:10.20944/preprints202410.1215.v1

and a range up to 100 m (330 ft). The bit rates for Bluetooth are 1 Mbps and 2 Mbps [34]. It is very useful for transmitting small fragments from different IoT sensors [35].

Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a wireless technology that is designed to complement both classic Bluetooth and the lowest power wireless technology possible. It is a distinct technology with different design goals and market segments than classic Bluetooth. BLE transmits data over 40 channels in the 2.4 GHz band (2.402 to 2.48 GHz) [36]. It can be used to create different types of network, from simple point-to-point connections to complex mesh networks. This flexibility makes BLE work with a wide range of applications, including the Internet of Things [37].

LoRa & LoRaWAN

LoRa is an unlicensed band physical layer technology that transmits data transmit signals in the subGHz ISM band. LoRa allows low-data-rate long-range, low-power wireless communication. LoRa has a range of up to 15 km in rural areas and up to 5 Km in urban areas, with data rates of 0.3 kbps - 50 kbps in Europe and 0.9 kbps - 50 kbps in the US [38] [39]. LoRaWAN is an open standard that was developed on top of LoRa. It consists of an end device, gateway, network server, and application server. LoRaWAN sits on the data link layer and provides a complete solution by adding a network layer that includes features such as security, authentication, and data routing [20].

SigFox

Sigfox is a low-power wide area network (LPWAN) technology designed for the Internet of Things. It uses ultra-narrowband technology to transmit data with a very low power consumption over long distances [40]. It operates in the 862 - 928 MHz frequency band and has a channel bandwidth of 100 Hz [41]. With a range of up to 50 km in rural areas and up to 10 km in urban areas, Sigfox can work well with applications that require long-range communication with battery-powered devices. Its data rate ranges from 100 to 600 Bps, depending on the region [29,42].

Narrowband Internet of Things (NB-IoT)

NB-IoT is a low-cost, low-power, and low-data-rate cellular technology built from LTE functions; therefore, it uses the same infrastructure as cellular networks, which makes it a scalable and reliable technology that can be deployed in a variety of locations. It has a range of up to 15 Km in rural areas and 1-5 Km in urban areas, a data rate up to 250 Kbps, and a 200 KHz bandwidth [20,43].

Near Field Communication (NFC)

Near Field Communication (NFC) is a short-range wireless communication protocol used by mobile devices to do all kinds of applications, such as payments, digital keys for homes and cars, and data transferring. NFC provides secure communication between various devices. It has a short range of 4-10 cm, a data rate of 0.02-0.4 Mbps, and it runs on a 13.56 MHz spectrum [44]. NFC is used to enhance different IoT solutions with short-range capabilities [45].

Z-Wave

Z-Wave is a subGHz wireless communication protocol used by different IoT applications. It is an ultra-low-power, mesh network protocol that lets devices communicate with each other over long distances (has a range of 100 meters). Its data rates are 9.6 kbps, 40 kbps, or 100 kbps, and it uses a frequency of 908.42 MHz [46]. Z-Wave deployments can be scaled by linking together Z-Wave networks. Z-Wave is well suited for applications that require reliable, secure, and low-power communication, such as control smart home devices (lights, locks, thermostats, and security systems) [47].

 doi:10.20944/preprints202410.1215.v1

Li-Fi

Li-Fi is a bidirectional short-range wireless technology that uses a visible light communication (VLC) system for data transmission to transfer and receive data. Li-Fi uses overhead LED lighting commonly found in homes as a means of transport and a photo-diode for decoding data. It has a maximum speed of 224 Gbps, which allows a high-definition video to be downloaded in seconds. Because Li-Fi is reliable in light use, it is limited in range since light cannot pass through objects, which makes Li-Fi effective only in closed spaces [48]. Even though its range limitation could be seen as a problem, this limitation provides an additional layer of security by keeping data away from leaking into public spaces, giving malicious actors access to your network [49,50].

Ultra-Wideband (UWB)

Ultra wideband is a short-range, high-bandwidth and energy-efficient wireless communication protocol that can be used for radar imaging, sensor data collection, precise location and tracking. UWB operates at frequencies 3.1 to 10.6 GHz, has a bandwidth of 500 MHz, and a data rate of up to 1 Gbps [51,52]. UWB can be used to accurately measure the distance between two devices. This information can be used for a variety of IoT applications [53].

Advanced Message Queuing Protocol (AMQP)

Advanced Message Queuing Protocol (AMQP) is a reliable and versatile M2M binary protocol. It offers two levels of QoS for the delivery of messages, uses TCP as a transport protocol, and uses TLS/SSL & SASL for security, making it a good fit for high-bandwidth, reliable and secure networks [54]. It supports various messaging patterns, including request/response, publish/subscribe, and transactions (allowing multiple messages to be sent and received as a single unit of work) and topic-based publish-and-subscribe messaging (allows messages to be published to topics so that subscribers can receive messages that are relevant to them) [55].

Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is a lightweight M2M binary protocol with a fixed header of 4 bytes and small message payloads from the IETF CoRE Working Group designed for constrained IoT devices. It supports both request-response and resource-observe architectures and can be used to interoperate with HTTP and the RESTful Web API [56]. CoAP uses UDP as a transport protocol and DTLS for security, making it efficient for use on low-bandwidth and unreliable networks. It is designed to be as lightweight as possible, making it suitable for use on constrained devices with limited resources [54].

Message Queuing Telemetry Transport Protocol (MQTT)

The Message Queuing Telemetry Transport Protocol (MQTT) is a publish/subscribe messaging protocol used for lightweight machine-to-machine (M2M) communications in constrained networks. MQTT uses the Transmission Control Protocol (TCP) as its transport protocol that guarantees the delivery of messages. MQTT also uses TLS/SSL for security, which encrypts messages to protect them from unauthorized access. MQTT supports three levels of QoS, making it more reliable when delivering messages. It uses a small amount of bandwidth and processing power. This makes it ideal for use with small devices that have limited resources. MQTT is also suited for large networks because it can efficiently handle a large number of devices. This is because it uses a publish/subscribe model, which allows devices to receive only messages that are relevant to them [54].

Data Distribution Service (DDS)

Data Distribution Service (DDS) is a machine-to-machine protocol developed by the Object Management Group (OMG) that features decentralized nodes of clients throughout a system (nodes

 doi:10.20944/preprints202410.1215.v1

can identify themselves as subscribers or publishers through a localization server). DDS was created to overcome the disadvantages of centralized publish-subscribe architectures. Provides many quality-of-service parameters that allow users to control the behavior of the DDS system, such as improved scalability, increased reliability, reduced latency, bandwidth, and enhanced security (provides authentication, access control, confidentiality, and integrity to the information distribution) [57,58].

3. IoT Vulnerability Layers

All IoT devices typically show some kind of vulnerability or weakness that dampens its ability to function without issue. These vulnerabilities can typically allow users to gain access to data by utilizing the weakness to get into the device, or they can be used to track and manipulate the device. Either way, this is detrimental to the device and its functionality. Some of the ways that adversaries can get access to this is through network vulnerabilities, software vulnerabilities, or even hardware vulnerabilities, as shown below in Figure 4 [59]. It is important to know these weaknesses so that one can know the best route to actually avoid or mitigate them.

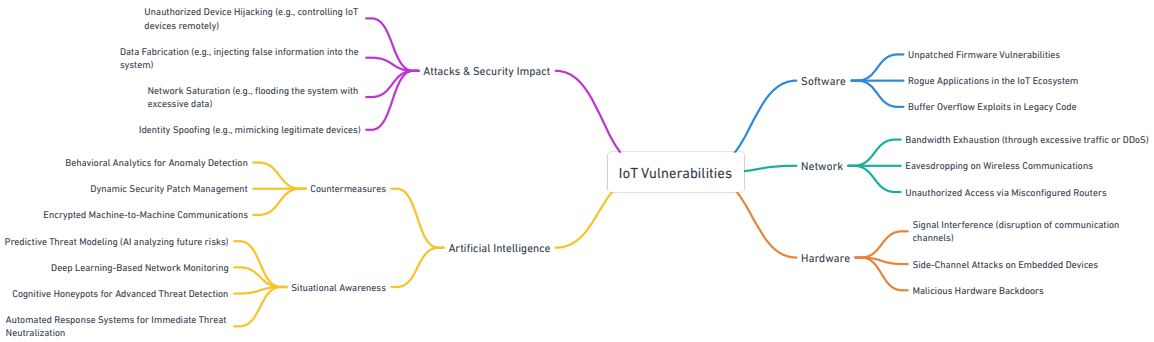

Figure 4. IoT Vulnerability Flow. The image has been inspired by [59]

As we can see in figure 4, these vulnerabilities are just part of the attack layer. How these vulnerabilities are used is the other half of the attack layer. The adversary can quite easily attack any IoT device through any of the vulnerabilities and by doing so, they can either choose to do an attack against Data Integrity, Confidentiality, Authentication, or even Availability. Confidentiality is designed to protect IoT devices and information from unauthorized access and is generally enforced with the use of encryption, access control, and also authentication of user and data [59]. Through any of these vulnerabilities above, we can see information leakage which would cause any kind of data to be releasing to anyone, whether its the adversary or a normal person. Integrity typically would guarantee the protection of unauthorized modifications to the hardware or software of a device by enforcing encryptions, input validations, interface monitoring and restrictions, and so many more [59]. These are designed to keep any portion of the device from being vulnerable. However, once again, during the design phase of a device, typically smaller things are overlooked, which would allow the attacker to use that device for their own malicious purposes. Accountability is the idea of tracking actions and tasks to make sure that it is doing what it is supposed to do. That is, it is monitoring everything the device does and limiting the device to specific tasks [59]. An attacker could use manufacturer vulnerabilities to allow them to enter a device and motif the event path. They can use it to change the purpose of devices or even reroute data to them for data monitoring. Lastly, we have the idea of availability. Availability is the idea that the device is always available for use when the user needs it [59]. By an attacker using these vulnerabilities, they can delay the device or even go offline, which would render the availability weak. All of these security impacts are the results of the types of attack that the adversary will carry out. Whether it is a physical, software, or even a network attack. The IoT device at the other end of the attack will affect one or often many of the aspects shown above.

 doi:10.20944/preprints202410.1215.v1

3.1. Hardware Vulnerabilities

Most, if not all, of IoT devices operate without supervision and typically have limited tamper resistant properties that make it extremely easy for an attacker to gain access to the device [59]. The attacker can modify the IoT device with respect to its services that it provides, as well as obtain data that it should not have access to and that could cause serious harm to many individuals [59]. For example, if a hacker were to gain access to a smart doorbell, it would be able to modify all the settings inside, view the data inside, or even delete the data inside. Now, let us take this step further into a very delicate environment. If an attacker had access to a camera in some way for a military base, they could use that access to follow where the data are being sent to. In this case, they would find a server with a lot of other camera data on it. They can then turn off the cameras and attack or even release military secrets that were caught on camera.

3.1.1. Radio Frequency Attacks

The hacker can gain access to a device at the hardware level in a variety of ways, but the most common way is usually through a man-in-the-middle attack. Now, usually when someone hears about a man-in-the-middle attack, they are looking at it from a network level; however, that is only a portion of what this attack has to offer. All electronic devices give off a radio frequency that can be seen with tools such as a spectrum analyzer or even a portable software defined radio such as a hackrf or Ettus radio, etc.

The idea is that the attacker must first recognize the signal. Once the signal is initially found, the hacker is able to recognize the kind of signal being emitted, given the specific properties that the signal gives off. These properties can be anything from bandwidth, frequency, strength of the signal, how it behaves, and looks. These are all key definers for what the signal is actually. Once they have this information, they can figure out how to remove the signal or manipulate it by attacking the weaknesses of the wireless protocol found, since all wireless signals have weaknesses to some extent. This allows them to take control of that signal. However, with RF signal knowledge, the attack is also able to listen in and decipher the communication that is sent. They can either listen to the signal in the middle of the two transmitting devices or they can simply mimic the communication and manipulate the signal to send the communication to the hacker all together or send it to the hacker and the hacker will send the communication to the actual intentional destination spot. The thing about this time of communication is that the hacker is typically the only one who would know that this is happening.

 doi:10.20944/preprints202410.1215.v1

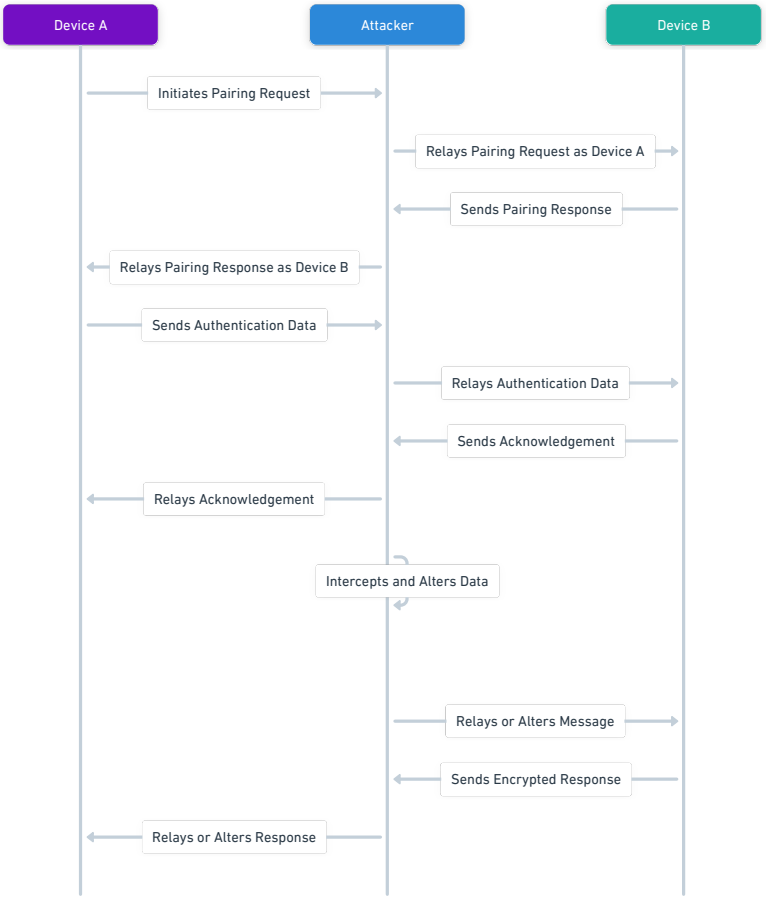

Figure 5. Bluetooth Man-in-the-Middle Attack. The image has been inspired by [60]

The attack can be easily depicted with a Bluetooth man in the middle attack, which is widely known, as shown in Figure 5. Although each version of the wireless signals for this attack will be different, there are many similarities, which is why I am showing this specific one. The idea of this attack is that the attacker will do one of the following. Either they will completely jam the wireless signal band together with a wide-band signal, or they will hop onto the victims device by sending random data in every single time slot. The purpose of this is to "shut down all of the piconets within the range susceptibility", which would ultimately make the user frustrated and want to attempt to re-pair their devices over again [60]. Next, the user is going to re-pair their device, and the attacker can use the man-in-the-middle attack to "forge messages exchanged during the IO capabilities exchange phase" [60]. Now, this becomes extremely easy if the two devices are already paired for the first time, as the attacker does not have to force one device to reconnect. From there, the attacker will simply carry out the attack as shown below: in Figure 5, where the attacker will be connected to both devices simultaneously and relay messages between the two devices while also listening [60]. This ultimately allows the adversary to inject their own messages, modify messages, and listen only to messages.

3.1.2. Hardware Reverse Engineering & Micro-probing

The idea of reverse engineering is to try to understand the processes that make a device work. Now, when you take that to the hardware aspect, this allows the user to know everything they can about a device including the weaknesses, strengths, etc. These types of attack can be broken down into three categories, Invasive, Semi-Invasive, and Non-Invasive, as shown in Figure 6 [61].

 doi:10.20944/preprints202410.1215.v1

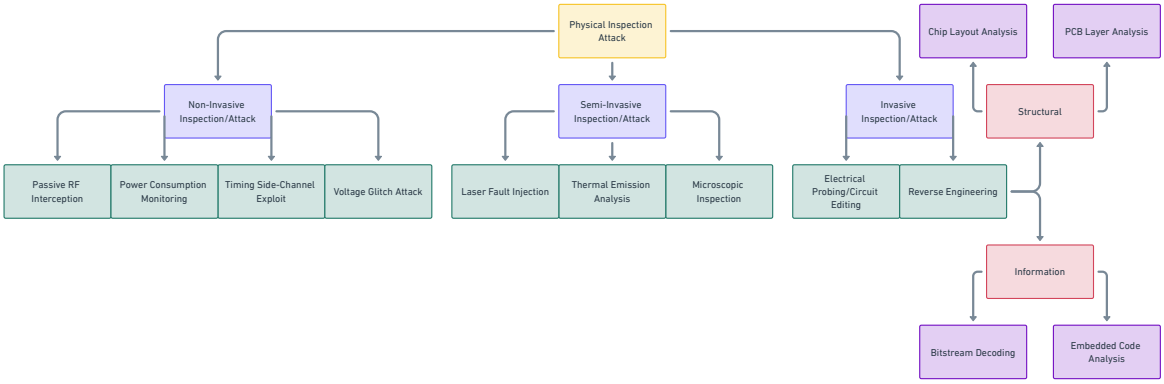

Figure 6. Reverse Engineering and Inspection Attacks. The image has been inspired by [61]

A Non-Invasive attack is when the adversary attempts to grab data without messing with the packaging or structure of the IC or PCB and these can be activated both passively or actively [61]. These can be seen as brute force attacks, fault injection, etc. and allow the attacker to go at the device with full force utilizing tools from their toolbox [61]. These tools can be items commonly purchased from the shelf, or they can also be items that they developed, such as a voltage glitch attack device, which would send a voltage signal to a specific portion of the device at a specific time, which would have the sole purpose of getting data and shutting down something like security parameters at boot [62].

Then we have Invasive and Semi-Invasive attacks which do require access to the internal components of the IC or PCB [61]. These kinds of attack can be done simply by probing the device, soldering wires onto the device, or even cutting down chips on the device in hopes of gaining some kind of information from it. Some of the tools can be a JTagulator, logic analyzer, oscilloscope, laser, signal generator, power supply, X-ray, microscope, and many other tools [61]. Usually this leaves the device with the device not being aware that you were there, as some damage is usually done to the board to make this happen, but in the process a lot of information can easily be gathered [63]. The invasive attacks are when the user directly interacts with the board, and the semi-invasive attacks are the methods that use optical means such as microscope, optical fault injection, optical probing, etc [61].

3.1.3. Implants & Hardware Trojans

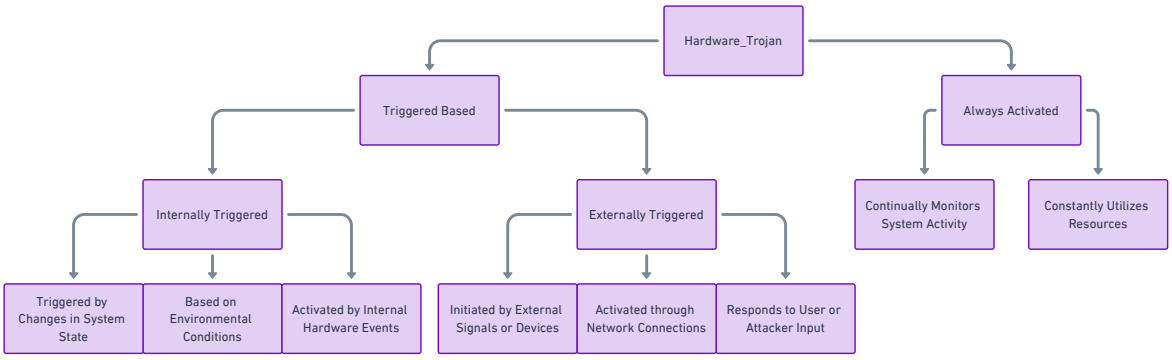

Figure 7. Trojans Activation Modes. The image has been inspired by [64].

By definition, a hardware trojan is any kind of malicious circuit or modification of the hardware of the Integrated Circuit (IC) during the design or fabrication process. And there is simply no way to really prevent this early enough, as the IC design process does not have a good way to secure each stage of the design process and it is unpredictable as to which stage of the design process it will actually be added [64,65].

 doi:10.20944/preprints202410.1215.v1

Ultimately, what makes this kind of hardware attack dangerous is the fact that the user is not knowing that anything is even wrong with their device, and the scary thing is that this can happen to anyone, any kind of device and for any function. In this kind of attack, the adversary can change the input, output, leak information, and control the device all together without the user's knowledge. These types of trojan also vary in how they run as some are activated automatically while others are triggered to activate, as shown in Figure 7, making it extremely unpredictable as a whole and more difficult to even recognize [64,65].

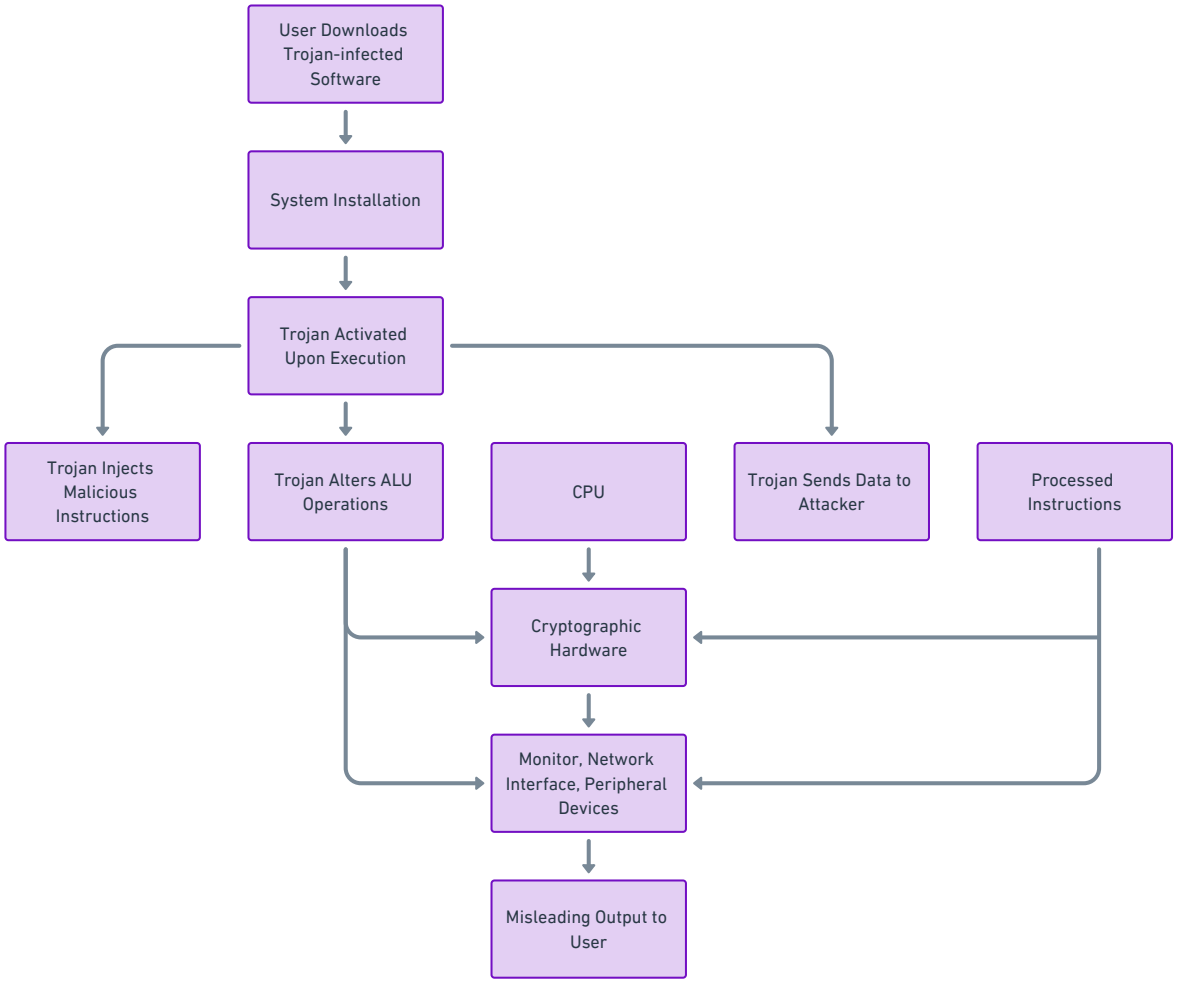

Figure 8. Working Process of a Simple Hardware Trojan. The image has been inspired by [64].

With this all in mind, take it on a larger scale instead of just the IC. This would just be a regular hardware implant and can easily be added to any device at any point of the device's life instead of just being added during the design process like how the Trojan is. These are usually devices added to the circuits of hardware that allow the user to do all of the same as shown above, but what makes this even more dangerous is that this can happen even if the user has already been using the device for years. All the stored data can now be in the hands of the attacker.

3.2. Software Vulnerabilities

IoT functionality can vary greatly from device to device and from use-case to use-case, and what makes this variance is often the software on the device itself. You could have 3 identical IoT devices each with the Bluetooth module that look the same. However, the code within the microchip on the board can be programmed to do different things. We may have one programmed to transmit Bluetooth

 doi:10.20944/preprints202410.1215.v1

only and another programmed to receive it only from specific users. The issue comes into place when access to that software gets into the wrong hands, as that could mean the release of information or even a software implant.

The way an attacker would generally do this to gain access to the device, connect it to a PC, or even probe the pins on the device. In the best-case scenario, they would have access to the actual code and files on the device. In the worst-case scenario, they may only get access to the machine code. However, this is still very dangerous because when the user uploads the code to a program called Ghidra, IDA Pro, or just a general Hex Editor, they can follow that machine code to know what the device is doing and all of the functionality of the device and how it works. They are also able to convert that machine code into the actual C code that can be used later for modifications, as shown in Figure 9 [66]. With this information, they can know everything about the device, from its weaknesses to what it is doing that could be detrimental to the owners of that device.

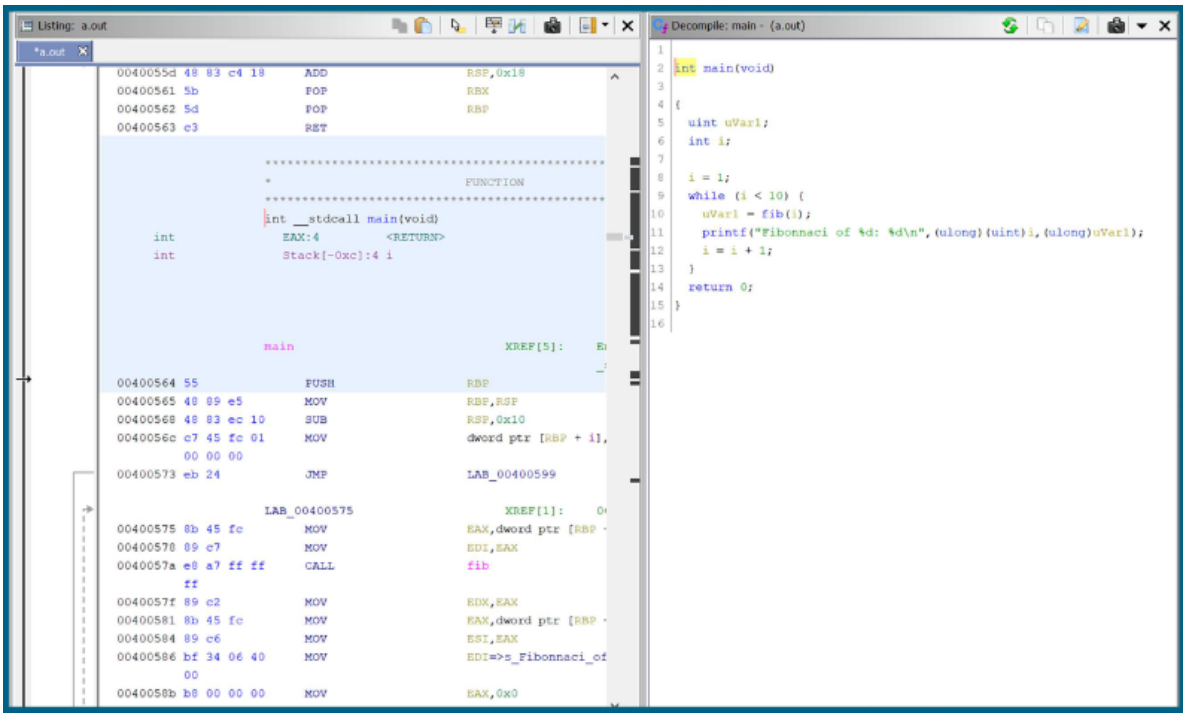

Figure 9. Sample Ghidra Layout [66].

Now, all of this implies that they have the actual device they are attacking in hand, but that is not always the case. IoT devices are relatively inexpensive devices that are open source to some extent. Meaning that if they got their hands on another device that functions the same, they could still capture that device hex data to understand all of this information as well. The only thing that they will not be able to get in this case would be the modified code if there were any. Besides using this attack to understand the IoT device, the attacker would be able to modify the code to allow them to watch the device, have future control, stay hidden, and this would be simply with them adding a few lines of code to the devices code and re-uploading that code to the device, which is typically a quick and easy process.

3.3. Network Vulnerabilities

The idea of a network can be broken down into two different subsets of networks. The typical network that is discussed is through the use of WiFi and Ethernet. This type of network has direct access to the Internet, allowing for the connection of devices through this route. The other type of network that is not commonly spoken about is the kind of network of other wireless signals. This can be a Bluetooth network, zigbee network, UWB network, and so many more. This network is called a wireless sensor network as it typically includes many different wireless sensors, all with different

 doi:10.20944/preprints202410.1215.v1

wireless protocols. During this section, we will explore these two very similar but distinctly different networks and how the vulnerabilities vary between the two.

3.3.1. WiFi & Ethernet Based Networks

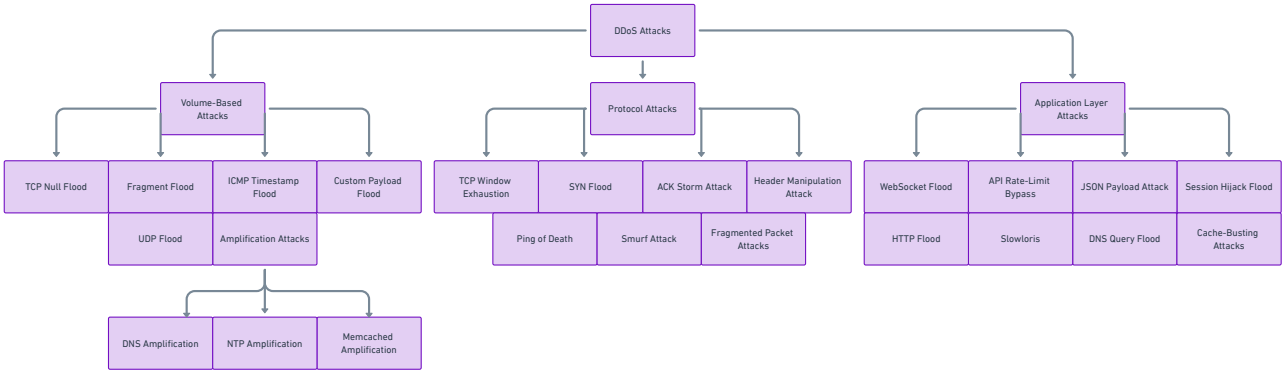

Figure 10. Types of Attacks. The image has been inspired by [67]

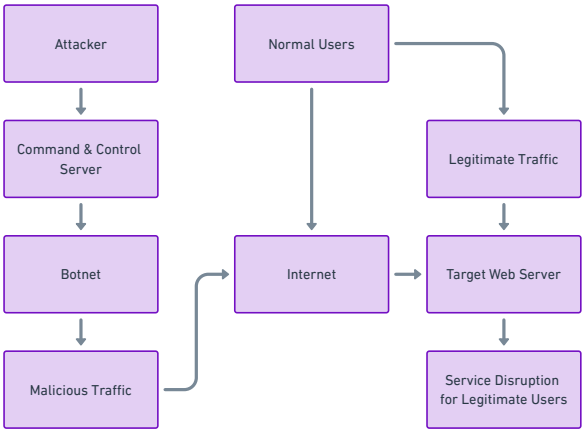

Figure 11. DDoS Attack. The image has been inspired by [68].

The most common network within the IoT ecosystem is the WiFi and Ethernet based networks. The reason is that this allows devices to connect to each other via a cloud platform or via the individual home network. The purpose of this is to allow users to connect to IoT devices from anywhere in the world, even if they are not near the devices. This can be good because it allows IoT data to be transferred far and wide, but it can also be extremely dangerous because it allows the adversary to have more room to gain access to the device data during transfer. One of the most simple kinds of attack on a WiFi enabled device is the Denial of Service (DoS), Distributed Denial of Service (DDoS), and Energy-Oriented Distributed Denial of Service (E-DDos) attacks [67]. A simple DDoS attack would look like in figure 11 where the adversary knows the IP address of the device that is connected to the Internet. From there, they will send a ton of traffic data towards that IP address in hopes of flooding the network of that other device. From there, the network will slow down drastically with more traffic diverted, and it will eventually just stop all together and crash, making that device go off-line [68]. In addition to the use of a DoS and DDos attack, there is also something called an E-DDoS attack. This is the kind of attack where the purpose of the attack is not to shut down the device through malicious traffic but rather to overload the devices power consumption with the purpose of either short circuiting out something on the devices PCB board or by increasing the cost it takes to run the device. These types of attack are common as well, because usually they are not targeted attacks, but rather they are just from someone targeting a group of devices [67].

 doi:10.20944/preprints202410.1215.v1

As we can see in figure 10, various attacks on the Denial of Service attacks can be split into three types of attacks. The first would be the volume-based attacks where the purpose is to flood the TCP, ICP, or UDP which would cause the internal network to be overwhelmed by traffic. Next we would have the protocol based attacks where the attacker would do a SYN Flood, Ping of Death or Smurf Attack. These kinds of attack are designed to overload the resources of the network. For example, a ping-of-death attack would simply send a data packet that is larger than the maximum size allowed by the server. A smurf attack would send out an echo to every single device on the network, which would choke the network in the end, and lastly, a SYN flood attack is when the attacker would initiate a network connection without actually finalizing the connection. Lastly, we have the application layer attacks which would attack the actual web servers or application firewalls. These include Slowloris, which would make the attacker use partial HTTP requests to keep connections to a web server open for a long period of time. It also includes HTTP flood and SMTP attacks. The HTTP flood attack that attempts the application server with http requests and an SMTP attack is any exploitation of the SMTP server in hopes of gaining access to the data within the server [67].

Other than the basic DoS style attacks that are currently present and will always be present in some manner, we have some other attacks that allow for traffic monitoring over the network such as WiFi fingerprinting, which allows for WiFi eavesdropping. For example, if we were to look at a buildings camera system bit-rate, we would be able to infer the object movements from within the building, which would tell the adversary what they believe is happening. It will not give specific information on the uses of the devices, but it would give enough information where the adversary may be able to infer its use case and what kind of device it is [69]. However, some other attacks include analyzing TCP/IP level packets using network monitoring applications such as wireshark or even WiFi over air sniffers. By doing this, the adversary can analyze key characteristics of the specific network of devices. They can see who is talking and what kind of information they are transmitting to each other [69]. And they can take this step further and actually break down the packets to its individual bits and then rebuild it again so that they can see more specific information on what kind of data is being transferred. Figure 12 shows how such an attack may occur with an adversary and target.

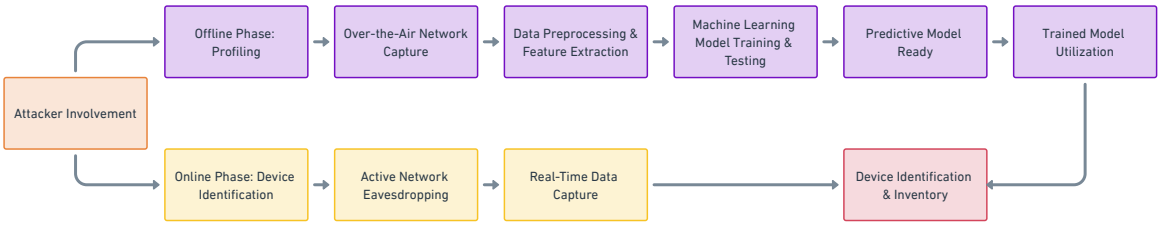

Figure 12. WiFi Over The Air Attack. The image has been inspired by [69].

These kinds of attack, as shown above, can work in multiple different ways; however, in the above case, we have two phases, one being offline and the other being online. During the offline phase, the attacker will be performing the profiling, while in the online phase, the attacker will be performing device identification. The offline phase consists of the network capturing over the air with a sniffer. From that, the attacker preprocesses the data and extracts features so that it can upload them to a machine learning algorithm which would go into training and testing mode. The model will then predict what is happening on that network. The next phase would be the online phase where they would actively eavesdrop on the network and grab that device and make a prediction of the devices that are being used. This is the kind of information that is needed for an attacker as it tells them the inventory of all of the WiFi enabled devices in the vicinity.

And finally, regarding the network attacks for Ethernet and WiFi, there are numerous open-source programs that are built within Kali Linux as well as throughout the Internet that allow users to do essentially anything they want to a network. There are also numerous devices out there built for this purpose, and instructions on how to build your own device for this purpose. Such devices may require

 doi:10.20944/preprints202410.1215.v1

a raspberry pi, Arduino, etc., and some already built-out devices may include WiFi sniffers, WiFi Pineapple, etc. As for software, we have software programs that allow for WiFi cracking, WiFi analysis, such as wire shark, and also specific programs built to inject information. With this being said, these kinds of attack are always evolving as the security evolves, which is the most dangerous thing, as it means that it is difficult to prevent as a whole.

3.3.2. Wireless Sensor Networks

A wireless sensor network (WSN) is essentially any network that is spatially dispersed with specific sensors that transmit data. This is the very reason the vast majority of IoT devices come into play because these networks can range from Zigbee and Zwave to GPS and 5G networks. These networks just refer to all of the wireless protocols as a whole, but for the sake of this paper, I have split the WiFi and Ethernet based approaches from this section due to the amount of vulnerabilities that exist in those networks alone. For wireless sensor network attacks, the DoS taxonomy can be broken down in Figure 13.

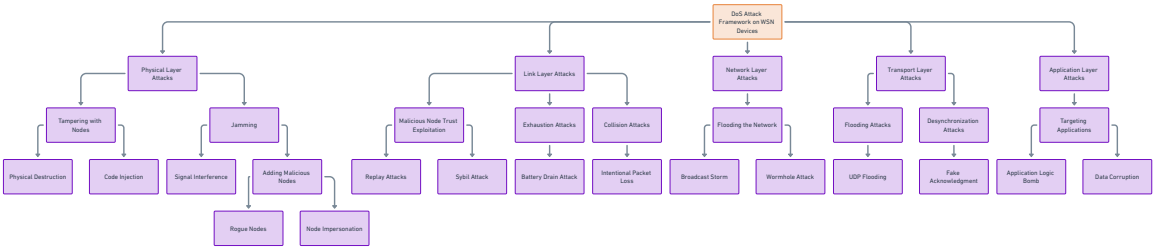

Figure 13. Wireless Sensor Networks DoS Attacks. The image has been inspired by [70].

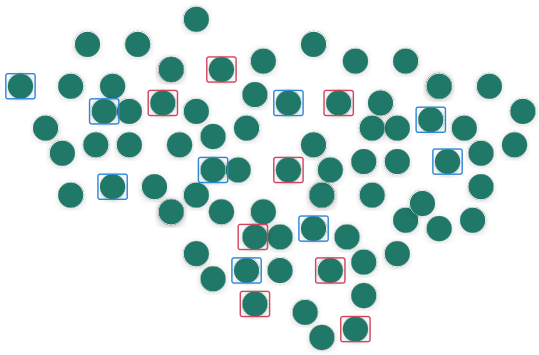

Figure 14. Wireless Sensor Network Diagram. The image has been inspired by [71].

In the above framework, we can see that for WSN devices, a DoS attack framework is filled with a variety of attacks that include those of WiFi and Ethernet, but also many more. The most common attack from this framework is on the physical layer DoS attack, where the adversary would be tampering with nodes in the network as well as jamming them. By doing this kind of attack, they are able to gather the code from the device itself and either modify it or delete it, and for the jamming aspect, if they were to add a malicious node to the network, they would be able to stop nodes from transmitting or even receiving data by blocking the connections [70]. We then have the link layer DoS attacks which can include the malicious node trying to gain the trust of a neighboring node, exhausting a neighboring node by constantly sending requests to it, and then collision attacks which have a purpose of shutting down the node all together by erroring it out [70]. Then we have the network layer attack where the goal is to drown out the network in a similar way to what was seen above in the WiFi section. Then we have the transport layer, which includes flooding attacks where the network is flooded with malicious nodes, and then desynchronization attacks where they rearrange the messages that are sent between the node, making them not in sync or behind other nodes on

 doi:10.20944/preprints202410.1215.v1

information [70]. Lastly, we have the application layer attack where, much like in the WiFi section, they target the applications itself.

In figure 14, we can see that the green dots are the good nodes that just collect information with their sensors. Then they receive directions from the blue square nodes and the red square nodes are the malicious nodes. In this case, we can see that the malicious nodes would look identical to the regular nodes and the source node, which allows those nodes to just integrate themselves as part of the system [71].

Looking at the idea of adding a malicious node to the network, the adversary can also perform what is called a black hole attack, as shown in Figure 16. In this kind of attack, the idea is that the malicious node would get all other nodes to redirect their data transfer to them so that it has full control of the data. Usually by doing this, they are able to stop the full data transfer from occurring by essentially being a black hole for the data [71,72]. A similar style of attack to this is what is called a wormhole attack, where the data is essentially rerouted to a completely different node than the node for which it was intended by creating a tunnel, as shown in the following figure [73,74].

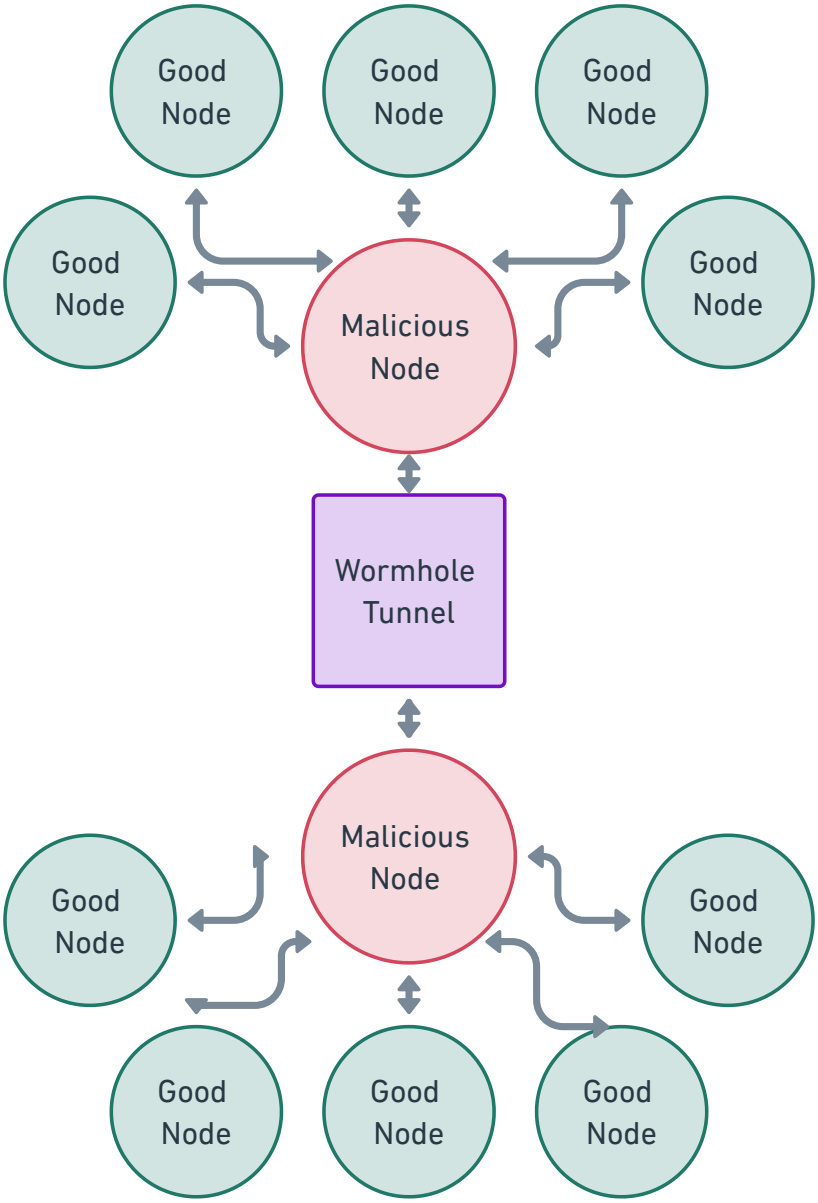

Figure 15. Wormhole Attack Tunnel. The image has been inspired by [74].

 doi:10.20944/preprints202410.1215.v1

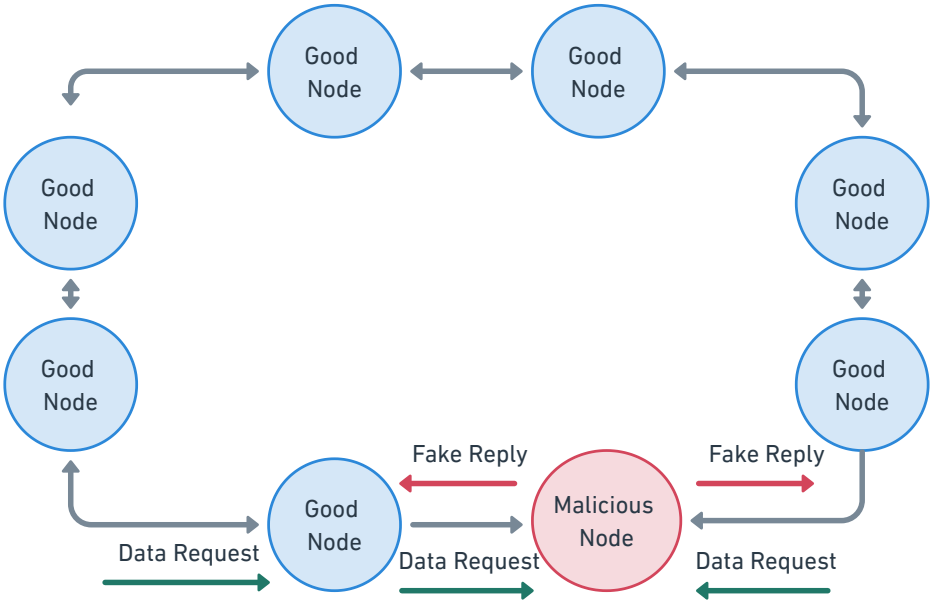

Figure 16. Blackhole Attack. The image has been inspired by [75].

In Figure 15, we can see that the malicious node grabs data from the good nodes near node A and then sends the data through a wormhole tunnel to node B, and node B does the same. This allows the two nodes to control the flow of the data and reroute the data between each other [74]. And we can also break down the modes of a wormhole attack into four distinct categories. The first one is a packet encapsulation, where if there are two or more nodes, data can be compressed and sent between these two nodes, which would prevent the hop count increments that usually occur in wireless sensor networks [74]. Then we have the packet relay mode, where any node within the network can be affected and launch the attack. This is also known as a relay-based attack [74]. Next we have the out-of-band channel attack where only one sensor is needed for the attack; however, it must have a high transmition power, which would affect the route of the packet that passes through it. Lastly, we have the protocol distortion mode where malicious nodes attempt to invite traffic by altering the routing protocol [74].

Just as there are sniffers and other technologies built just for Wi-Fi networks and Ethernet networks, the same applies to every other kind of signal out there. For example, you can find and purchase prebuilt UWB sniffers, Zigbee sniffers, Zwave sniffers, Bluetooth sniffers, and so many more. Even if there is no prebuilt one that you can buy off the counter, they also have open source code that works with Arduino boards and Raspberry Pi boards that would allow you to do the same. With this being said, this allows the attacker to analyze the packets of any kind of signal in use, no matter what kind of wireless signal it is.

3.3.3. Cloud Based Networks

When discussing network vulnerabilities within the Internet of Things, it is important to mention a big and recent important player in the world of the Internet of Things, the IoT cloud computing architecture. As IoT technology alone cannot fully meet the growing number of consumers and their computing requirements, the concept of IoT cloud computing comes into place and refers to the integration of IoT technology and cloud computing resources. Although there are pros to the integration of IoT and cloud computing, there are several security issues that also play a role in the cloud of the IoT. Since there are several articles that do a good job summarizing different IoT cloud vulnerabilities, I am going to cover two contemporary vulnerabilities that relate to Phantom-Delay Attacks and Post-quantum Cryptography.

 doi:10.20944/preprints202410.1215.v1

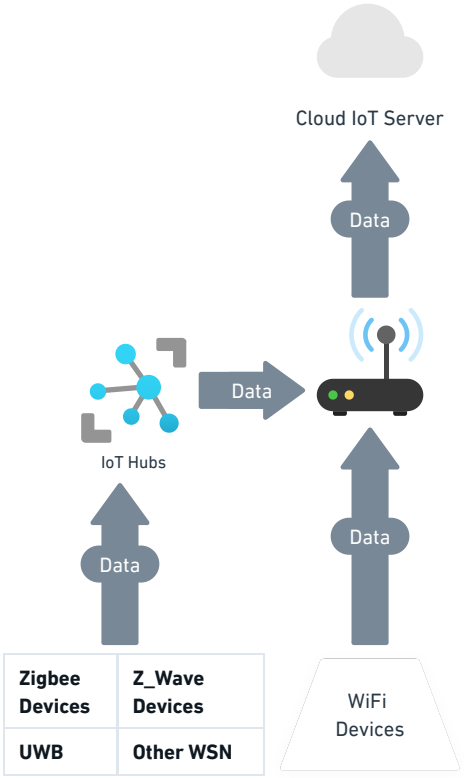

Figure 17. System model of a typical smart cloud-based home deployment. The image has been inspired by [76].

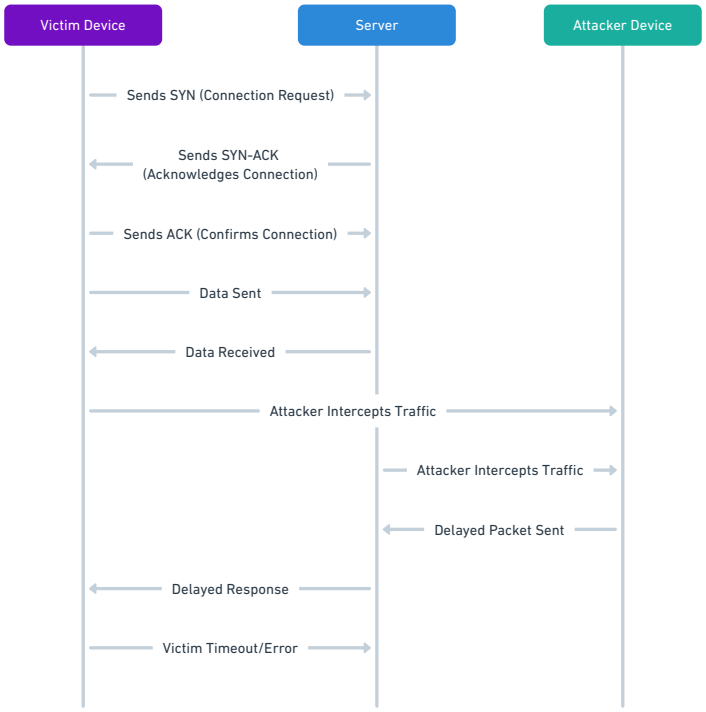

Figure 18. Phantom Delay Attack. The image has been inspired by [76].

Phantom-Delay Attacks are IoT vulnerabilities that were recently discovered. These attacks exploit a vulnerability in communication between IoT devices and servers. Attacks allow the attacker to delay IoT messages, resulting in several serious consequences [77]. There are two main attack

 doi:10.20944/preprints202410.1215.v1

primitives when discussing Phantom-Delay Attacks: IoT Event Message Delay (e-Delay) and IoT Command Message Delay (c-Delay). There are two types of IoT events between an IoT device and a server: an IoT device can send a 'device state' event (such as a 'motion active' event) to the server, and a server issues an IoT command (such as a 'front door lock' command). Because of the time it takes to transmit a packet, a delay will always occur from the moment of an event/command to the moment of delivery (typically sub-seconds and do not cause issues). Using a Phantom Delay Attack, an adversary can increase the delay to minutes or even hours, which means that when the IoT cloud server receives the event sent a while ago, it will still assume that the device's state update is fresh [78].

There are three types of attack categories: State Update Delay Attack - uses e-Delay to packets of an event message that reports an IoT state update (can be used on critical IoT devices such as IoMT devices and Internet of Agricultural Things that require a fast response); Action Delay Attack – Due to automation, an event can trigger a spontaneous action, so by using an e-Delay attack you can delay this action (in theory it could be used to delay actions like an insulin pump that uses a continuous glucose monitor (CGM) to check blood sugar level); Erroneous Execution Attack – there are two kinds of Erroneous Execution attacks: 1) Spurious Execution which happens when an action command that should not happen still occurs (even though the automation condition is set to false, a delay attack will keep it as true); 2) Disabled Execution, which happens when an action command that should occur, still do not occur (even though the automation condition is set to true, a delay attack will keep it as false).

Unlike jamming, these attacks can be launched from an ordinary Wi-Fi device and do not discard any packets, and thus do not trigger re-transmission. To deploy phantom delay attacks, an adversary must be able to have one Wi-Fi device in control of the victim's environment, which can be used to sniff traffic and hijack the TCP session of the target device (ARP spoofing can be used) [79].

4. Artificial Intelligence

Artificial intelligence in a high level is a concept that is almost impossible to fully explain. The idea behind Artificial Intelligence is to have a machine think like a human which is difficult as we don't even fully understand human intelligence. But in simple terms for the definition, AI is the ability of a computer or machine of some sort to function, think, and act like a human when it comes to various tasks, whether they are simple or complex. However, something to note is that AI does have a massive weakness, that is, the fact that it is unable to do tasks that are simple for humans and is only able to do tasks that are more complex for humans. That is, a machine is unable to do tasks that come naturally to us such as walking, talking, eating, etc. So this leads to how a machine is actually able to do all of this. Well that is simple. It gets help from agents, whether those agents are sensors or information that humans feed to the machine, and the way machines go through all of this data is through the use of specific AI models.

4.1. Artificial Intelligence General Overview

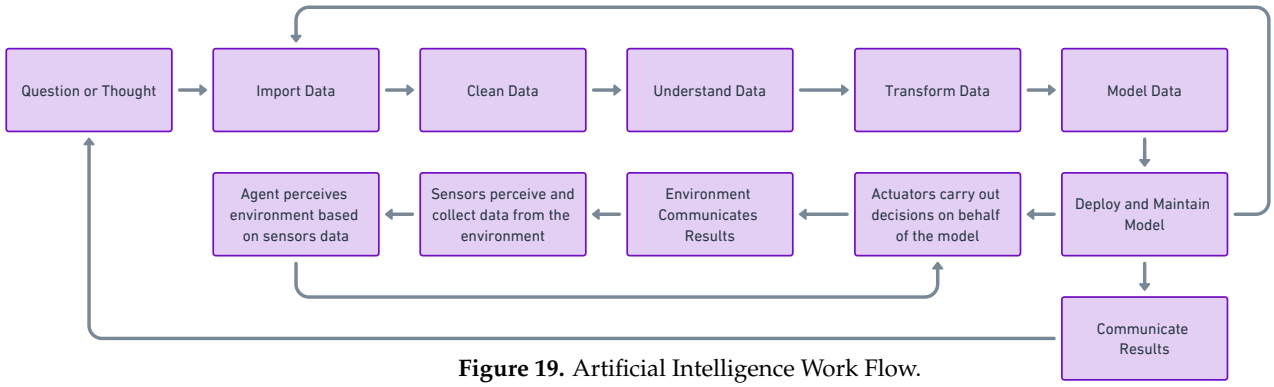

Figure 19. Artificial Intelligence Work Flow.

 doi:10.20944/preprints202410.1215.v1

AI sees the world and interacts with the world through the use of agents, as shown in Figure 19. Figure 19 may seem slightly confusing, but it is quite simple. The Environment feeds data through the sensors which get perceived in the AI model. From there, the AI model will typically cause something to happen as a result of the data that it receives, and it then interacts with the environment. An example of this would be if you had an IoT sensor that measures the amount of water and nutrients in the soil. So, given that example, the sensor will detect something from the environment that would give a data readout for the water content and the nutrient content. The AI model will receive the readout of the data and make an assumption based on the data that the water content is high/low or the nutrients are bad/good. From these, the AI machine will react based on its perception of the data, and it would then activate the sprinklers or leave them off for the water content section or even activate the fertilizer spray for the nutrients. Therefore, the sprinkler and fertilizer spray are simply the actuators that cause a reaction on the environment according to the perception data. Now, this does not always have to be interacting with the world, but rather, this can all be kept in a closed ecosystem. Think of another example in which you are playing chess against the computer. You make a move, the computer will analyze your move and make another move based on its perception of the data. This is the same thing as shown in the environment data above; however, it is all kept within a computer system and does not interact with the physical world. So this leads to the idea of what defines the environment for AI? Well, AI can have many different types of environment. We have full observable where the sensors always tell the AI model what is happening in the world in ALL aspects around versus a partial environment where sensors only give the mode, data in some or partial aspects and not all the time. Then we have deterministic, where the next move that the AI makes is based around the current location [80]. An example of this would be playing chess with a computer. Then there is episodic where the agent full experience is separated into different episodes and each episode consists of a single action being performed by the agent. Static environments mean that the world is never changing versus dynamic where the world is always changing. Discrete environments are where there is only a limited amount of percepts and actions versus continuous environments where there is no limit. And lastly, we have a single agent versus a multi-agent where one has only a single agent acting and the other relies on many agents. So what is an agent? An agent can be a simple reflex agent where it reacts to what is happening. We have agents that are continuously taking in data from the environment. Then there are goal-based agents that operate under specific goals, and then utility-based, where the agents are designed for a specific utility for the model. The last two ideas that are important to note regarding AI is the idea of how does it even work, like what does it look like?

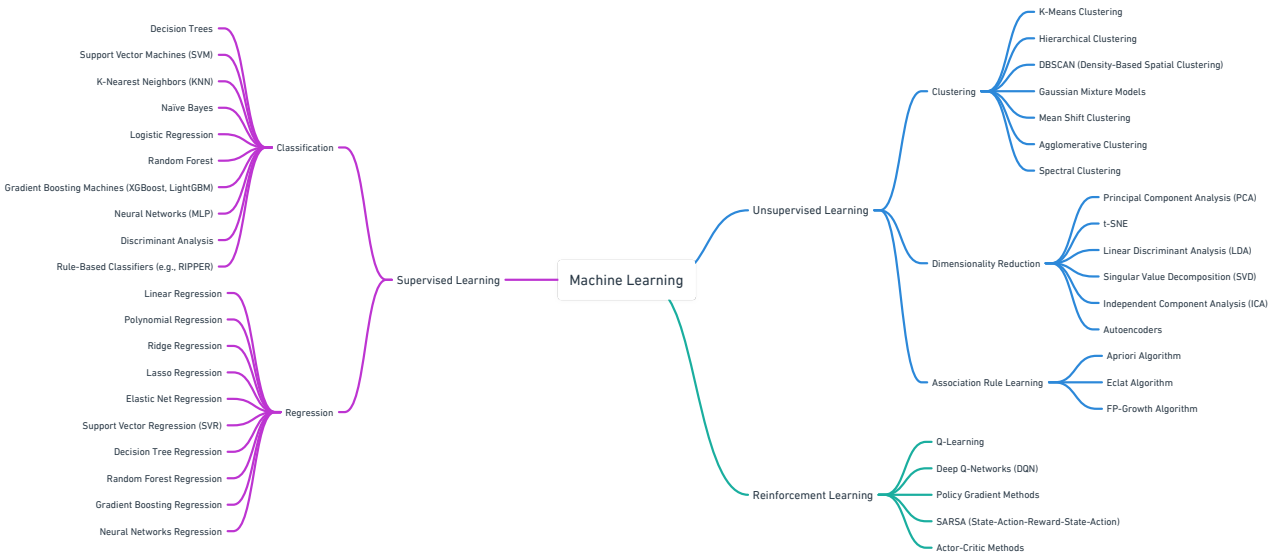

Figure 20. Machine Learning Techniques.

 doi:10.20944/preprints202410.1215.v1

The answer to these questions can be seen in Figure 19. As we can see, we must start with a question or thought. From these we would need to import data to train the machine learning algorithm. But first, we would need to clean the data, which in simple terms just means to get rid of impurities and keep only the best data. So, we would make sure all of the data are similar in categories, etc. Then we would need to understand the data, modify and transform the data as needed to fit our algorithmic model needs, and do this process over and over again. The reason why this is done over again is to get the highest accuracy score possible to use when the model is tested on new data. But after we are done, we will then deploy and maintain the model by testing new data into it, and then we would have results be exported via either the use of a functionality or through actual written data. The last concept needed is related to the above idea of training and testing the data. Because how are you suppose to do that? Do you need to tell the model what the data is or do you let it figure it out on its own? Well, this is the difference between supervised and unsupervised learning.

This idea can be seen in Figure 20 where the unsupervised learning is that the model will learn only from the inputted data on its own using the clustering method. In supervised learning, the data is already given a label of how it should be viewed, and there is already an idea on how the output and input should look with regard to the model. This method also utilized classification and regression of the data. As we can see, all of the above methods utilized a different kind of machine learning algorithm, and that is because different algorithms are used and better for different situations. This then leads to IoT because, in order to keep the IoT secure, different algorithms will need to be used for specific reasons.

4.2. Artificial Intelligence & IoT

The main purpose of Artificial Intelligence is to help automate decision-making and actions. The hope of using this kind of model with the IoT would allow the IoT devices and the model running on them to infer what is actually happening. That is, the IoT device and the owner of the IoT device would be able to automate the IoT use cases [81]. A case where this would benefit would be in the farm situation. A farmer may have many IoT devices within the field that measure sunlight, the amount of water in the soil, nutrients, etc. Using AI in this circumstance, the farmer can then say when the water level of the soil reaches a specific level, to activate the sprinklers, or if it becomes too moist with water, then to drain the field of some water. From them we can take it a step further and say that while measuring the water content, we measure the nutrients in the soil and we can automatically feed the nutrients that are low to the crops in the field. This would save the farmer time and allow for much fewer human error situations if one arises. However, in the above case, IoT devices are still at risk of cyberattack at all levels of the device and the IoT ecosystem itself. So, let us now look at the idea of using artificial intelligence to actually keep the device secure.

4.2.1. Learning Based Detection for Cybersecurity Use

When it comes to machines, we can break that down into multiple steps. The first steps would be basic machine learning, while the second would be deep learning. But what is the difference between the two? Well, that can be simply seen as the idea that deep learning relies on neural networks with multiple layers, while machine learning does not. Also, note that machine learning is a much broader idea that encompasses many more techniques and ideas than deep learning, where it is more specific. In the basic machine learning section and the deep learning section, we can break that down even further as shown below in Figure 21 [81].

So, let us look at figure 21. In this figure, we can see that the basic machine learning model of the diagram consists of supervised, unsupervised, semi-supervised and semi-unsupervised learning methods. Supervised learning is essentially what happens when specific data points are already predefined for the model to reach from a specific set on inputs [81]. And under this section, we have the regression and classification models which in simple terms mean that for regression, the idea is to predict an output, while in classification, the goal is to assign an input based on predefined categories.

 doi:10.20944/preprints202410.1215.v1

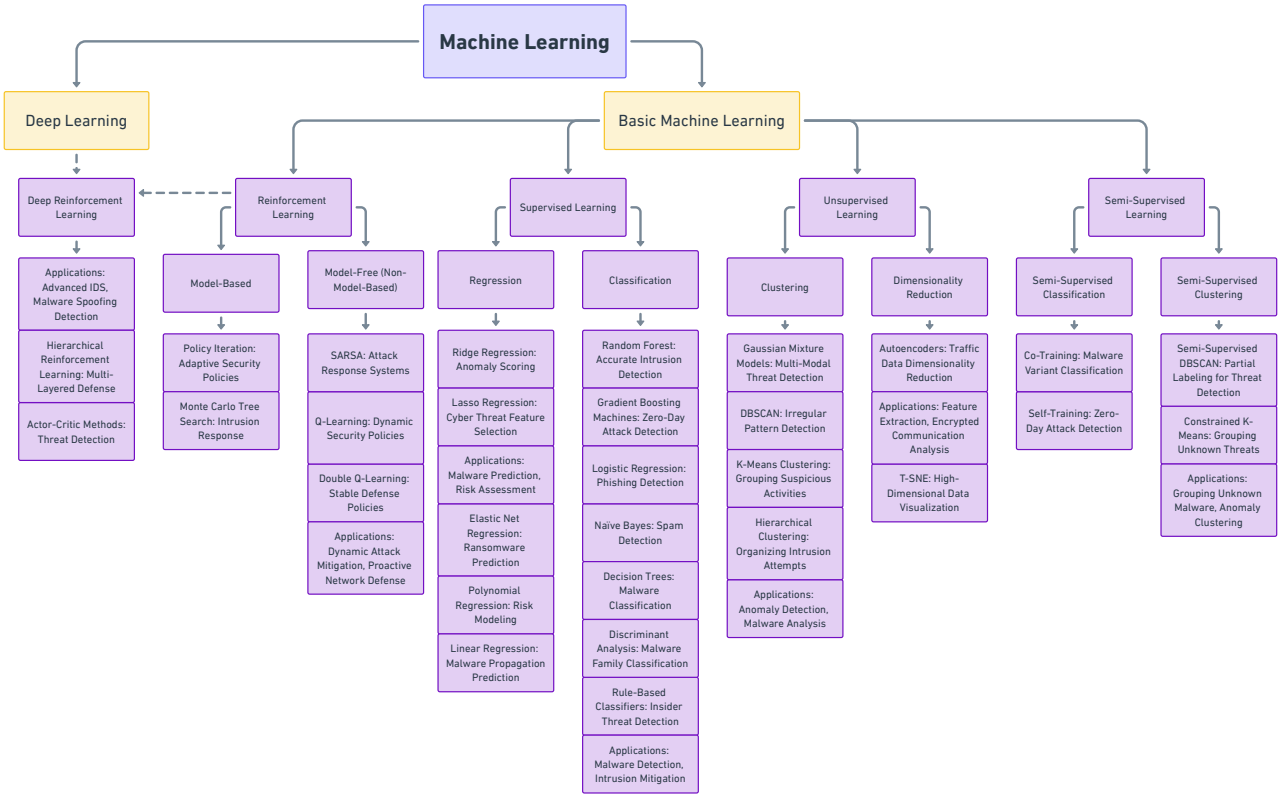

Figure 21. Machine Learning Techniques. The Image has been inspired by [81].

In these categories, we are able to perform IoT security tasks such as malware analysis, intrusion detection, anomaly detection, attack mitigation, and access control, as well as many more. Then we have the section that is unsupervised learning and that consists of clustering and dimensionality. Unsupervised learning, where the environment only provides inputs with no desired goal or target. It is also important to note that it does not require any labeled data, as it will move the data into groups on its own [81]. The idea of clustering is where the data are clustered according to similarities, while in the reduction of dimensionality, the number of features of the data is reduced while it also preserves as much of the original information as possible. For these two models under the unsupervised learning tree, we can perform security tasks such as anomaly and intrusion detection, as well as malware analysis. Then we have semi-supervised learning, which is when data are provided but it is not labeled [82]. This section consists of anomaly detection and zero-day attacks, as these attacks mean that you are the first to identify a vulnerability or attack. Lastly, for this section, we have reinforcement learning, which is where no outcomes are defined and the model learns only from trial and error through the interaction with the environment [81]. Under this section, we have model-based vs. nonmodel-based. The differences are that in a model-based model, the model builds a predictive model of the environment, while in a nonmodel-based model, no predictive model is needed [83]. These methods allow for control, malware analysis, DDOS attacks, and authentication. This was it for basic machine learning; however, we still have the deep learning side of the tree, which consists of deep reinforcement learning, which is when the model learns from trial and error but also builds a neural network that grows as it gains more knowledge [84]. In these models, we have jamming attacks, IDS, malware, and spoofing. As we can see, in each learning model time, different benefits are found for different security ideas. All of these can be implemented through strategically to allow for an IoT devices to have full security functionality for detecting, mitigating and even attacking.

 doi:10.20944/preprints202410.1215.v1

4.2.2. Deep Learning for IoT Vulnerabilities

Out of all of the techniques used for vulnerability analysis, as seen in the figure. The one that is higher above all others in all aspects is deep learning models. The reason is that in a deep learning model, raw data is constantly gathered and processed. As more data is processed, more neural network chains are developed, which means that the system learns on its own each time. This is perfect for IoT vulnerabilities, where attackers are constantly finding holes within the mitigations that are created and within the signals that are used. It is important to note that, by using the deep learning method, it does take time to train and for it to be the most accurate it can. However, once trained, it would be ready to use its neural network to combat any problem by comparing current issues with previous ones and comparing vulnerabilities with other vulnerabilities of wireless signals to come to a conclusion.

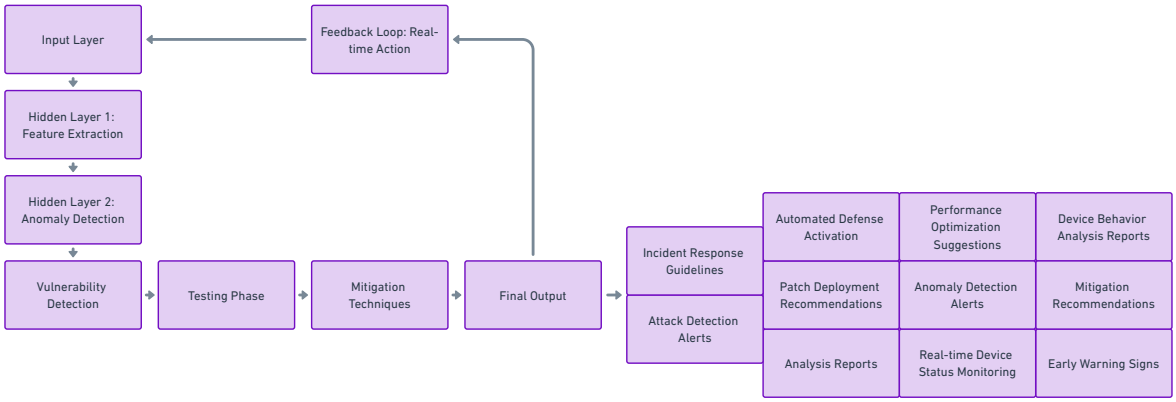

Figure 22. IoT Neural Network. The image has been inspired by [85].

Take a look at Figure 22. In this figure, we have all the IoT devices that feed raw data into the input layer of the neural network. The neural network will then analyze these data and run the newer data against the previously known data to find similarities. It will then look at the trends in which the data is going and be able to think on its own above where the next vulnerability may be. Then it will perform a test on its own in the area in which the vulnerability may be in and also use knowledge from previous data and vulnerabilities to figure out how the vulnerability may work [86]. From there, it will create a mitigation technique based on what worked in the past with newer data and then release the information it discovered. This could ultimately be used for attack detection, attack analysis, early warning signs, and mitigation issues.

4.3. Issues with using Artificial Intelligence for IoT Security

When it comes to using AI for IoT, a few issues come into mind that must be addressed. The first is the issue of data availability. For this, we would have to think about the fact that AI to be the best it can usually needs to train and test thousands of data pieces. It cannot be just any data pieces, but rather the data itself needs to be of quality as well, which is another part of the issue. Since IoT is so vast in protocols, weaknesses and vulnerabilities are as well which make it difficult to train an AI algorithm to detect issues on one IoT device, let alone a whole system of them [87]. The next major issue that comes to mind is power and storage. Once again, AI needs to be able to store a ton of data so that the model can constantly train, and typically IoT devices are standalone devices that only function for one purpose, so they may not have the requirement's required to hold the data or even run the algorithm due to the lack of a GPU and adequate power which may be needed in some instances. It may be challenging to give a simple IoT device the ability to run its own AI algorithm, in general [86].

5. The Cloud

The cloud model/concept is for convenient, ubiquitous, on-demand network access to a pool of shared computing resources that are all able to be configured specifically to the needs of the end-of-user.

 doi:10.20944/preprints202410.1215.v1

This allows for a cheaper infrastructure for the end user, as they would not have to purchase, configure, and maintain the computing infrastructure themselves. It also allows for scalability and customization's which mean that you can add, modify, or even remove resources depending on your needs and the needs of your end users [88]. And lastly, this kind of platform allows for reliability where you know everything is backed up and being monitored by technicians on the cloud providers site. That is, if something goes wrong, know that they will solve the issues. Something that is important to know when it comes to the cloud is the fact that there are many different service models. One of the most popular models is IaaS which is Infrastructure as a service [89]. This service model means that the cloud provider maintains the infrastructure and resources, while the customer will maintain the OS, security policies and the application itself. Then we have PaaS, which is Platform as a Service. In this model, the customer is only responsible for the application itself, while the cloud provider will do everything else. Lastly, we have SaaS, which is Software as a Service. In this, the cloud provider manages the application, infrastructure, and scaling needs. There are also many others, such as FaaS (Function as a service), DaaS (Desktop as a service), and many more. However, the main three can be seen below in Figure 23 [90,91].

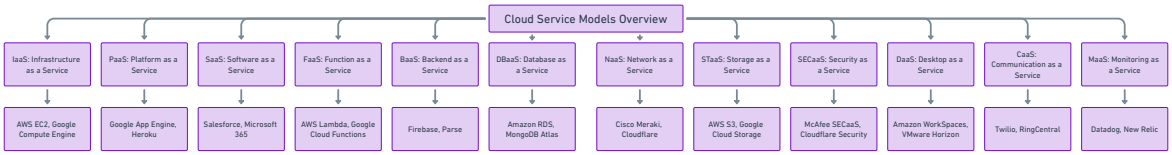

Figure 23. Cloud Service Models.

5.1. Artificial Intelligence & The Cloud for IoT Security

When discussing Artificial Intelligence models, we will usually find ourselves talking about the model training process, which involves many complex computations. During the training phase of an AI model, we will usually use computers that contain graphics processing units that can handle a large amount of processing capacity but at the same time, this oftentimes can reach very high prices [92]. As discussed above, our world is heading toward the cloud computing trend. Cloud computing provides us with scalable and on-demand resources such as powerful computing resources that are made for AI model training. This helps developers and researchers save time and money in maintaining on-premises machines and provides easy access to fast, scalable cloud-based computing machines. Using cloud-based artificial intelligence, IoT devices can instantly analyze data streams and identify suspicious activities and potential vulnerabilities. This type of architecture is called artificial intelligence of things (AIoT). Figure 24 shows an overview model of the AIoT architecture. The end layer is used to detect, act on and control the physical world and executes small AI computational tasks or pre-processed data. The edge layer comprises various nodes, including wireless base stations, routers, IoT gateways, and access points [93]. These nodes receive data from the end devices and send control flows back to the devices via wireless interfaces. The upper-edge servers compute tasks using the received data. Edge servers also perform authentication, authorization, offloading, and storage of data that passes between networks. The cloud layer is used to process large data and train AI models and can also provide additional storage resources [93].

 doi:10.20944/preprints202410.1215.v1

Figure 24. Overview of an AIoT architecture. The image has been inspired by [93].

5.2. Issues with using Artificial Intelligence with The Cloud for IoT Security

Although the AIoT architecture is an important component in the integration of IoT, AI, and cloud computing, some issues and concerns may arise. When sending data to the cloud for storage or processing, one of the first questions to ask yourself is how confidential is this data. For example, data sent from day-to-day smart home IoT devices differ from data sent from Internet of Battlefield Things (IoBT) devices in how confidential that data is. Although using cloud computing services on premises has its benefit, one thing to remember is the potential risk associated with third-party access and the need for robust encryption and access control mechanisms [94]. As the integration of AI, IoT, and cloud computing progresses, advances in data confidentiality and the implementation of robust security protocols will be essential to harness the full potential of AIoT while mitigating its risks [95].

6. Conclusion and Future Research to be Worked

This paper discusses critical topics in modern cybersecurity, focusing on the cloud, artificial intelligence, and the multifaceted nature of vulnerabilities within hardware, software, cloud infrastructure, networks, and sensor networks. Post-quantum cryptography is also briefly mentioned as an emerging area of concern. Through this analysis, it is evident that no component within networked and IoT environments is fully secure, as vulnerabilities exist in various domains. Addressing these vulnerabilities through rigorous mitigation strategies is imperative for securing future systems. The research introduces an AI-driven architecture designed to identify and mitigate these vulnerabilities. This architecture will leverage AI to detect anomalies across wireless sensor networks, cloud environments, wireless networks, hardware, and software. A cloud-hosted AI infrastructure will be deployed on embedded IoT devices using Nvidia Jetson Orin Nano, ensuring secure data transmission through encryption. The AI system will continuously monitor IoT devices, performing intrusion detection and conducting penetration tests to classify and address identified vulnerabilities. For unknown threats, the AI will adaptively employ historical mitigation strategies

 doi:10.20944/preprints202410.1215.v1

to resolve potential security risks. Furthermore, this research advocates for further enhancement of the security of IoT devices by incorporating post-quantum cryptographic techniques. As quantum computing progresses, traditional cryptographic methods will become increasingly vulnerable to its advanced computational capabilities. To future-proof IoT systems and protect sensitive data, the early implementation of post-quantum cryptographic methods is critical. Although this research focuses primarily on AI-driven vulnerability mitigation, the incorporation of post-quantum cryptography is essential to ensure that IoT systems remain secure in the face of evolving quantum technologies.

7. Acknowledgment

The authors thank Eadan Plotnizky for his contribution to the research efforts and to the writing / editing of this survey. The authors also thank Dr. Mehrdad Nojoumian for his guidance, support, and resources as we wrote this survey and conducted the research. Finally, we thank anonymous reviewers for their constructive feedback and inspiring comments. The invaluable comments of the reviewers significantly improved this survey document.

Table 1. Common Abbreviations.

Abbreviation	Full Form	Abbreviation	Full Form	Abbreviation	Full Form
5G	Fifth Generation (Cellular Network)	6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks	AES	Advanced Encryption Standard
AI	Artificial Intelligence	AIoT	Artificial Intelligence of Things	AMQP	Advanced Message Queuing Protocol
ARP	Address Resolution Protocol	BLE	Bluetooth Low Energy	CGM	Continuous Glucose Monitor
CoAP	Constrained Application Protocol	CPU	Central Processing Unit	DaaS	Desktop as a Service
DDoS	Distributed Denial of Service	DDS	Data Distribution Service	DL	Deep Learning
DoS	Denial of Service	E-DDoS	Energy-Oriented Distributed Denial of Service	ECG	Electrocardiogram
FaaS	Function as a Service	FPGA	Field-Programmable Gate Array	GPU	Graphics Processing Unit
HMI	Human-Machine Interface	HTTP	Hypertext Transfer Protocol	IaaS	Infrastructure as a Service
IC	Integrated Circuit	ICMP	Internet Control Message Protocol	ICP	Internet Cache Protocol
IDS	Intrusion Detection System	IEEE	Institute of Electrical and Electronics Engineers	IoAT	Internet of Agricultural Things
IoBT	Internet of Battlefield Things	IoMT	Internet of Medical Things	IoT	Internet of Things
IP	Internet Protocol	IPv6	Internet Protocol Version 6	JTAG	Joint Test Action Group
LAN	Local Area Network	LoRaWAN	Long Range Wide Area Network	LPWAN	Low-Power Wide-Area Network
LTE-A	Long-Term Evolution Advanced	ML	Machine Learning	MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrowband Internet of Things	NFC	Near Field Communication	PaaS	Platform as a Service
PC	Personal Computer	PCB	Printed Circuit Board	QBit	Quantum Bit
SaaS	Software as a Service	SMTP	Simple Mail Transfer Protocol	SSL	Secure Sockets Layer
SYN	Synchronize	TCP	Transmission Control Protocol	TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security	UDP	User Datagram Protocol	UAV	Unmanned Aerial Vehicle
UWB	Ultra-Wideband	V2I	Vehicle-to-Infrastructure	V2V	Vehicle-to-Vehicle
WEP	Wired Equivalent Privacy	Wi-Fi	Wireless Fidelity	WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network	WPA	Wi-Fi Protected Access	WPA2	Wi-Fi Protected Access II
WPA3	Wi-Fi Protected Access III	WSN	Wireless Sensor Network		

References

1. Zhong, C.L.; Zhu, Z.; Huang, R.G. Study on the IOT Architecture and Gateway Technology **2015**. pp. 196–199. doi:10.1109/DCABES.2015.56.

2. Pico-Valencia, P.; Holgado-Terriza, J.A.; Quiñónez-Ku, X. A Brief Survey of the Main Internet-Based Approaches. An Outlook from the Internet of Things Perspective **2020**. pp. 536–542. doi:10.1109/ICICT50521.2020.00091.
3. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials* **2015**, *17*, 2347–2376. doi:10.1109/COMST.2015.2444095.
4. Singh, D.; Tripathi, G.; Jara, A.J. A survey of Internet-of-Things: Future vision, architecture, challenges and services **2014**. pp. 287–292. doi:10.1109/WF-IoT.2014.6803174.
5. Arasteh, H.; Hosseinnezhad, V.; Loia, V.; Tommasetti, A.; Troisi, O.; Shafie-khah, M.; Siano, P. Iot-based smart cities: A survey **2016**. pp. 1–6. doi:10.1109/EEEIC.2016.7555867.
6. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *Internet of Things Journal, IEEE* **2012**, *1*. doi:10.1109/JIOT.2014.2306328.
7. Sivapriyan, R.; Rao, K.M.; Harijyothi, M. Literature Review of IoT based Home Automation System **2020**. pp. 101–105. doi:10.1109/ICISC47916.2020.9171149.
8. CardiacSense. Heart Rate Monitor Watch. https://www.cardiacsense.com/heart-rate-monitor-watch/, 2023. Accessed: June 2023.
9. Islam, S.M.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K. The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access* **2015**, *3*, 678–708. doi:10.1109/ACCESS.2015.2437951.
10. Farooq, M.S.; Riaz, S.; Abid, A.; Abid, K.; Naeem, M.A. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access* **2019**, *7*, 156237–156271. doi:10.1109/ACCESS.2019.2949703.
11. SeeTree. About Us. https://www.seetree.ai/about-seetree, 2017. Accessed: September 2023.
12. Shachar, O.; Yushchuk, M.; Salton-Morgenstern, G. Recurrent pattern image classification and registration, Jan, 2020.
13. Kott, A.; Swami, A.; West, B.J. The Internet of Battle Things. *Computer* **2016**, *49*, 70–75. doi:10.1109/MC.2016.355.
14. Russell, S.; Abdelzaher, T. The Internet of Battlefield Things: The Next Generation of Command, Control, Communications and Intelligence (C3I) Decision-Making **2018**. pp. 737–742. doi:10.1109/MILCOM.2018.8599853.
15. Farahani, S. *ZigBee Wireless Networks and Transceivers*; Newnes, 2011.
16. Ergen, S.C. ZigBee/IEEE 802.15. 4 Summary. *UC Berkeley, September* **2004**, *10*, 11.
17. Elahi, A.; Gschwender, A. *ZigBee Wireless Sensor and Control Network*; Pearson Educ.: London, U.K., 2009.
18. Norair, J. Introduction to DASH7 technologies. *Dash7 alliance low power RF technical overview* **2009**, pp. 1–22.
19. Piromalis, D.; Arvanitis, K.; Sigrimis, N. DASH7 mode 2: A promising perspective for wireless agriculture. *IFAC Proceedings Volumes* **2013**, *46*, 127–132.
20. Ayoub, W.; Samhat, A.E.; Nouvel, F.; Mroue, M.; Prévotet, J.C. Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility. *IEEE Communications Surveys and Tutorials* **2019**, *21*, 1561–1581. doi:10.1109/COMST.2018.2877382.
21. Czyz, J.; Luckie, M.J.; Allman, M.; Bailey, M. Don't forget to lock the back door! A characterization of IPv6 network security policy. Proc. NDSS, 2016.
22. Lashkari, A.H.; Danesh, M.M.S.; Samadi, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i) **2009**. pp. 48–52.
23. Kohlios, C.P.; Hayajneh, T. A comprehensive attack flow model and security analysis for Wi-Fi and WPA3. *Electronics* **2018**, *7*, 284.
24. Alliance, W.F. Wi-Fi Alliance Official Website. https://www.wi-fi.org/. Accessed: September 2023.
25. Banerji, S.; Chowdhury, R.S. On IEEE 802.11: Wireless Lan Technology. *International Journal of Mobile Network Communications &Telematics* **2013**, *3*, 45–64. doi:10.5121/ijmnct.2013.3405.
26. Ezhilarasan, E.; Dinakaran, M. A review on mobile technologies: 3G, 4G and 5G. 2017 second international conference on recent trends and challenges in computational models (ICRTCCM). IEEE, 2017, pp. 369–373.
27. Akyildiz, I.F.; Gutierrez-Estevez, D.M.; Reyes, E.C. The evolution to 4G cellular systems: LTE-Advanced. *Physical communication* **2010**, *3*, 217–244.
28. Zhang, Y.; Årvidsson, A. Understanding the characteristics of cellular data traffic. Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, 2012, pp. 13–18.

29. Chettri, L.; Bera, R. A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet of Things Journal* **2019**, *7*, 16–32.
30. Zeqiri, R.; Idrizi, F.; Halimi, H. Comparison of Algorithms and Technologies 2G, 3G, 4G and 5G. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2019, pp. 1–4.
31. Shelby, Z.; Bormann, C. *6LoWPAN: The wireless embedded Internet*; John Wiley & Sons, 2011.
32. Mulligan, G. The 6LoWPAN architecture. Proceedings of the 4th workshop on Embedded networked sensors, 2007, pp. 78–82.
33. Mulligan, G. The 6LoWPAN architecture **2007**. pp. 78–82.
34. Baker, N. ZigBee and Bluetooth: Strengths and weaknesses for industrial applications. *Computing and Control Engineering* **2005**, *16*, 20–25.
35. Bisdikian, C. An overview of the Bluetooth wireless technology. *IEEE Communications Magazine* **2001**, *39*, 86–94. doi:10.1109/35.968817.
36. Tosi, J.; Taffoni, F.; Santacatterina, M.; Sannino, R.; Formica, D. Performance evaluation of bluetooth low energy: A systematic review. *Sensors* **2017**, *17*, 2898.
37. Group, B.S.I. Technology Overview. https://www.bluetooth.com/learn-about-bluetooth/tech-overview/. Accessed: September 2023.
38. Haxhibeqiri, J.; De Poorter, E.; Moerman, I.; Hoebeke, J. A survey of LoRaWAN for IoT: From technology to application. *Sensors* **2018**, *18*, 3995.
39. Khutsoane, O.; Isong, B.; Abu-Mahfouz, A.M. IoT devices and applications based on LoRa/LoRaWAN. IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2017, pp. 6107–6112.
40. Lavric, A.; Petrariu, A.I.; Popa, V. Long range sigfox communication protocol scalability analysis under large-scale, high-density conditions. *IEEE Access* **2019**, *7*, 35816–35825.
41. Fourtet, C.; Ponsard, B. An introduction to Sigfox radio system. In *LPWAN Technologies for IoT and M2M Applications*; Elsevier, 2020; pp. 103–118.
42. Sigfox. Sigfox 0G Technology. https://www.sigfox.com/, 2023. Accessed: May 2023.
43. Ratasuk, R.; Vejlgaard, B.; Mangalvedhe, N.; Ghosh, A. NB-IoT system for M2M communication **2016**. pp. 1–5. doi:10.1109/WCNC.2016.7564708.
44. Coskun, V.; Ok, K.; Ozdenizci, B. *Near field communication (NFC): From theory to practice*; John Wiley & Sons, 2011.
45. Coskun, V.; Ozdenizci, B.; Ok, K. A survey on near field communication (NFC) technology. *Wireless personal communications* **2013**, *71*, 2259–2294.
46. Danbatta, S.J.; Varol, A. Comparison of Zigbee, Z-Wave, Wi-Fi, and bluetooth wireless technologies used in home automation. 2019 7th International Symposium on Digital Forensics and Security (ISDFS). IEEE, 2019, pp. 1–5.
47. Alliance, Z.W. Z-Wave Official Website. https://www.z-wave.com/. Accessed: September 2023.
48. Bhavya, R.; Lokesh, M. A Survey on Li-Fi Technology. *An International Journal of Engineering & Technology* **2016**, *3*.
49. Haas, H.; Yin, L.; Wang, Y.; Chen, C. What is LiFi? *Journal of Lightwave Technology* **2016**, *34*, 1533–1544. doi:10.1109/JLT.2015.2510021.
50. LiFi.co. How LiFi Works. https://lifi.co/how-lifi-works/, 2023. Accessed: May 2023.
51. Zhuang, W.; Shen, X.; Bi, Q. Ultra-wideband wireless communications. *Wireless communications and mobile computing* **2003**, *3*, 663–685.
52. Hirt, W. Ultra-wideband radio technology: overview and future research. *Computer Communications* **2003**, *26*, 46–52.
53. Aiello, G.; Rogerson, G. Ultra-wideband wireless systems. *IEEE Microwave Magazine* **2003**, *4*, 36–47. doi:10.1109/MMW.2003.1201597.
54. Naik, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP **2017**. pp. 1–7. doi:10.1109/SysEng.2017.8088251.
55. Fernandes, J.L.; Lopes, I.C.; Rodrigues, J.J.; Ullah, S. Performance evaluation of RESTful web services and AMQP protocol. 2013 Fifth international conference on ubiquitous and future networks (ICUFN). IEEE, 2013, pp. 810–815.

56. Betzler, A.; Gomez, C.; Demirkol, I.; Paradells, J. CoAP congestion control for the internet of things. *IEEE Communications Magazine* **2016**, *54*, 154–160.
57. Chen, Y.; Kunz, T. Performance evaluation of IoT protocols under a constrained wireless access network. 2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT), 2016, pp. 1–7. doi:10.1109/MoWNet.2016.7496622.
58. (OMG), O.M.G. The real-time publish-subscribe wire protocol DDS interoperability wire protocol specification. *OMG, Version 2.2* **2014**.
59. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Communications Surveys and Tutorials* **2019**, *21*. doi:10.1109/COMST.2019.2910750.
60. Haataja, K.; Toivanen, P. Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures. *IEEE Transactions on Wireless Communications* **2010**, *9*, 384–392. doi:10.1109/TWC.2010.01.090935.
61. Rahman, M.T.; Shi, Q.; Tajik, S.; Shen, H.; Woodard, D.L.; Tehranipoor, M.; Asadizanjani, N. Physical Inspection & Attacks: New Frontier in Hardware Security. 2018 IEEE 3rd International Verification and Security Workshop (IVSW), 2018, pp. 93–102. doi:10.1109/IVSW.2018.8494856.
62. Wurm, J.; Hoang, K.; Arias, O.; Sadeghi, A.R.; Jin, Y. Security analysis on consumer and industrial IoT devices. Proc. 21st Asia South Pac. Design Autom. Conf. (ASP-DAC), 2016, pp. 519–524.
63. Bou-Harb, E.; Fachkha, C.; Pourzandi, M.; Debbabi, M.; Assi, C. Communication security for smart grid distribution networks. *IEEE Commun. Mag.* **2013**, *51*, 42–49.
64. Koley, S.; Ghosal, P. Addressing Hardware Security Challenges in Internet of Things: Recent Trends and Possible Solutions. 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015, pp. 517–520. doi:10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.105.
65. Pan, Z.; Mishra, P. Design of AI Trojans for Evading Machine Learning-based Detection of Hardware Trojans. 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2022, pp. 682–687. doi:10.23919/DATE54114.2022.9774654.
66. Gaydos, M.G.; Wallace, N.L.; Brown, R.G. Reverse Engineering and Embedded Processor Analysis. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.
67. Tushir, B.; Dalal, Y.; Dezfouli, B.; Liu, Y. A Quantitative Study of DDoS and E-DDoS Attacks on WiFi Smart Home Devices. *IEEE Internet of Things Journal* **2021**, *8*, 6282–6292. doi:10.1109/JIOT.2020.3026023.
68. Ashfaq, M.F.; Malik, M.; Fatima, U.; Shahzad, M.K. Classification of IoT based DDoS Attack using Machine Learning Techniques. 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), 2022, pp. 1–6. doi:10.1109/IMCOM53663.2022.9721740.
69. Alyami, M.; Alharbi, I.; Zou, C.; Solihin, Y.; Ackerman, K. WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic. 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022, pp. 385–392. doi:10.1109/CCNC49033.2022.9700674.
70. Sinha, S.; G, K. Network layer DoS Attack on IoT System and location identification of the attacker. 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 22–27. doi:10.1109/ICIRCA51532.2021.9545071.
71. Kumavat, K.S.; Gomes, J. Performance Evaluation of IoT-enabled WSN system With and Without DDoS Attack. 2023 International Conference for Advancement in Technology (ICONAT), 2023, pp. 1–5. doi:10.1109/ICONAT57137.2023.10080467.
72. Siddiqui, M.N.; Malik, K.R.; Malik, T.S. Performance Analysis of Blackhole and Wormhole Attack in MANET Based IoT. 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), 2021, pp. 1–8. doi:10.1109/ICoDT252288.2021.9441515.
73. Tatar, E.E.; Dener, M. Wormhole Attacks in IoT Based Networks. 2021 6th International Conference on Computer Science and Engineering (UBMK), 2021, pp. 478–482. doi:10.1109/UBMK52708.2021.9558996.
74. Verma, M.K.; Dwivedi, R.K. A Survey on Wormhole Attack Detection and Prevention Techniques in Wireless Sensor Networks. 2020 International Conference on Electrical and Electronics Engineering (ICE3), 2020, pp. 326–331. doi:10.1109/ICE348803.2020.9122944.

75. Ali, S.; Khan, M.A.; Ahmad, J.; Malik, A.W.; ur Rehman, A. Detection and prevention of Black Hole Attacks in IOT & WSN. 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), 2018, pp. 217–226. doi:10.1109/FMEC.2018.8364068.
76. Fu, C.; Zeng, Q.; Chi, H.; Du, X.; Valluru, S.L. IoT Phantom-Delay Attacks: Demystifying and Exploiting IoT Timeout Behaviors. 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2022, pp. 428–440. doi:10.1109/DSN53405.2022.00050.
77. Fu, C.; Zeng, Q.; Chi, H.; Du, X.; Valluru, S.L. Iot phantom-delay attacks: Demystifying and exploiting iot timeout behaviors. 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2022, pp. 428–440.
78. Nassi, B.; Nassi, D.; Ben-Netanel, R.; Mirsky, Y.; Drokin, O.; Elovici, Y. Phantom of the adas: Phantom attacks on driver-assistance systems. *Cryptology ePrint Archive* **2020**.
79. Whalen, S. An introduction to arp spoofing. *Node99 [Online Document]* **2001**, *563*.
80. Kim, D.W.; Jang, H.Y.; Kim, K.W.; Shin, Y.; Park, S.H. Design characteristics of studies reporting the performance of artificial intelligence algorithms for diagnostic analysis of medical images: results from recently published papers. *Korean journal of radiology* **2019**, *20*, 405–410.
81. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Communications Surveys & Tutorials* **2020**, *22*, 1686–1721. doi:10.1109/COMST.2020.2986444.
82. Abu Al-Haija, Q.; Zein-Sabatto, S. An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks. *Electronics* **2020**, *9*, 2152.
83. Zhang, J.; Pan, L.; Han, Q.L.; Chen, C.; Wen, S.; Xiang, Y. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica* **2021**, *9*, 377–391.
84. Handa, A.; Sharma, A.; Shukla, S.K. Machine learning in cybersecurity: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2019**, *9*, e1306.
85. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys & Tutorials* **2020**, *22*, 1646–1685. doi:10.1109/COMST.2020.2988293.
86. Zohuri, B.; Moghaddam, M. Deep learning limitations and flaws. *Mod. Approaches Mater. Sci* **2020**, *2*, 241–250.
87. Raji, I.D.; Kumar, I.E.; Horowitz, A.; Selbst, A. The fallacy of AI functionality. Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 959–972.
88. Hou, L.; Zhao, S.; Xiong, X.; Zheng, K.; Chatzimisios, P.; Hossain, M.S.; Xiang, W. Internet of things cloud: Architecture and implementation. *IEEE Communications Magazine* **2016**, *54*, 32–39.
89. Lin, Y.; Shao, L.; Zhu, Z.; Wang, Q.; Sabhikhi, R.K. Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development* **2010**, *54*, 4–1.
90. Varia, J. Cloud architectures. *White Paper of Amazon, jineshvaria. s3. amazonaws. com/public/cloudarchitectures-varia. pdf* **2008**, *16*.
91. Wilder, B. *Cloud architecture patterns: using microsoft azure*; " O'Reilly Media, Inc.", 2012.
92. Wan, J.; Yang, J.; Wang, Z.; Hua, Q. Artificial intelligence for cloud-assisted smart factory. *IEEE Access* **2018**, *6*, 55419–55430.
93. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things. *IEEE Internet of Things Journal* **2021**, *8*, 13849–13875. doi:10.1109/JIOT.2021.3088875.
94. Mohanta, B.K.; Jena, D.; Satapathy, U.; Patnaik, S. Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of Things* **2020**, *11*, 100227.
95. Ghosh, A.; Chakraborty, D.; Law, A. Artificial intelligence in Internet of things. *CAAI Transactions on Intelligence Technology* **2018**, *3*, 208–218.