ISSN 2664-4150 (Print) & ISSN 2664-794X (Online)
South Asian Research Journal of Engineering and Technology
Abbreviated Key Title: South Asian Res J Eng Tech

| Volume-4 | Issue-4 | July-Aug- 2022 | **DOI: 10.36346/sarjet.2022.v04i04.003**

Review Article

A Comprehensive Survey of Internet of Things Applications, Threats, and Security Issues

Y. Harshavardhan Reddy[1*], Adnan Ali[1], P. Vinay Kumar[1], M. Hari Srinivas[1], K. Netra[1], V. Jeswanth Achari[1], R. Varaprasad[1]

[1]G. Pullaiah Collage of Engineering and Technology, Kurnool - 518002, Andhra Pradesh, India

***Corresponding Author:** Y. Harshavardhan Reddy
G. Pullaiah Collage of Engineering and Technology, Kurnool - 518002, Andhra Pradesh, India

Article History
Received: 19.06.2022
Accepted: 30.07.2022
Published: 03.08.2022

Abstract: The Internet of Things (IoT) is a novel way of providing a wide range of solutions. The IoT includes a variety of small smart devices. Their use, size, power, and computation power distinguish them from other devices. Due to the fact that the most of Internet technologies and communication protocols have not been intended to function with the IoT, there are numerous security concerns. The digitalization of the Internet of Things has also raised concerns about safety, privacy, cyber-attacks, and organised crime of the IoT has also prompted public safety concerns about privacy, cyber-attacks, and organised crime. This paper aims to include a full overview of edge-side security threats and countermeasures for people who want to learn more about security and add value to its improvement. This layer is divided into three levels: We'll begin by discussing three prominent IoT reference models, as well as security. Second, we will discuss the promising applications of the IoT as well as the motivations of hackers who seek to take advantage of this new technology. Third, we'll take a look at some of the most serious threats. Fourth, we discuss possible countermeasures.

Keywords: Internet of Things (IoT), Attacks, Countermeasures.

I. INTRODUCTION

There isn't a single definition for the IoT. To provide any service, the IoT allows humans, devices, and other devices to communicate with one another [1, 2]. The "broad definition" of the IoT is a way to connect things that aren't alike, such as people, sensors, and other things that may need or offer a service [3].

A new paradigm known as the IoT has emerged as the most intriguing developments to occur throughout the last decade. Because of the growth of several communication protocols, as well as the miniaturisation of transceivers, an isolated device is now able to communicate with another network. In addition, the processing power, energy and storage capacity of minimal computing and sensing devices have all increased, whereas their overall dimensions have shrunk by a significant margin. Due to recent advancements in electronics and computer science, it is now possible to create an ever-increasing number of sensors and computers that are connected to the Internet (commonly referred to as "smart devices").

As a result, the number of possible attacks on a thing or person's security or privacy has increased significantly as well. As a result, these security requirements are not widely known. As a result, it is imperative that security issues and prevalent privacy concerns be examined and investigated and addressed in greater detail. Smart devices that can automate buildings and keep tabs on a person's health would be a lot easier to make with this technology. Smart devices that can automate buildings and keep tabs on a person's health would be a lot easier to make with this technology [4].

Academic, industrial, and government researchers are increasingly focusing on IoT security as a research topic. The conceptualization and advancement of IoT-based systems is being undertaken by many companies and organisations around the world [5]. To create a large number of dependable solutions, developers must overcome numerous challenges. To be specific, they have to deal with issues related to security research. Many research projects are underway to identify security issues and devise countermeasures. In this survey, IoT security issues are broken down into their various levels of severity and how to deal with them. The paper's primary objective is to demonstrate to the reader what attacks have been made, how they have indeed been dealt with, and what vulnerabilities remain [6].

The rest of the paper is structured in this way. Section II, discusses the reference model. Section III, explores IoT applications. Section IV, discusses possible ways to attack the IoT. Section V, talks about how to safeguard from all these attacks. Section VI, sums up our findings and conclusion.

II. IoT Reference Models

Three IoT reference models have been extensively discussed in academic and business published papers. Figure 1, which depicts these models and their various levels, can be found at the bottom of the page. In the early days of IoT research, people came up with the three-level model [7]. Think of it as a larger version of the wireless sensor networks that have been around for decades (WSNs). As a result, it sees the IoT as a mix of wireless sensor networks and cloud servers, both of which provide a variety of services to customers. Using the five-level model [3], people from various parts of an organisation can more easily communicate with one another. It simplifies complicated systems by dividing them into smaller, more manageable applications with clearly defined components. CISCO's three-and five-level models were completely revamped by CISCO in 2014. Cisco's seven-level method has the ability to be widely adopted and standardised [8]. Data flows in both directions in this type of model. But the prevalent flow of data is derived by the use of data, not the other way around. Data and commands move from the applications level to the lower levels of a control system (edge node level).It's common for data and commands to flow upward in a monitoring scenario.

All three IoT reference models aren't covered in great detail in this paper. In this paper, we demonstrate IoT security attacks and countermeasures using the CISCO reference model. Next, we'll go over each level of this model in a brief discussion [9].

Level 1-Edge devices: Computing nodes of such as smart controllers, sensors, RFID tags, and other similar devices, as well as various types of RFID tags, typically make up the first level of this model. Starting here, data security and integrity must be considered from the ground up.

Level 2-Communication: Communication between the first and second levels of devices is possible. Devices in the second tier communicate with one another, with devices in the third tier, and so forth (edge computing level).

Level 3-Edge Computing: There are three levels to the model: the first two are cloud computing, and the third is edge computing. It's where the model begins and where simple data processing begins. This is critical if the higher levels are to avoid having to do a lot of work. Many real-time applications must run as quickly as possible at the network's edge. There are many factors that contribute to how much work this level does, including service providers, servers, and computing nodes. In this case, basic signal processing and algorithms are typically employed.

Level 4-Data Accumulation: The majority of applications do not necessitate immediate data processing. If the data is moving, it can be stored for future analysis or shared with other computers at a higher level, which is what this level does. Converting data over the network packets into database tables and filter and store data only relevant to higher levels are the primary functions of this level.

Level 5-Data abstraction: At this level, data can be displayed and stored, making it easier and more efficient to do more with it in the future. They perform tasks such as normalising and deformalizing data. Indexing data, consolidating it into a single location, and making it easy for people to access different data stores are also necessary.

Level 6-Applications: Data is interpreted at the application level, and software works with the data accumulation and data abstraction levels. To be sure, there are numerous uses for the IoT across a wide range of industries and markets.

Level 7- Collaboration & Processes: It is the top level of the IoT when the people who use it are there. It's up to people to use the apps and the analytical data that comes with them.

III. Applications

The IoT has a wide range of applications. Few examples include smarter buildings and homes as well as electronic health aids or smart cars. Smart devices may be able to do a variety of things to help people understand what they are seeing. However, we have only scratched the surface of what the IoT can do [10]. There are a lot of different services you can use with the IoT because it combines sensors with communication networks, authentication, identification, and computing. At any given time, you can learn about anything smart. Figure 2 depicts the IoT's applications.

1. Wearables

As you can see, there are many different types of wearable devices out there. Virtual glasses, heart rate monitor bands, and GPS belts are just a few examples. Our daily lives are enhanced by the Internet of Things, thanks to companies like Google, Apple, Samsung, and a slew of other names in the technology sector. Sensors, hardware, and software are all included in these small and energy-efficient devices. They can be used to make measurements and reads, and they can accumulate and organise data about their users.

2. Smart vehicles

Cars that can drive themselves have begun to change people's daily routines. A small IoT-based system can be used to unlock and lock cars remotely, download maps, and gather traffic information. Vehicle tracking and management is greatly facilitated by the use of RFID (Radio Frequency Identification) and AIDC sensors. On the basis of microchips, RFID uses radio waves to locate objects. These tags can provide information on a vehicle's make, model, destination, and speed. The use of an intelligent traffic management system can be advantageous to those who are connected to the internet. RFID can help you locate missing vehicles. Real-time tracking of automobile traffic. Cars are better suited to transmitting signals.

3. Health

This technology enables doctors to keep track of patients beyond the hospital and in timely manner, using wearables that are hooked. When doctors use the IoT to monitor patients' vital signs, they are able to better care for them and help prevent lethal events by sending them alerts. Using IoT technology in hospital beds, which are now referred to as "smart beds," is another example. They have sensors that can read vital signs like blood pressure, oximetry and body temperature as well as a variety of other things [11].

4. Buildings

It's easier to conserve energy when you have a smart building or home. With sensors and data-analytic algorithms built in, smart thermostats are able to control air conditioners based on what people like and how they usually act. Using a smart controller, the lighting can be adjusted based on how the user is interacting with it. Many appliances in the home, such as refrigerators, televisions, and home security systems, may be equipped with their own processing units and be able to access the Internet. Using smart devices makes it easier for people to accomplish their goals and objectives. Remote-controllable devices respond to user commands by carrying out actions that have an impact on the environment. Consequently, physical harm can result from attacks on these devices [12].

5. Water Supply

An Internet-connected sensor, either built into or attached to water metres, collects, processes, and analyses data. This enables the service provider to better understand how people use water, find problems with the service, and report the results and suggestions for what to do going forward. Finally, customers can see their own consumption data on a web page in real time and get alerts when their consumption deviates from their average usage record, that could be an indication of a leak.

6. Smart Grid and Energy Saving

Smart IoT-based systems with integrated sensors and actuators can be used to save energy. Controllable remotely from smart TVs to power outlets and refrigerators are expected to share data with energy providers to make smart homes more energy efficient. These devices can also be controlled remotely and configured to save a lot of energy. Smart metres (metres with sensors) are becoming more common. Sensors are strategically placed in manufacturing and distribution facilities. This improves the electrical network's monitoring and control. Communication between a service provider and a customer can yield a wealth of useful data. Issues, decisions, and solutions can all be determined by this data. Using this app, users can learn how much energy they use and how to reduce or change it [10].

7. Hospitality

Things get interesting in the hotel industry when IoT is used. Many processes can be automated by sending electronic keys to each guest's mobile device. To put it another way, this means that things like sending information about things that might be of interest to guests and making orders for room service can all be done easily through integrated applications that use the Internet of Things. Fast and simple check-out is made possible by electronic keys. Electronic keys prevent doors from being opened, show which rooms are available, and assign housekeeping tasks to maintenance workers.

8. Environmental Monitoring

It's possible to use smart devices with built-in sensors to monitor the environment and look for emergencies like floods that require immediate action. Using IoT-enabled devices, air and water quality can also be monitored. It is also possible to monitor the humidity and temperature with ease [13].

9. Agriculture

It's possible to run a smart farm. The IoT provides farmers with a wealth of information about their soils. The quality of the soil is essential to the success of a farm's crops. The use of IoT sensors can provide a wealth of information about soil health and development stages. There are a lot of chemical information that can help farmers control irrigation, improve water use efficiency, and even identify diseases in plants and the soil.

10. Fleet management

It is possible to connect fleet vehicles, managers, and drivers more efficiently with the help of sensors installed in each vehicle. The software that collects, processes, and organises data can be accessed by the driver and manager/owner to learn more about the vehicle's status, operation, and requirements. Without the driver's knowledge, get real-time maintenance alerts. Geolocation, performance analysis, telemetry control, fuel savings, reduction of polluting emissions, and even driver improvement are some of the benefits of the IoT in fleet management.

11. Production and Assembly Line Supervision

IoT-based systems enable quick product development and interactive demand response by allowing communication between sensors and control systems [14]. The smart management of real-time measurements, in addition to energy conservation and safety measures.

12. Traffic Monitoring

Smart cities can benefit from the Internet of Things because it can help with traffic management. Using apps like Waze or Google Maps, we can use our phones as sensors to collect and share data from our cars. This helps the IoT by showing different routes and giving and enhancing information on the same destination, distance, and estimated arrival time. This helps the Internet of Things as well.

13. Food Supply Chain

The application is intricate and decentralised. The Internet of Things can provide chain managers with valuable data. While the IoT is already being used in supply chain management, the perks are still very small. A major advantage of IoT in supply chain management is improved product security [15]. These devices can alert you to any level of unauthorised access to the supply management system.

14. Maintenance Management

There are many IoT applications, but maintenance management is among the most commonly used ones. It is possible to prolong the service life of tangible assets also while ensuring their availability and reliability with the help of sensors and CMMS/EAM maintenance management software. Applications for sensor data processing and organisation software developed for real-world assets are virtually endless. For example, using AI algorithms like Machine Learning(ML) or Deep Learning(DL) to predict failures before they occur is possible with this technology.

IV. Attackers and Their Motives

An IoT-based system could be utilised for a variety of purposes, including factory management and health monitoring. As a result, the IoT framework has become an attractive choice for a wide variety of attackers. Among these are casual hackers, government hackers, cyber-criminals, and so on.

It's possible that an attacker could target IoT devices in order to steal credit card information, location information, financial account information, and health related information. To attack a third party, they may attempt to reconcile IoT hardware such as edge nodes. Suppose a spy agency infects millions of IoT devices (like remote monitoring systems) and smart TVs, for example (like smart TVs). It can spy on targets or launch large-scale attacks using infected systems and devices. Additionally, hacktivists or the organization's detractors may picket smart devices as a form of protest.

V. An IoT-Centric View of Security

Secure thing and security attack are two of the most frequently used IoT terms when classifying what is secure, it is critical to understand the aspects of security. Confidentiality, integrity, and availability are the three pillars of the CIA's security model. It is the practise of restricting unauthorised access to information in order to keep it private. Medical records and prescriptions are examples of sensitive data that must be protected by IoT devices.

Health information can be leaked or even lives are put in danger if unauthorised access to personal health devices is allowed [16]. Integrity is a prerequisite for long-term reliability. All commands and data collected must be verified by the device. There are serious consequences to a breach of one's moral compass. attacks on the integrity of medical devices, such as an insulin pump or a pacemaker, can have fatal consequences [17, 18]. The IoT(IoT) must be available for a fully functioning Internet-connected environment. As a result, data collection doesn't cause any

disruptions to the service. The CIA-security inadequacy triad has been previously discussed [19-21]. New threats aren't being addressed by the CIA-triad, Various sources in the fields of information assurance and security were examined in order to compile this definitive list. The IAS-octave is being suggested as an implication of the CIA triad. Security requirements, definitions and abbreviations for IAS octave are summarised in Table 1. Continue reading for more on IAS-octave requirements [22].

TABLE I: Security requirements

Requirement	Definition	Abbreviations
Confidentiality	Ensuring that only authorized users access the information	C
Integrity	Ensuring completeness, accuracy, and absence of unauthorized data manipulation	I
Availability	Ensuring that all system services are available, when requested by an authorized user	A
Accountability	An ability of a system to hold users responsible for their actions	AC
Auditability	An ability of a system to conduct persistent monitoring of all actions	AU
Trustworthiness	An ability of a system to verify identity and establish trust in a third party	TW
Non-repudiation	An ability of a system to confirm occurrence/non-occurrence of an action	NR
Privacy	Ensuring that the system obeys privacy policies and enabling individuals to control their personal information	P

VI. IoT Security Vulnerabilities

Edge-side layer attacks and vulnerabilities are discussed in detail in this section.

A. Edge nodes:

This section discusses in detail different attacks first level of the reference model, which consists of computing nodes and RFID tags.

1) Edge Nodes: Initial targets include devices such as RFID readers, sensors, and small-footprint controllers.
Hardware Trojan: IC security is threatened by hardware Trojans [23-27]. Hardware In order to gain access to data or software, an attacker can use maliciously modified integrated circuits (Trojans). Before or during fabrication, hardware trojans can be installed by altering the design and specifying how they are activated. In order for a Trojan to be activated from within the integrated circuit, it must first be triggered by an external signal, such as a countdown circuitry, or it will remain dormant.

Non-Network Side-Channel Attacks

Even if nodes are not communicating wirelessly, they can still provide valuable information. As an example, the node's electromagnetic (EM) signature can reveal important information about its health. Recent publications of EM-based attacks [28-30] have started to develop the idea of nonnetwork side-channel threats. Acoustic and electromagnetic (EM) signals from medical devices, for instance, have been shown in recent research to be useful in providing information about the patient or device. According to that work, detecting known signals or protocols may put the user's safety at risk, especially if the device is expensive. Medical systems could also be affected by this attack.

As an example, think of someone with an extremely stigmatised medical condition who has a medical device on their body that indicates it. With this device, patients may feel humiliated. Side-channel data, such as glucose levels and blood pressure, can also be retrieved from the devices to assess the health of an individual [31].

Denial of Service (DoS) attacks: There are three different types of DoS attacks that can be used against edge computing nodes: battery draining, sleep deprivation, and outage attacks.
1. Battery Draining: Due to their small size, nodes typically carry low-capacity batteries. Battery-draining attacks can indirectly lead to node failures or failures to report emergencies in an emergency situation.. An attacker who can deplete a smoke detector's battery can disable the fire detection system [32]. If recharging is difficult, they can devastate a network. As a result of bombarding a node with packets, an attacker can drain its battery [33]. According to the literature, there are a number of battery draining attacks [34-36].

2. Sleep Deprivation: A battery-powered node with limited energy capacity is the unfortunate victim of a sleep deprivation attack. Requests that appear to be legitimate are sent by the attacker in this attack. Because it drains the battery in a more complex manner, it's more difficult to detect this attack. The idea of sleep deprivation[37], it was ground-breaking. One of the first studies to look at the impact of sleep deprivation attacks on energy-constrained devices.

3. Outage Attacks: For an edge node to go down, it must stop functioning normally. An administrator device or a group of devices may go down from time to time. It's possible for a node to go down because of a manufacturing error or battery drain or sleep deprivation or code injection. One example is the Stuxnet [38] attack on Iran's nuclear process control programme. To avoid detection of abnormal behaviour, Stuxnet alters industrial process control sensor signals. This means that even in the event of an emergency [39], the system will continue to function.

Physical Attacks/Tampering: This makes edge devices vulnerable to both hardware and software attacks. With physical access to the device, an attacker has the ability to extract sensitive cryptographic data, alter the software and/or reinstall the OS [40-42]. Edge nodes can be permanently destroyed by physical attacks. It's because of this that they are designed to extract data for future use, like the fixed shared key. A hacker recently attempted to replace the Nest thermostat's default firmware with a malicious one. It doesn't matter if the attacker no longer has physical access to the thermostat; this attack will still work for attacker.

Node Replication Attacks: A malicious node is added to an existing set of nodes by copying the identification number of another node. These attacks have a significant impact on network speed. In addition, an attacker can easily corrupt or redirect packets to the replica. Attackers can extract cryptographic shared keys from the system as a result of this attack. A node's authorization can be revoked [43-45] with the help of these tools [45].

Camouflage: An attacker can remain undetected at the edge by launching an attack on an authorised node or by creating a fictitious edge node. The modified/counterfeit node must first be modified in order to obtain, process, send, or redirect packets [44]. Additionally, this node is only able to monitor traffic in passive mode [46].

Corrupted/Malicious Node: Unauthorized access to a network is the goal of corrupting nodes. It is possible for nodes infected with a virus to gain access to other nodes and take command of the network [44]. A malicious node can be used by an attacker to insert false data into the system or to block the delivery of true messages [47].

2. RFID tags: We talk about the attacks on RFID tags here.
Tracking: A serious danger arises from the reading of RFID tags without the proper authorization. Unfortunately, almost none of them do. Bystanders can easily read tags attached to products or individuals if they are close enough. A lot of tracking information [48]. It is possible for an attacker to use a large number of RFID readers to read these fixed identifiers Personal information (e.g. credit/loyalty card number) and personal profile (e.g. tag identifier) can be stored in the cloud [49].

Inventorying: Some tags provide useful information about the items they are attached to. Manufacturer codes and product codes are two custom fields on EPC tags. A tag reader can look at the products of anyone who has an EPC tag, so they are subject to inventorying [50]. This privacy concern is raised by this threat. Medical devices (like insulin pumps) can help an assailant determine a patient's condition, for example (like diabetes).

Physical Attacks/Tampering: Having physical access to a tag gives an attacker the ability to carry out this attack. A lab is used to physically manipulate the tags [51]. The physical destruction of RFIDs is not unknown. A probe attack is one such example [52]. Using these attacks, data can be extracted or fraudulently modified from tags.

Tag Cloning: Hackers can make a lot of money by selling cloned RFID tags and impersonating RFID tags. Potential harm can be magnified by automation [53]. To gain entry to secure areas, such as a bank account or other private data. The alteration of an item's identity is at the heart of counterfeiting, which is typically accomplished through the manipulation of tags. As compared to spoofing attacks, counterfeiting attacks require a smaller amount of information. Partially manipulating a tag is the goal of these attacks. Commercial proximity cards can be read and partially simulated using an RF tape recorder, circumventing building security systems [54].

Counterfeiting: Tampering with tags is the most common method of altering an item's identity in a counterfeiting attack. As compared to spoofing attacks, counterfeiting attacks require a smaller amount of information. Partially manipulating a tag is the goal of these attacks. Commercial proximity cards can be read and partially simulated using an RF tape recorder, circumventing building security systems [54].

DoS Attacks: The tags can't be read by tag readers because of interference with the RF channels, which prevents the RFID services from working as intended. The jamming of all RFID-based doors can render a building inoperable. Additional RFID DoS vulnerabilities are discussed in [55].

Eavesdropping: Intercept, read and save messages for later analysis are the attacker's goals. Intercepted data can be used in other attacks, such as tag cloning. RFID eavesdropping is not a new concept, and it has been discussed in the literature

before. Eavesdropping in the RFID environment has been brought up in recent NIST and DHS reports [56], as well as surveys [57, 58]. A variety of realistic attack scenarios and the experimental setups that were used to test them.

Side-Channel Attacks: Even encrypted messages can be intercepted and processed using modern tools to extract data from various patterns. The tags at the building's entrance, for example, can be used by an attacker to determine how many communications are present in the structure. Over-the-air timing attacks against RFID tags have not been tested for effectiveness [57]. It was reported that Carluccio et al. used electromagnetic waves to attack RFIDs [59].

B. Communication

Eavesdropping: Eavesdropping (also known as sniffing) [60] is the deliberate listening to private communications over communication links. An attacker can gain access to sensitive data if it is left unprotected. Passwords and usernames can be easily retrieved in this situation. There are times when eavesdropping on packets containing access control information like network passwords and node identification numbers can be useful. The thief can use the information he or she has gathered to launch additional attacks. As long as the hacker can get their hands on the information needed to add an unauthorised node, they can quickly and easily get their hands on a malicious node [61].

Side-channel attacks: Side-channel attacks against encryption are effective despite their difficulty in execution. They put the integrity of cryptographic systems in jeopardy. Edge node side-channel attacks are possible, as previously stated. A communication side-channel attack is usually non-invasive, unlike an attack on an edge node. They only use data that has been accidentally released. Examples of unintentional information leakage include the time interval between two consecutive packets, the frequency band, and communication modulation. Only reducing leakage or adding noise to leaked data can protect against non-invasive attacks, which are undetectable.

DoS attacks: Radio signal jamming is the most well-known DoS attack at the communication level. An active jamming attack is either continuous or intermittent (also known as noncontinuous), and the latter allows nodes to send and receive packets on an as-needed schedule. Intermittent jamming slows down time-sensitive systems, whereas constant jamming stops all transmissions. Fire alarm systems that detect unusual changes in gas levels and notify the fire department are one possibility. The reliability of the system can be severely compromised if communication between nodes and the base station is disrupted. The system will be brought to a halt if the attacker employs continuous jamming.
DoS attacks against Bluetooth [62], ZigBee [63], and 6LowPan [64] have been examined in several studies. Additionally, an attacker may use malicious nodes or routers to disrupt communication. An accidental protocol violation can result in collisions or jamming of communication channels. It is possible for a malicious router or node to either block or redirect messages [65]. It can be done on a regular basis or on a constant basis. Continuous DoS attacks can be easily detected, but intermittent DoS attacks necessitate precise and efficient monitoring.

Injecting fraudulent packets: A malicious packet can be inserted, manipulated, or replicated (also known as replayed) into communication links by an attacker. The attacker inserts new packets into the network in insertion attacks. An insertion attack, on the other hand, is capable of creating and sending malicious packets that appear to be genuine. It is possible to capture an entire packet and send it back with a different set of contents after modifying the header, checksum, and data inside. In a replication attack, the attacker captures and replays packets that have already been exchanged between two targets. Due to the lack of a history of previous packets or states, a stateless system is vulnerable to replay attacks [66].

Routing Attacks: Routing attacks [67] are attacks that change how messages are sent. If an attacker wants to fool, redirect, misdirect, or drop packets at the communication level, they can do so with these kinds of tricks. In the simplest kind of routing attack, the attacker changes the routing information. For example, he or she could make routing loops or send false error messages. Altering attacks aren't the only serious ones that have been proposed. Black Hole, Gray Hole, Wormhole, Hello Flood, and Sybil [68] are some of the other attacks that have been proposed. These are just a few of them. We'll talk about them a little more next.

1) **Black Hole:** Black Hole attacks utilise a malicious node that falsely claims to have the quickest route to the target network [2, 9]. Packets are received by all nodes, which can process or discard them.
2) **Gray Hole:** A Gray Hole attack [9] is a variant of a Black Hole attack in which nodes drop some packets but not others.
3) **Worm Hole:** Even if all communications are genuine and confidential, a devastating attack known as a "Worm Hole" can still occur. Attackers can record packets in one place on the network and then tunnel them to another.
4) **Hello Flood:** A node that broadcasts "HELLO PACKETS" to its neighbours is the foundation of a Hello Flood attack [69]. It is possible for receiving nodes to incorrectly assume that they are within communication range of the sender. Sending "HELLO PACKETS" to every node in the network, the malicious node pretends to be their neighbour with high transmission power.

5) Sybil: Sybil attacks involve the introduction or use of fictitious nodes, called Sybil nodes, into the victim's network by the attacker [68]. Sybil nodes in the system have the ability to outvote honest nodes.

Unauthorized Conversation

Every edge node must communicate with other nodes in order to share or access data. However, each node should only communicate with those who need its data. In systems with both insecure and secure nodes, this is a must-have feature. A smoke detector's data is required by a smart home's thermostat to turn off the heat in an emergency. An attacker may be able to take over the entire home automation system by hacking the insecure smoke detector [70].

C. Edge Computing Level

Edge (Fog) computing is a relatively new concept. This means that the flaws in the system have not been uncovered yet. For edge computing attacks, there are only a few reports focusing on threats to sensor networks [71, 72]. Attacks on an edge computing system are described below. Some of these attacks are applicable to edge computing systems as well as to traditional systems and networks.

Malicious Injection

Inadequate input validation makes it possible for malicious input to be introduced into a system. This means that service providers may act on the attacker's behalf if they receive malicious input. In order to attack the servers, a malicious component could be added to a lower level (such as a communication or edge node). Because of this, the attacker may be able gain access to confidential data or bypass security measures. Consequently. Database error messages can also be of assistance to an attacker. If the attacker isn't familiar with the database's tables, forcing an exception may reveal more information about the table and its fields [73].

Integrity Attacks Against Machine Learning

It is possible to launch attacks against IoT machine learning methods that are causal or exploratory in nature. To exploit a vulnerability, an attacker must change the training process (causative attacks) or manipulate the training dataset (exploratory attacks). In recent studies, a new cause of attack known as poisoning has been discovered [74-76]. The term "poisoning" refers to the practise of introducing erroneous data points to a dataset used for training purposes. An attacker could launch this attack on the learning algorithm by directly accessing the server or computing nodes, or by adding enough malicious nodes to lower levels of the IoT model. By altering the dataset, the classification algorithm is prevented from building a reliable model as a result of this manipulation.

Side-Channel Attacks

Side-channel attacks on the edge node and communication components were discussed earlier in this article. It is possible for an attacker to launch side-channel attacks by obtaining information from other components (like service providers and servers). A service that generates verbose error warnings can be useful to designers and developers. Excessive data can be provided by the same warnings in operational settings [77].

Non-Standard Frameworks and Inadequate Testing

The privacy and security of your data can be jeopardised by bugs in the code. Nodes typically connect to intermediate servers, which means that a compromise could have amplification effects as well. In order to create an edge computing system, it is necessary to combine resources and devices from various manufacturers [78]. There is also a lack of a common framework and policy framework for the implementation of edge computing-based systems. As a result, many security and privacy flaws may go unnoticed.

Insufficient/Inessential Logging

An intrusion or hacking attempt can be detected with the help of logging. There should be a log of all authentication, authorization, and application errors. Edge computing systems may be harmed by insufficient logging [79]. The log files should also be encrypted.

V. COUNTERMEASURES

A. Solutions for security issues in edge nodes

Edge node security issues can be addressed with these solutions.

1) Computing nodes: We begin by addressing attacks on computing nodes.

Side-Channel Analysis: If a device has been infected with malware, it is possible to detect it using side-channel signal analysis.

1. Trojan Detection: Side-channel signals like timing [65], power [81], and spatial temperature are used to detect Trojans in the network. By altering wire and gate conductivity, as well as the silicon IC's heat distribution, a Trojan can

take over a circuit. A hardware Trojan can be detected by comparing the physical characteristics and/or heat distribution map of a suspect IC to a reference IC that is free of Trojans.
Using power-based analyses, the IC can be monitored for the presence of Trojans. To detect Trojans, timing-based methods use efficient delay tests that are sensitive to small changes in circuit delay along the affected paths [82]. Electronic components' temperatures can be mapped using infrared imaging techniques. Infrared imaging techniques can map thermal infrared emissions due to silicon's infrared transparency [83].

2. Malicious Firmware/Software Detection: Side-channel signal analysis has been shown to be effective in detecting malicious firmware and software in previous studies [84, 85]. Side-channel signals, as discussed earlier in Section IV, can provide valuable information about a device. To detect malware-induced abnormal device behaviour, such as an increase in power consumption, side-channel signals can be processed in a manner similar to how Trojans are detected.

Trojan Activation: Partially or fully activating the Trojan circuitry in order to aid detection is the goal of these strategies. Activation methods for Trojans have recently been proposed [86, 87]. In order to detect and magnify the differences between a Trojan-free circuit's behaviour, outputs, and side-channel leakages, such strategies are employed. While increasing Trojan detection, Chakraborty et al. proposed MERO [88] to reduce testing time and costs. MERO is capable of increasing the sensitivity of side-channel Trojan detection. The basic idea is to find Trojans that can be activated by a small subset of these rare conditions.

Policy-Based Mechanisms and Intrusion Detection Systems (IDSs): IoT security and privacy concerns may be solved with policy-based approaches. An IDS is capable of continuously detecting policy violations. An IDS performs a general rule-checking function. It guards against battery drain and sleep deprivation attacks by detecting unusual requests to the node. It is now possible to design effective and efficient IDS for monitoring and detecting threats on edge nodes [89-93].

Circuit Modification: Changing the circuit can protect against physical, side-channel, and Trojan attacks. Specific circuit changes and modifications can be made to address/prevent each of these attacks.

1. **Tamper Proofing And Self-Destruction:** All nodes have the ability to withstand physical attacks. A variety of tamper-proofing methods have been proposed and used in home automation sensors, such as smoke detectors, for designing the physical packages of the nodes. Physical attacks [94] can be thwarted by employing self-destruction mechanisms as well.
2. **Minimizing Information Leakage:** Adding random delays, intentionally generated noise [95], balancing Hamming weights [96], using constant execution path code [96], improving cache architecture [97], and shielding are some well-known countermeasures.
3. **Integrating Physically Unclonable Function (PUF) into the circuity:** A PUF is a noisy function that is embedded in a programme [98]. Response y is generated by the PUF's intrinsic physical properties and the challenge x. As a result of their uniqueness, tamper-evident PUFs can be used to identify and authenticate a device and to detect Trojans [99, 100]. Physical layout changes can alter the parasitic parameters Trojans are able to detect.
4. **Securing Firmware Update:** Updates to the firmware can be initiated locally or remotely. To inform users of a remote firmware update, a command (CMD) is sent from the base or server. An advertisement (ADV) is then sent out to other nodes. Users who have received ADV can compare the new and old firmware versions [101], and if necessary, request (REQ) the new one. After all is said and done, the advertiser provides the requested information to those who have requested it. Authentication of CMD, ADV, REQ, and data packets is required for remote firmware updates. Every step of the protocol should take DoS attacks into account [102]. Firmware can be updated remotely or directly on some nodes. For example, a USB cable can be utilised. In this case, the integrity of the firmware and the identity of the updater should be checked. An attacker can replace legitimate device firmware with a malicious node if there are insufficient integrity check mechanisms.

2) RFID tags: solutions and ideas for preventing RFID tag attacks.
Kill/Sleep Command: RFID tags are designed to be destroyed. The PIN of an RFID tag is 32-bit. After receiving the correct PIN, the RFID reader can disable the tag [56]. The tags are put to sleep and become inactive for a period of time when a sleep command is issued [55]. This may appear to be an easy concept, but it requires advanced techniques to design and implement secure PIN management schemes.

Isolation: The privacy of tags is effectively protected by isolating them from all electromagnetic waves. Create and utilise isolation chambers. In most cases, the cost of constructing such rooms is prohibitive. Alternatively, a metal mesh container can be used. It is possible for a Faraday cage to block specific electromagnetic wave frequencies [103]. It is also possible to employ an active RF jammer, which continuously disrupts specific RF channels, as an additional strategy.

Cryptographic schemes: Three types of cryptographic schemes have been talked about a lot in the past literature to help safeguard RFID tags from being hacked.

1. **Encryption:** Full encryption usually requires a lot of hardware. It hasn't been possible to use it in RFID tags because the tags need to be cheap (a few cents). Feldhofer [111] came up with a way to make sure you are who you say you are based on the Advanced Encryption Standard (AES). However, for a standard implementation of AES, 20-30K gates are usually needed [112]. RFID tags, on the other hand, can only store a few bits and support 5-10K logic gates. The number of gates and the cost of the tag suggest that the security mechanism can only use 250-3500 gates. The traditional implementation of AES wasn't right until Jung et al. came up with a new one that only needs 3595 logical gates [113]. However, no fully developed version of AES has been used in any RFID tag yet.
2. **Hash-based schemes:** In order to address RFID security concerns, these methods are frequently employed. In [104-107], you will find the most recent hash function research. proposes a simple security mechanism based on hash values. Locked and unlocked tags both respond to queries with their hashed key (which responds to normal queries). It is necessary for the reader to send an encrypted key to the back-end database in order to unlock a tag. As a result, a key is sent to the locked tag by the reader's device. The tag is now free to be used. Security has improved, but tracking remains an issue. A more complex scheme involves changing the hashed key in an unpredictable way [108].
3. **Lightweight cryptographic protocols:** In order to keep the cost of RFID tags low, a number of lightweight cryptographic protocols have been proposed. Peris et al. propose a lightweight and simple mutual authentication protocol for low-cost RFID tags. Their method, they claim, is secure enough for some applications even with the smallest RFID tags and requires fewer than 300 gates to implement. An easy-to-implement approach to tag-reader mutual authentication is put forth [109]. Their protocol uses a shared secret and a pseudorandom function to encrypt messages between the tag and the reader. A challenge-response authentication protocol, such as the [110] example, is another. In order to authenticate tags, these protocols can be used, but they can also be broken.

B. Solutions for Security Issues in Communication

Reliable Routing: Security protocols for IoT networks are hampered by the need to access message content before forwarding it. There have been a slew of plausible routing attacks put forth in the literature. The first detailed security analyses of major routing protocols, attacks, and countermeasures [111]. Routing security and privacy have been addressed in other studies [112-114].

IDS: At the communication level, an IDS is required to monitor network operations and communication links and to raise an alert in the event that any anomaly is observed, such as when a pre-defined policy is violated [115, 116]. Customized IDS for WSNs or the Internet as a whole. IoT security and privacy concerns have not been addressed directly in recent IDS proposals. In the world of IoT(IoT) devices, SVELTE [117] is an IDS designed specifically for IPv6. Spoofed or altered data as well as Black Hole attacks can be detected by this routing system. Another IoT intrusion detection method is presented in [118].

Cryptographic Schemes

One of the most effective defences against eavesdropping and simple routing attacks is the use of cryptographic schemes to secure communication protocols. Communications have been protected by various encryption methods [119, 120]. There are a lot of IoT components that don't fit into the traditional wired network encryption-decryption framework. An edge node is a small sensor with limited processing power, battery life, and memory. Encryption increases the risk of data and packet loss [121]. In the Internet of Things, the use of AES-based secure communication has shown some potential. CLEFIA [122] and PRESENT [123] are two other quick and easy encryption options. Public key encryption methods that provide adequate security without sacrificing portability are as yet unpromising.

Role-Based Authorization

A role-based authorization system checks to see if a component, such as an edge node, service provider, or router, can access, share, or modify information. Both parties must be authenticated and authorised by the authorization system [124].

C. Solutions for Security Issues at the Edge Computing Level

Pre-testing: Testing is required before any changes or design implementations are put into use on mission-critical systems. It is important to test the system's behaviour by feeding it various inputs and monitoring its outputs [125]. During pre-testing, potential attack scenarios are identified and simulated to see how the system responds [126]. Information about what data should be logged and what data should not is included in this section as well. Malicious code should be checked for in the input files as well. The attacker should not be able to execute commands that have been injected into the input files.

IDS: A malicious node can be detected by an IDS if it tries to insert invalid data or violate policies. A number of recent studies have suggested IDS as a possible solution to the injection problem, DIGLOSSIA [127], a new server code injection detection tool, was designed and implemented in [128].

VII. CONCLUSION

Many potential threats and attacks on the security and privacy of things and people have arisen in the last decade due to the growth of the IoT paradigm. This is only going to get worse in the future. The threat of IoT security are largely unknown. According to this survey, a number of IoT security threats and methods for preventing them were compiled. Finally, we wanted to show the reader which threats have been made, how they have been dealt with, and which threats still exist. It is imperative that these threats are dealt with quickly and proactively by both industrial and academic research communities, as well as by manufacturers, due to the wide range of IoT applications.

REFERENCES

1. D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *2014 IEEE world forum on Internet of Things (WF-IoT)*, 2014, pp. 287–292.
2. T. A. S. Srinivas and S. S. M. Manivannan, "Preventing collaborative black hole attack in IoT construction using a CBHA--AODV routing protocol," *Int. J. Grid High Perform. Comput.*, vol. 12, no. 2, pp. 25–46, 2020.
3. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
4. R. Somula, S. Nalluri, M. K. NallaKaruppan, S. Ashok, and G. Kannayaram, "Analysis of CPU scheduling algorithms for cloud computing," in *Smart Intelligent Computing and Applications*, Springer, 2019, pp. 375–382.
5. R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, 2012, pp. 257–260.
6. T. A. S. Srinivas, S. Ramasubbareddy, G. Kannayaram, and C. S. P. Kumar, "Storage Optimization Using File Compression Techniques for Big Data.," in *FICTA (2)*, 2020, pp. 409–416.
7. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.
8. J. Green, "The internet of things reference model," in *Internet of Things World Forum*, 2014, pp. 1–12.
9. T. Srinivas and S. S. Manivannan, "Black Hole and Selective Forwarding Attack Detection and Prevention in IoT in Health Care Sector: Hybrid meta-heuristic-based shortest path routing," *J. Ambient Intell. Smart Environ.*, no. Preprint, pp. 1–24, 2021.
10. T. A. S. Srinivas, S. Ramasubbareddy, A. Sharma, and G. Kannayaram, "Optimal Energy Distribution in Smart Grid.," in *FICTA (2)*, 2020, pp. 383–391.
11. S. Sankar, R. Somula, B. Parvathala, S. Kolli, S. Pulipati, and others, "SOA-EACR: Seagull optimization algorithm based energy aware cluster routing protocol for wireless sensor networks in the livestock industry," *Sustain. Comput. Informatics Syst.*, vol. 33, p. 100645, 2022.
12. M. M. Kermani, M. Zhang, A. Raghunathan, and N. K. Jha, "Emerging frontiers in embedded security," in *2013 26th international conference on VLSI design and 2013 12th international conference on embedded systems*, 2013, pp. 203–208.
13. M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE J. Emerg. Sel. Top. circuits Syst.*, vol. 3, no. 1, pp. 45–54, 2013.
14. E. Fleisch and others, "What is the internet of things? An economic perspective," *Econ. Manag. Financ. Mark.*, vol. 5, no. 2, pp. 125–157, 2010.
15. M. Tajima, "Strategic value of RFID in supply chain management," *J. Purch. supply Manag.*, vol. 13, no. 4, pp. 261–273, 2007.
16. M. Zhang, A. Raghunathan, and N. K. Jha, "Trustworthiness of medical devices and body area networks," *Proc. IEEE*, vol. 102, no. 8, pp. 1174–1188, 2014.
17. C. Li, A. Raghunathan, and N. K. Jha, "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system," *2011 IEEE 13th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2011*, pp. 150–156, 2011.
18. D. Halperin *et al.*, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008, pp. 129–142.
19. Y. Cherdantseva and J. Hilton, "A reference model of information assurance \& security," in *2013 International Conference on Availability, Reliability and Security*, 2013, pp. 546–555.
20. D. B. Parker, *Fighting computer crime: A new framework for protecting information*. John Wiley \& Sons, Inc., 1998.
21. M. E. Whitman, H. J. Mattord, and A. Green, *Hands-on information security lab manual*. Cengage Learning, 2014.
22. P. Bhuvaneshwari, A. Nagaraja Rao, T. Aditya Sai Srinivas, D. Jayalakshmi, R. Somula, and K. Govinda, "Evaluating the performance of sql* plus with hive for business," in *Advances in Big Data and Cloud Computing*, Springer, 2019, pp. 469–476.

23. S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan attacks: Threat analysis and countermeasures," *Proc. IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014.
24. H. Salmani and M. M. Tehranipoor, "Vulnerability analysis of a circuit layout to hardware Trojan insertion," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1214–1225, 2016.
25. T. Wehbe, V. J. Mooney, D. C. Keezer, and N. B. Parham, "A novel approach to detect hardware Trojan attacks on primary data inputs," in *Proceedings of the WESS'15: Workshop on Embedded Systems Security*, 2015, pp. 1–10.
26. S. Bhasin and F. Regazzoni, "A survey on hardware trojan detection techniques," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 2021–2024.
27. D. M. Shila and V. Venugopal, "Design, implementation and security analysis of hardware Trojan threats in FPGA," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 719–724.
28. H. Tanaka, "Information leakage via electromagnetic emanations and evaluation of tempest countermeasures," in *International Conference on Information Systems Security*, 2007, pp. 167–179.
29. M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards.," in *USENIX security symposium*, 2009, vol. 8, pp. 1–16.
30. A. M. Nia, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Physiological information leakage: A new frontier in health information security," *IEEE Trans. Emerg. Top. Comput.*, vol. 4, no. 3, pp. 321–334, 2015.
31. T. A. S. Srinivas, R. Somula, K. Aravind, and S. S. Manivannan, "Pattern Prediction Using Binary Trees," *Innov. Comput. Sci. Eng. Proc. 8th ICICSE*, vol. 171, p. 43, 2021.
32. A. Brandt, J. Buron, and G. Porcu, "Home automation routing requirements in low-power and lossy networks," 2010.
33. S. Seys and B. Preneel, "Authenticated and efficient key management for wireless ad hoc networks," in *Proceedings of the 24th Symposium on Information Theory in the Benelux*, 2003, pp. 195–202.
34. T. Martin, M. Hsiao, D. Ha, and J. Krishnaswami, "Denial-of-service attacks on battery-powered mobile computers," in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*, 2004, pp. 309–318.
35. M. H. R. Khouzani and S. Sarkar, "Maximum damage battery depletion attack in mobile sensor networks," *IEEE Trans. Automat. Contr.*, vol. 56, no. 10, pp. 2358–2368, 2011.
36. A. Agah and S. K. Das, "Preventing DoS attacks in wireless sensor networks: A repeated game theory approach.," *Int. J. Netw. Secur.*, vol. 5, no. 2, pp. 145–153, 2007.
37. F. Stajano, "The resurrecting duckling—what next?," in *International Workshop on Security Protocols*, 2000, pp. 204–214.
38. A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet under the microscope," *ESET LLC (September 2010)*, 2010.
39. A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*, 2011, pp. 355–366.
40. A. Becher, Z. Benenson, and M. Dornseif, "Tampering with motes: Real-world physical attacks on wireless sensor networks," in *International Conference on Security in Pervasive Computing*, 2006, pp. 104–118.
41. M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *IEEE Wirel. Commun.*, vol. 17, no. 6, pp. 44–51, 2010.
42. G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart nest thermostat: A smart spy in your home," *Black Hat USA*, no. 2015, 2014.
43. B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *2005 IEEE symposium on security and privacy (S\&P'05)*, 2005, pp. 49–63.
44. J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," in *Security in distributed, grid, mobile, and pervasive computing*, Auerbach Publications, 2007, pp. 367–409.
45. H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *2003 Symposium on Security and Privacy, 2003.*, 2003, pp. 197–213.
46. T. Aditya Sai Srinivas, S. Ramasubbareddy, and K. Govinda, "Loan Default Prediction Using Machine Learning Techniques," in *Innovations in Computer Science and Engineering*, 2022, pp. 529–535.
47. D. G. Padmavathi, M. Shanmugapriya, and others, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *arXiv Prepr. arXiv0909.0576*, 2009.
48. A. Juels, R. L. Rivest, and M. Szydlo, "The blocker tag: Selective blocking of RFID tags for consumer privacy," in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 103–111.
49. S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in pervasive computing*, Springer, 2004, pp. 201–212.
50. A. Juels, "RFID security and privacy: A research survey," *IEEE J. Sel. areas Commun.*, vol. 24, no. 2, pp. 381–394, 2006.
51. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "RFID systems: A survey on security threats and proposed solutions," in *IFIP international conference on personal wireless communications*,

2006, pp. 159–170.
52. S. H. Weingart, "Physical security devices for computer subsystems: A survey of attacks and defenses," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2000, pp. 302–317.
53. M. Lehtonen, D. Ostojic, A. Ilic, and F. Michahelles, "Securing RFID systems by detecting tag cloning," in *International Conference on Pervasive Computing*, 2009, pp. 291–308.
54. J. Westhues, S. Garfinkel, and B. Rosenberg, "Hacking the prox card," *RFID Appl. Secur. Priv.*, pp. 291–300, 2005.
55. D. N. Duc and K. Kim, "Defending RFID authentication protocols against DoS attacks," *Comput. Commun.*, vol. 34, no. 3, pp. 384–390, 2011.
56. T. Karygiannis, B. Eydt, G. Barber, L. Bunn, T. Phillips, and others, "Guidelines for securing radio frequency identification (RFID) systems," *NIST Spec. Publ.*, vol. 80, pp. 1–154, 2007.
57. I. Syamsuddin, T. Dillon, E. Chang, and S. Han, "A survey of RFID authentication protocols based on hash-chain method," in *2008 Third International Conference on Convergence and Hybrid Information Technology*, 2008, vol. 2, pp. 559–564.
58. G. Hancke and others, "Eavesdropping attacks on high-frequency RFID tokens," in *4th Workshop on RFID Security (RFIDSec)*, 2008, vol. 9440, pp. 259–288.
59. D. Carluccio, K. Lemke, and C. Paar, "Electromagnetic side channel analysis of a contactless smart card: first results," in *ECrypt Workshop on RFID and Lightweight Crypto*, 2005.
60. A. Mukherjee, "Physical-layer security in the Internet of Things: Sensing and communication confidentiality under resource constraints," *Proc. IEEE*, vol. 103, no. 10, pp. 1747–1761, 2015.
61. T. A. S. Srinivas, G. Mahalaxmi, R. Varaprasad, and D. Raziya, "A Comprehensive Survey of Techniques, Applications, and Challenges in Deep Learning: A Revolution in Machine Learning."
62. A. D. Wood, J. A. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks," in *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2007, pp. 60–69.
63. M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: Reactive jamming in wireless networks: How realistic is the threat?," in *Proceedings of the fourth ACM conference on Wireless network security*, 2011, pp. 47–52.
64. A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Commun. Surv. \& Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.
65. S. Chaturvedi, R. Karthikeyan, M. Vijayaraj, N. Kumar, M. Sangeetha, and others, "Medical Image Denoising and Classification Based on Machine Learning: A Review," *ECS Trans.*, vol. 107, no. 1, p. 6111, 2022.
66. T. Aditya Sai Srinivas, S. Ramasubbareddy, and K. Govinda, "Discovery of Web Services Using Mobile Agents in Cloud Environment," in *Innovations in Computer Science and Engineering*, Springer, 2019, pp. 465–471.
67. A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Trans. Emerg. Top. Comput.*, vol. 5, no. 4, pp. 586–602, 2016.
68. A. S. S. Thuluva, M. S. Somanathan, R. Somula, S. Sennan, and D. Burgos, "Secure and efficient transmission of data based on Caesar Cipher Algorithm for Sybil attack in IoT," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, pp. 1–23, 2021.
69. T. A. S. Srinivas and S. S. Manivannan, "Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm," *Comput. Commun.*, vol. 163, pp. 162–175, 2020.
70. T. A. S. Srinivas, S. Ramasubbareddy, and K. Govinda, "Loan Default Prediction Using Machine Learning Techniques."
71. I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *2014 federated conference on computer science and information systems*, 2014, pp. 1–8.
72. I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurr. Comput. Pract. Exp.*, vol. 28, no. 10, pp. 2991–3005, 2016.
73. S. W. Boyd and A. D. Keromytis, "SQLrand: Preventing SQL injection attacks," in *International conference on applied cryptography and network security*, 2004, pp. 292–302.
74. B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv Prepr. arXiv1206.6389*, 2012.
75. X. Lin and P. P. K. Chan, "Causative attack to incremental support vector machine," in *2014 International Conference on Machine Learning and Cybernetics*, 2014, vol. 1, pp. 137–142.
76. B. I. P. Rubinstein *et al.*, "Stealthy poisoning attacks on PCA-based anomaly detectors," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, pp. 73–74, 2009.
77. K. T.Aditya Sai Srinivas Govinda, "Warehouse stock prediction using krill herd algorithm," 2020.
78. K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, 2013, pp. 15–20.
79. B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Secur. \& Priv.*, vol. 9, no. 2, pp. 50–57, 2010.

80. A. Nejat, S. M. H. Shekarian, and M. S. Zamani, "A study on the efficiency of hardware Trojan detection based on path-delay fingerprinting," *Microprocess. Microsyst.*, vol. 38, no. 3, pp. 246–252, 2014.
81. N. Yoshimizu, "Hardware Trojan detection by symmetry breaking in path delays," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2014, pp. 107–111.
82. A. Srinivas, S. Ramasubbareddy, S. S. Manivannan, and K. Govinda, "Predicting User Behaviour on E-Commerce Site Using Ann," *Int. J. Eng. Technol.*, vol. 7, pp. 161–164, 2018.
83. K. Hu, A. N. Nowroz, S. Reda, and F. Koushanfar, "High-sensitivity hardware trojan detection using multimodal characterization," in *2013 Design, Automation \& Test in Europe Conference \& Exhibition (DATE)*, 2013, pp. 1271–1276.
84. S. S. Clark *et al.*, "Wattsupdoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices," in *2013 $\{$USENIX$\}$ Workshop on Health Information Technologies (HealthTech 13)*, 2013.
85. M. Msgna, K. Markantonakis, D. Naccache, and K. Mayes, "Verifying software integrity in embedded systems: A side channel approach," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2014, pp. 261–280.
86. N. Lesperance, S. Kulkarni, and K.-T. Cheng, "Hardware Trojan detection using exhaustive testing of k-bit subspaces," in *The 20th Asia and South Pacific Design Automation Conference*, 2015, pp. 755–760.
87. X. Ye, J. Feng, H. Gong, C. He, and W. Feng, "An anti-Trojans design approach based on activation probability analysis," in *2015 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, 2015, pp. 443–446.
88. R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: A statistical approach for hardware Trojan detection," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2009, pp. 396–410.
89. S. S. Doumit and D. P. Agrawal, "Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks," in *IEEE Military Communications Conference, 2003. MILCOM 2003.*, 2003, vol. 1, pp. 609–614.
90. C.-C. Su, K.-M. Chang, Y.-H. Kuo, and M.-F. Horng, "The new intrusion prevention and detection approaches for clustering-based sensor networks [wireless sensor networks]," in *IEEE Wireless Communications and Networking Conference, 2005*, 2005, vol. 4, pp. 1927–1932.
91. A. Agah, S. K. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: A non-cooperative game approach," in *Third IEEE International Symposium on Network Computing and Applications, 2004.(NCA 2004). Proceedings.*, 2004, pp. 343–346.
92. A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Quality of service \& security in wireless and mobile networks*, 2005, pp. 16–23.
93. M. S. I. Mamun, A. F. Kabir, M. Hossen, M. Khan, R. Hayat, and others, "Policy based intrusion detection and response system in hierarchical WSN architecture," *arXiv Prepr. arXiv1209.1678*, 2012.
94. A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer (Long. Beach. Calif).*, vol. 35, no. 10, pp. 54–62, 2002.
95. M. Zhang and N. K. Jha, "FinFET-based power management for improved DPA resistance with low overhead," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 3, pp. 1–16, 2011.
96. V. Sundaresan, S. Rammohan, and R. Vemuri, "Defense against side-channel power analysis attacks on microelectronic systems," in *2008 IEEE National Aerospace and Electronics Conference*, 2008, pp. 144–150.
97. D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of AES," in *Cryptographers' track at the RSA conference*, 2006, pp. 1–20.
98. [98] C. Wachsmann and A.-R. Sadeghi, "Physically unclonable functions (PUFs): Applications, models, and future directions," *Synth. Lect. Inf. Secur. Privacy, \& Trust*, vol. 5, no. 3, pp. 1–91, 2014.
99. K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *2010 IEEE international symposium on hardware-oriented security and trust (HOST)*, 2010, pp. 112–117.
100. A. Kanuparthi, R. Karri, and S. Addepalli, "Hardware and embedded security in the context of internet of things," in *Proceedings of the 2013 ACM workshop on Security, privacy \& dependability for cyber vehicles*, 2013, pp. 61–64.
101. T. Aditya Sai Srinivas, R. Somula, K. Aravind, and S. S. Manivannan, "Pattern Prediction Using Binary Trees," in *Innovations in Computer Science and Engineering*, 2021, pp. 43–52.
102. Y. W. Law, Y. Zhang, J. Jin, M. Palaniswami, and P. Havinga, "Secure rateless deluge: Pollution-resistant reprogramming and data dissemination for wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, pp. 1–22, 2011.
103. J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," in *International Conference on Research in Smart Cards*, 2001, pp. 200–210.
104. E. Y. Choi, S. M. Lee, and D. H. Lee, "Efficient RFID authentication protocol for ubiquitous computing environment," in *International Conference on Embedded and Ubiquitous Computing*, 2005, pp. 945–954.
105. T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *First*

International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), 2005, pp. 59–66.
106. S. M. Lee, Y. J. Hwang, D. H. Lee, and J. I. Lim, "Efficient authentication for low-cost RFID systems," in *International Conference on Computational Science and Its Applications*, 2005, pp. 619–627.
107. G. Avoine and P. Oechslin, "A scalable and provably secure hash-based RFID protocol," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, 2005, pp. 110–114.
108. A. S. T Srinivas, S. Ramasubbareddy, and K. Govinda, "Estimation of Web Vulnerabilities Based on Attack Tree and Threat Model Analysis," *J. Comput. Theor. Nanosci.*, vol. 16, no. 5–6, pp. 1993–2000, 2019.
109. D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 210–219.
110. I. Vajda, L. Buttyán, and others, "Lightweight authentication protocols for low-cost RFID tags," in *Second Workshop on Security in Ubiquitous Computing--Ubicomp*, 2003, vol. 2003.
111. C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2–3, pp. 293–315, 2003.
112. Y. W. Law and P. J. M. Havinga, "How to secure a wireless sensor network," in *2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2005, pp. 89–95.
113. P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002, no. CONF.
114. R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples," in *2012 IEEE international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*, 2012, pp. 1–7.
115. A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, and W.-C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surv. \& Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.
116. Y. Wang, H. Yang, X. Wang, and R. Zhang, "Distributed intrusion detection system based on data fusion method," in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, 2004, vol. 5, pp. 4331–4334.
117. S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.
118. C. Liu, J. Yang, R. Chen, Y. Zhang, and J. Zeng, "Research on immunity-based intrusion detection technology for the Internet of Things," in *2011 Seventh International Conference on Natural Computation*, 2011, vol. 1, pp. 212–216.
119. J. Daemen and V. Rijmen, "The design of Rijndael: AES-the advanced encryption standard Springer Science \& Business Media," 2013.
120. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*, 1997, pp. 394–403.
121. M. Katagi, S. Moriai, and others, "Lightweight cryptography for the internet of things," *Sony Corp.*, vol. 2008, pp. 7–10, 2008.
122. T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in *International workshop on fast software encryption*, 2007, pp. 181–195.
123. A. Bogdanov *et al.*, "PRESENT: An ultra-lightweight block cipher," in *International workshop on cryptographic hardware and embedded systems*, 2007, pp. 450–466.
124. S. Misra and A. Vaish, "Reputation-based role assignment for role-based access control in wireless sensor networks," *Comput. Commun.*, vol. 34, no. 3, pp. 281–294, 2011.
125. A. S. T Srinivas, K. Govinda, S. Ramasubbareddy, and E. Swetha, "Sentimental Analysis of Demonetization Over Twitter Data Using Machine Learning," *J. Comput. Theor. Nanosci.*, vol. 16, no. 5–6, pp. 2055–2058, 2019.
126. H. Mouratidis and P. Giorgini, "Security Attack Testing (SAT)—testing the security of information systems at design time," *Inf. Syst.*, vol. 32, no. 8, pp. 1166–1183, 2007.
127. S. Son, K. S. McKinley, and V. Shmatikov, "Diglossia: detecting code injection attacks with precision and efficiency," in *Proceedings of the 2013 ACM SIGSAC conference on computer \& communications security*, 2013, pp. 1181–1192.
128. Luong, V. "Intrusion detection and prevention system: SQL-injection attacks," 2010.