IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 16, 2020, accepted November 2, 2020, date of publication November 10, 2020,
date of current version November 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037032

A Cyber-Physical System to Detect IoT Security Threats of a Smart Home Heterogeneous Wireless Sensor Node

AKM JAHANGIR ALAM MAJUMDER, (Member, IEEE), CHARLES B. VEILLEUX, AND JARED D. MILLER
Division of Mathematics and Computer Science, University of South Carolina Upstate, Spartanburg, SC 29303, USA

Corresponding author: AKM Jahangir Alam Majumder (majumder@mailbox.sc.edu)

This work was supported by the Office of the Division of Mathematics and Computer Science at USC Upstate under Grant UP000-981065-A0001-101.

ABSTRACT The advent of the Internet of Things (IoT) allows the Cyber-Physical System (CPS) components to communicate with other devices, and to interact with safety-critical systems, posing new research challenges in security, privacy, and reliability. Efficient power measurement in smart IoT devices has become one of the key research topics. In this paper, we design and develop a CPS to detect IoT security threats via behavioral power profiling of a heterogeneous wireless sensor device using a Raspberry Pi and a smartphone. Experimentation and verification have been conducted on a group of smart IoT devices with different test scenarios, including the device in an idle and active state with distributed denial-of-service (DDoS) and a man-in-the-middle (MitM) attack. We propose to use the device power consumption rate to predict and detect a security threat using statistical signal processing and multivariate regression model. The proposed system can detect a potential security threat with an average accuracy of 80% and a device high of 89%.

INDEX TERMS Cybersecurity, power, AI, IoCPT, CPS, multivariate regression model.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

The interconnecting cyber and physical worlds give rise to new risky security challenges. The smart Internet of Cyber-Physical Things (IoCPT) is easily one of the most versatile technologies in existence [1], [2]. IoT devices in smart homes have become increasingly vulnerable to numerous security and privacy threats [3]–[7]. Over the past few years, devices such as Smart Hub, Smart Camera have been used to launch different cyber-attacks wherein attackers exploit weakly configured IoT devices and inject malicious code after discovering their credentials [8]. Between 2019 and 2030 the number of IoT connected devices in the world will grow from 7.6 billion to 24.1 billion, with revenue more than tripling from $465 billion to over $1.5 trillion. Those are the headline figures of the IoT Total Addressable Market (TAM) forecasts published by Transforma Insights in May 2020 [9].

The largest applications are Security, Electricity Smart Meters, Payment Processing (i.e. card payment machines), and consumer electronics in the form of a Personal-Portable Electronics and AV Equipment. All of these have revenues of over USD50 billion in 2025 [9]. Also, the design of an IoT Early Warning System (EWS) to save lives through early detection as one of the main technology integrations to help prevent the spread of coronavirus (COVID-19) [10]. The Cybersecurity, Internet of Things (IoT) along with artificial intelligence (AI) will help predict future outbreak areas [11]–[13]. IoT, a secure network of interconnected systems and advances in digital certification, data analytics, AI, and ubiquitous connectivity, can help by providing an early warning system to curb the spread of infectious diseases.

The variety and range of functions of smart devices present countless ways of improving different industries and environments. While the ‘‘things’’ in the IoT benefit homes, factories, and cities, these devices can also introduce blind spots and security risks in the form of vulnerabilities [14]–[18]. Vulnerable smart devices open networks to attack and can weaken the overall security of the internet. For now, it is better to be cautious and understand that ‘‘smart’’ can also mean vulnerable to threats. IoT devices are vulnerable largely because these devices lack the necessary built-in security to

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

counter threats. Aside from the technical aspects, users also contribute to the devices' vulnerability to threats [19]–[21].

B. LIMITATIONS OF PREVIOUS WORK

In previous studies, the application of IoT in different areas and industries has been widely discussed and reviewed [22]–[24]. Additionally, challenges and opportunities concerning the deployment of one or more IoT technologies have received a high level of technical assessment, e.g., sensors [25] or 5G networks [26]. One area 5G technology has the potential to affect in a big way is the IoT and its security. Concerning the energy sector, most survey studies have focused on one specific subsector, e.g., buildings or the technical potential of certain IoT technologies in the energy sector. For example, Stojkoska *et al.* [27] review smart home applications of IoT and the prospect of integrating those applications into an IoT enabled environment. Khatua *et al.* [28] review the key challenges in the suitability of IoT data transfer and communication protocols for deployment in smart grids.

In [29], the authors present an example of the increasing demand for power consumption and efficiency-measuring platforms for different IoT devices. However, it is also one example of interference from nearby radio frequency (RF) equipment that leads to increased packet loss, which, in turn, leads to improper results in power consumption. Another difficulty the authors face is the lack of a standard in data created by power monitoring solutions. Brick [30] is one of the proposed uniform schemas defining a concrete ontology for IoT sensors, subsystems, and relationships among them, which would enable portable applications.

Researchers presented a low-cost solution for implementing power consumption and environmental monitoring using an open-source IoT infrastructure, the monitoring in educational buildings [31]–[34]. With the use of IoT devices, it provides insights derived from initial results which concerns a deployment inside a university building. In [35], a different approach to power savings is presented by using occupancy sensors in large commercial buildings to determine the occupancy patterns in certain areas and thus creating a more efficient schedule pattern that can reduce power consumption by up to 38% while maintaining thermal comfort.

C. PROPOSED APPROACH

The primary goal of this research is to present a smart cyber-physical system to detect IoT security threats as shown in Figure 1. The proposed system uses automated detection of anomalies to alert security admins about suspicious behaviors of IoT smart wireless sensor devices, by comparing device current and behavioral power consumption data. Behavioral power profiling was chosen because it is the only data that can universally be gathered without physical intrusion. In this research, we design and develop a CPS to assess the integrated wireless sensor-based security and privacy interventions using statistical signal processing and multivariate logistic regression models. The proposed system can also be integrated into Energy Management Systems (EMS) such as smart home EMS for security analytics. This solution may also be implemented at the core/data center level or as a service in the Cloud.

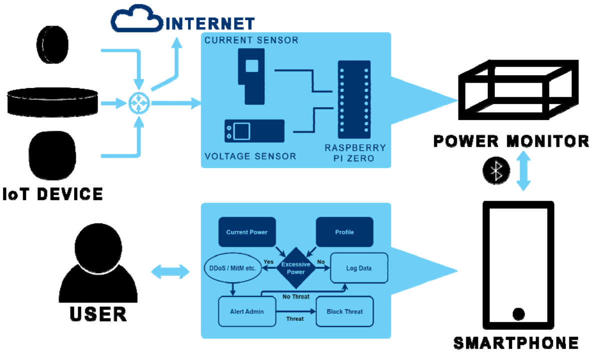


FIGURE 1. Overview of our proposed system.

D. MAJOR CONTRIBUTIONS

In this paper, we propose to use a smart IoT system as the platform for developing a CPS for detecting an IoT security threat by behavioral power monitoring of the wireless sensor devices. This work is a continuation of previous research published in [36]. Our major contributions of this paper are as follows:

- Developed a smart Cyber-Physical System (CPS) to detect cybersecurity threats through behavioral power profiling.
- Proposed proactive and provable monitoring of power uses in smart IoT devices using a smartphone.
- Used a general multivariate regression model and statistical signal classification techniques to detect abnormal power behavior and notify the admin.

The rest of the paper is organized as follows: in Section II, we describe the background and related work. In Section III, we discuss the solution process of designing our system. In Section IV, we discuss the methodology of our system, including the data collection process and follow with sections V and VI, which are the results and evaluation of our smartphone-based prototype system. In section VII, we discuss the raspberry pi implementation. Finally, in Section VIII, we conclude the paper with some future research directions.

II. RELATED WORK

Recently, the need for IoT device security has been profoundly researched by many. Most research has centered around detecting abnormal activity from the device or on the network.

Botnets and malware change at a rapid rate and as such the means of detecting these compromised IoT devices vary but many have not shown promising results.

A. NON-POWER BASED P2P BOTNET DETECTION

In a study of peer-to-peer (P2P) botnets, Alauthaman *et al.* [37] purpose a method of detection in which TCP control packets are scanned within a network. As P2P based bots typically communicate via TCP control packets, it was theorized that by filtering network traffic and only monitoring TCP control packets that the reliability and speed of network-based bot detection would be improved. After the extraction of features from the packet header and reduction of features to reduce classification errors they used a neural network to build a detection scheme. With their detection method, they were able to achieve around a 99% accuracy in the detection of P2P based botnets using TCP control packets. Along with this [38] purposed a method of detection aimed at detecting P2P botnets based on behavior analysis. They aimed to detect the command channel communication by detecting abnormal behavior of a device when connecting to an application or service. Since the behavior of a bot differs from a typical user in how it establishes a connection that behavior can be used to determine if a device may be compromised. A compromised device typically with dynamically search for other peers in an attempt to establish a connection to send and receive orders, the traffic generated in doing so is distinct and abnormal. With this method, they were able to achieve a high detection rate of compromised devices and could actuality distinguish between normal and abnormal activity.

B. DATA ENCRYPTION

Device security also extends to the transfer of data from IoT devices to outside sources. Through spoofing methods such as MitM, an attacker can intercept transmitted data. Most IoT devices use some form of encryption, such as DES, AES, or RSA to ensure that even if data is intercepted it is difficult to decrypt. The drawback to these encryption methods is that they require processing power to encrypt and decrypt. As many IoT devices do not draw much power and do not have much processing power to spare, encryption and decryption can be an expensive task for these devices. [39] proposed the use of quantum-safe algorithms, but stated that they are not a viable option due to the key size used in these methods being too great for IoT devices to handle. [40] propose an encryption method using both AES and RSA algorithms. First, the data is encrypted via AES then again with RSA before it is sent. They state that the use of RSA compensates for the shortcoming of AES. The combination of RSA and AES was found to be slower and resulted in the greater CPU utilization but was more secure than either algorithm on its own. The CPU utilization of this method was seen to even out when the number of users accessing the system increased.

Another method proposed by [41] is proxy re-encryption. They propose that a central proxy server handles the collection of all nodes (IoT devices). Each node creates a re-encryption key and sends it to the proxy server. This method allows the proxy server to handle encryption, reducing the computation power required for communication by each node. This method opens up the option to use resource-intensive encryption methods as the encryption is handled by a proxy. Using this method, communication time between devices was found to be drastically reduced when a greater number of nodes are connected.

A much more complex method is proposed by [42] in which they propose their JEDI system. JEDI is an E2EE protocol that allows for communication between decoupled devices. JEDI works via a publish and subscription method based on attributes of the data. Access to data is delegated in a hierarchy that is set with expiration dates. JEDI is based on WKD-IBE encryption, the attributes and time are used as the WKD-IBE vector. WKD-IBE allows for use of wild cards in its vector, as such, they can grant access to specific attributes by replacing wildcards of the attribute hierarchy with specific attributes. They show that with this method a many-to-many E2EE method is possible between IoT devices, but does still draw more power than many low power IoT devices can handle.

C. DEVICE PROFILING

Bridges *et al.* [43] have shown that profiling of device power is a viable approach. They found that by directly monitoring the power usage of a CPU that an accurate power profile could be constructed. Each profile was produced for a specific CPU, having its power usage recorded while in a neutral state and while compromised by a rootkit. They found that the rootkit compromised devices yielded a significant difference in CPU power usage compared to the uncompromised devices across every CPU. They also acknowledge a plateau in power once a device had achieved 100% CPU usage, which was accounted for in their detection of Malware. Along with this power plateau, they noticed periods of high-power usage, which they attributed to background OS processes. As the authors acknowledged, this random power variation is a major obstacle in the detection of malicious software on devices via power analysis. This noise in the power usage can easily obscure abnormal behavior that may occur, along with this any malware designed to use minimal power may not be detected at all [44].

D. MALWARE'S EFFECT ON POWER

Many prior studies have shown that malware produces changes in the power usage of a device. Clark *et al.* [45] showed that their power monitoring device WattsUpDoc was able to detect a compromised piece of medical equipment with an accuracy of 99.5%. Their device monitored system-wide power usage, unlike CPU specific monitoring of [46], [47]. The device required knowledge of the device's activity in both a normal and abnormal state before use. This means that the device must be trained per device before use. Similarly, [48]–[51] performed power analysis on smartphones, comparing power usage of both compromised and normal device states. While they were able to find if a device was compromised, they concluded a high degree of accuracy was required to do so. Along with this accuracy, the normal activity would have to be known to the device before use.

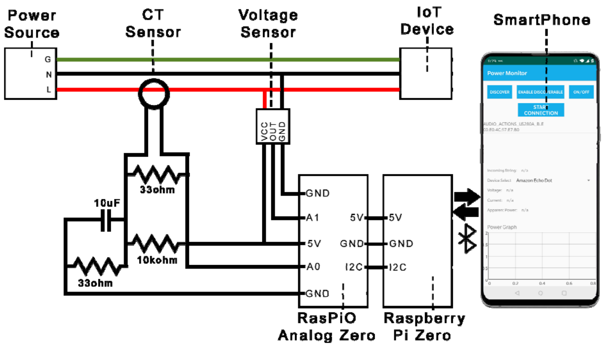


FIGURE 2. Architecture of our proposed system.

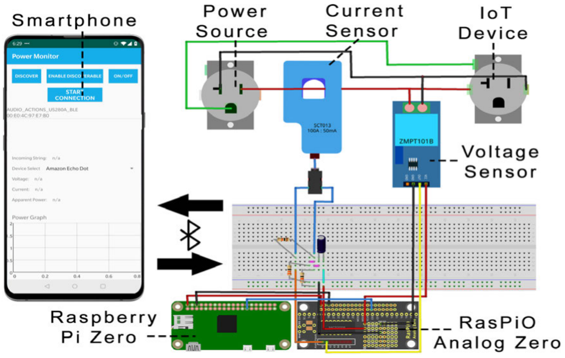


FIGURE 3. Proposed system with hardware components.

III. SYSTEM ARCHITECTURE

For versatility and ease of use, the system is independent of the monitored device and uses minimal power. The collected data is recorded and transmitted over Bluetooth to a smartphone, where the power, voltage, and current may then be monitored by a user. The architecture of our proposed system is shown in Figure 2.

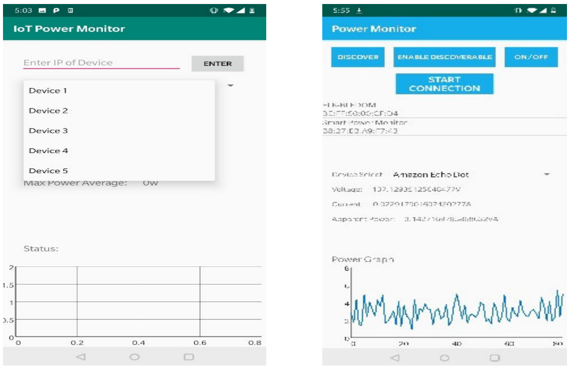
A. DESIGN OVERVIEW

The system is comprised of a Raspberry Pi Zero, RasPiO Analog Zero, YHDC SCT013 current sensor, ZMPT101B voltage sensor, and a smartphone for monitoring purposes. The RasPiO Analog Zero is a HAT (Hardware Attached on Top) that allows for the reading of analog signals. The RasPiO Analog Zero uses an MCP3008, a 10-bit analog to digital converter allowing it to output integer values ranging from 0-1023 over 8 analog channels (A0 to A7).

The YHDC SCT013 is a non-invasive alternating current (AC) sensor that is clipped around the N leg leading to the device. The developed prototype with hardware components is shown in Figure 3. As the current sensor outputs over a 3.5mm jack, a 3.5mm adapter is used to connect it to the Raspberry pi. The YHDC SCT013 returns an analog reading that is received by the RasPiO Analog Zero then sent to the Raspberry Pi Zero. The ZMPT101B voltage sensor returns an analog reading between 0 and 1024 which is received by the Raspberry Pi Zero via the RaspPiO Analog zero and converted into a voltage reading. The current and voltage readings are used to calculate the power drawn by the device. All readings are then sent via Bluetooth from the Raspberry Pi to the smartphone for monitoring by the user. The device is powered via the PWN IN port on the Raspberry Pi Zero.

B. SMARTPHONE INTERFACE

The system uses an Android application to monitor data, which can be seen in Figure 4. In the current implementation of the system, the data is sent over file transfer to the phone. The file consists of lines containing the power, voltage, and current recorded by the system. Each line of the file contains all three values found in one second, each value is separated by a delimiter. The string is then separated into the three separate readings for power, voltage, and current.



Max Power Average: 0W

Device Select: Amazon Echo Dot
Voltage: 137.12935125648477V
Current: 0.0229173016374 50277A
Apparent Power: 3.142 [illegible]VA

(a) IoT device data collector interface

(b) Application UI

FIGURE 4. Smartphone screenshots for proposed system sensor node data collection.

The apparent power is graphed overtime on the phone. If the system detects abnormal power a separate line is included notifying the application to alert the user. If the user determines that there was no threat and the notification was a false-positive, the notification is ignored and blocked. In the current system, the admin must still determine if the device is actually under attack and how to block the threat. In future implementations, we hope to display real time data to the application as opposed to processing sets of data after they are collected.

IV. DATA COLLECTION

For data collection, a standardized test was established that could be used for any IoT device tested. Each test lasted for a total duration of 6 hours. Each test was made up of 6 scenarios:

1) One hour in an idle state.
2) One hour in an active state.
3) One hour idle in an idle state while the IoT device is under a simulated DDoS attack.
4) One hour active in an active state while the IoT device is under a simulated DDoS attack.
5) One hour idle in an idle state while the IoT device is under a real MitM attack.
6) One hour active in an active state while the IoT device is under a real MitM attack.

amazon
works with Amazon Alexa/Google Home/IFTTT
WIFI control
One APP control your home
Control your devices anywhere
Timing home appliances
Customize schedules for daily uses
gosund

(a) Echo Dot 2nd Gen (b) SmartThing Hub (c) Smart Socket

FIGURE 5. **Tested IoT devices for validation of systems.**

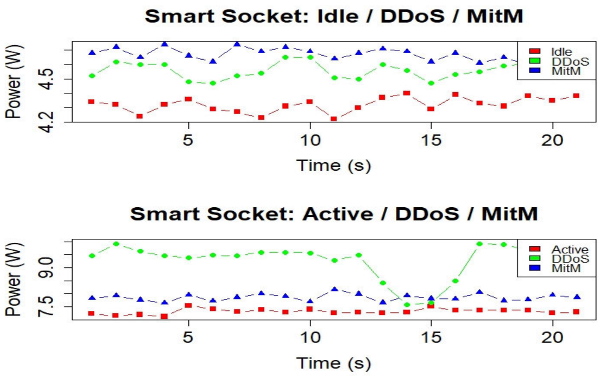


FIGURE 6. **A sample power variation on smart socket with and without threat.**

21600 data points were collected in a test, which consisted of six scenarios. Each scenario accounted for 3600 total data points or 1 data point per second. For data collection from our proposed system, a variety of IoT devices were used for testing. The tested devices are mainly comprised of wireless IoT components. The IoT Devices used were the Amazon Alexa Echo Dot 2nd Gen, 3rd Gen SmartThings Hub, and Avatar Mini Smart Socket as shown in Figure 5.

A sample real-time variation of power with/without attacks for the smart socket is shown in Figure 6. This can further be used to match events with spikes and sudden changes in power consumption, or even a behavior change.

We established a baseline period for each of the testing scenarios. This was achieved by manually finding the start (t_{start}) and end (t_{end}) events. We optimized the power parameters using the manually determined ground truth periods with and without attacks.

For $t_{0-Start} \leq t_{Start} \leq t_{0-End}$ and $t_{n-Start} \leq t_{End} \leq t_{n-End}$ as shown in Figure 7, it helps to reduce false positive, false negative, and total errors as follows,

$$\begin{aligned} \in_P &= \left(t_{Start} - t_{0-Start}\right) + \left(t_{n-End} - t_{End}\right) \\ \in_N &= \sum_{i=1}^{n} \left(t_{i:start} - t_{(i-1):End}\right) \\ \in_{total} &= \in_p + \in_N \end{aligned} \tag{1}$$

To minimize the error in the sample data, we eliminated 10 initial and end data samples in each test scenario to create more uniform matrices.

Currently, the collected data can be fully or partially processed on the smart IoT device. The data processing depends on the IoT application. The pseudocode of the IoT device power calculation is described in Algorithm 1 in more detail.

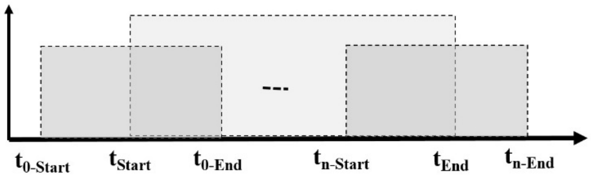


FIGURE 7. **Illustration of IoT device power intervals.**

Algorithm 1 IoT Device Power Calculation

```
Input: (Current Sensor Value, $i(t)$), and (Voltae Sensor
       Value, $v(t)$)
Output: apparentPower
Initialize apparentPower = 0, ADC = 1024,
offsetLevel = 512, currentValue = 0, voltageValue
= 0, offsetI = 320, offsetV = 320, currentFiltered =
0, voltageFiltered = 0, IRatio = 0.001, VRatio =
1.953125
loop
    numSamples ← 0
    voltageSum ← 0
    currentSum ← 0
    Open file for storage
    while (time < 1sec)
        numSamples+ = 1
        currentValue ← i (t) × (ADC − 1)
        voltageValue ← v (t) × (ADC − 1)
        currentFiltered =
        currentVal − offsetI − offsetLevel
        voltageFiltered =
        voltageVal − offsetV − offsetLevel
        currentSum+ = $currentFiltered^2$
        voltageSum+ = $voltageFiltered^2$
    end while
    Irms ← IRatio × $\sqrt{\frac{currentSum}{numSamples}}$
    Vrms ← VRatio × $\sqrt{\frac{voltageSum}{numSamples}}$
    apparentPower = Irms ∗ Vrms
    write apparentPower to file
    close file
end loop
```

V. OVERVIEW OF MULTIVARIATE LOGISTICS REGRESSION MODEL

We used the following approaches of multivariate logistic regression techniques to analyze the smart IoT devices state to identify security threats.

A. POWER SPECTRAL DENSITY

The Power Spectral Density (PSD) is the measure of a signal's power variation with frequency. The Periodogram of Power

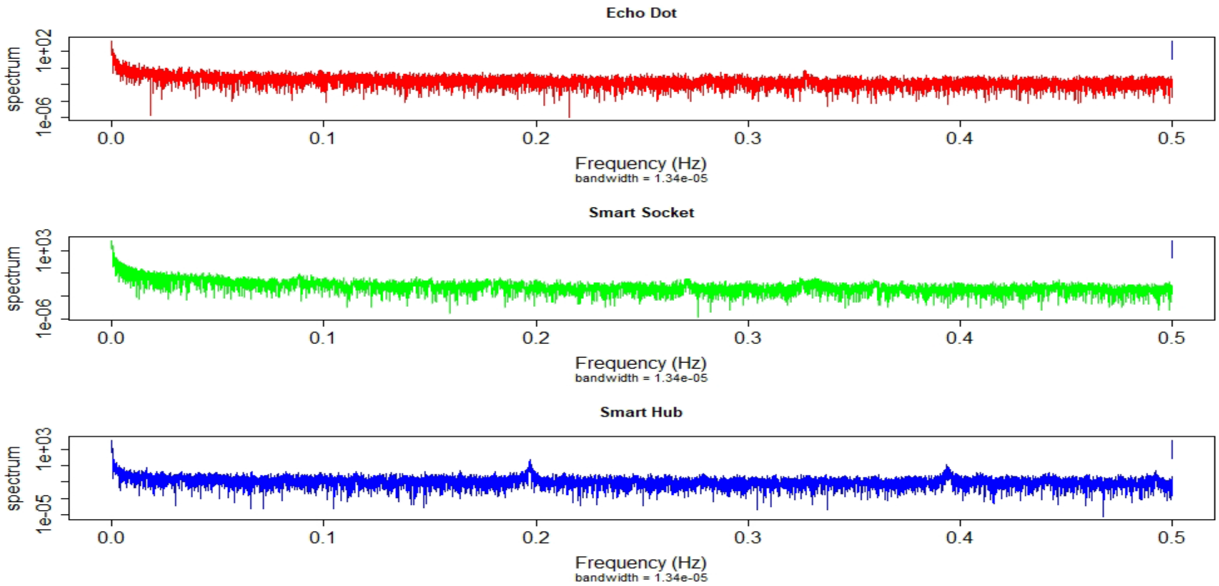


FIGURE 8. **Periodogram from PSD for Echo Dot, Smart Socket, and Smart Hub.**

Spectral Density (PSD) is an estimate via the periodogram method [52]. The periodograms for the Echo Dot, Smart Socket, and Smart Hub are shown in Figure 8.

Let, $x(t)$ is the voltage across or current through a unit resistor and $x^2(0t)$ is the instantaneous power of the signal, $x(t)$. The average power P of a signal $x(t)$ is therefore given by the following time average:

$$P = \lim_{n\to\infty} \frac{1}{T} \int_0^T |x(t)|^2 dt \tag{2}$$

and the expected instantaneous power is given by,

$$E\left[x^2(t)\right] = R_{xx}(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(j\omega)\, d\omega, \tag{3}$$

where, $S_{xx}(j\omega)$ is the Continuous-Time Fourier Transform (CTFT) of the autocorrelation function of $R_{xx}(\tau)$. The integral part in equation 3 suggests that we might be able to consider the instantaneous power in a frequency band of width $d\omega$ to be given by, $\frac{1}{2\pi} S_{xx}(j\omega)\, d\omega$. The instantaneous power in the output $y(t)$ can be expressed as the expected power that $x(t)$ has in the selected passband. Using the fact that,

$$S_{yy}(j\omega) = |H(j\omega)|^2 S_{xx}(j\omega) \tag{4}$$

where, $H(j\omega) = \frac{Y(j\omega)}{X(j\omega)}$ is the transfer function. We see that this expected power can be computed as,

$$\begin{aligned} E\left\{y^2(t)\right\} &= R_{yy}(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{yy}(j\omega)\, d\omega \\ &= \frac{1}{2\pi} \int S_{xx}(j\omega)\, d\omega \end{aligned} \tag{5}$$

Thus, $\frac{1}{2\pi} \int S_{xx}(j\omega)\, d\omega$ is the instantaneous power of $x(t)$ in the passband. Therefore $S_{xx}(j\omega)$ is the PSD of $x(t)$.

B. GENERALIZED LINEAR MODEL

Generalized linear models are a form of logistic regression. The general model follows the form of:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{6}$$

where:

β_0	is the odds of success when all other predictors are set to 0.
β_n	is the coefficient of the explanatory variable x_n.
n	is the number of explanatory variables.
p	is the odds of an event occurring.
$1-p$	is the odds of an event not occurring.

This can then be rearranged to:

$$p = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n}} \tag{7}$$

This results in an S-shaped curve. Values above a threshold are predicted one way, while values below the threshold are predicted to be the opposite. 0.5 is a standard threshold for binary outcomes.

For our models:

$$\begin{aligned} x_1 &= Power_{mean} \\ x_2 &= PSD_{mean} \\ \log\left(\frac{p}{1-p}\right) &= y = Threat\ orNo\ \ Threat \end{aligned}$$

Echo Dot Model:

$$\text{y} = 6.342 - 5.911x_1 - 27136x_2 \tag{8}$$

Smart Socket Model:

$$\text{y} = 3.081 - 0.9531x_1 - 166100x_2 \tag{9}$$

Smart Hub Model:

$$y = -0.4824 + 4.0263x_1 - 408.5575x_2 \tag{10}$$

New data is then fed into the model that corresponds with the device the data was collected from.

For our models:

$$x_1 = Power_{mean} \tag{11}$$

$$x_2 = PSD_{mean} \tag{12}$$

$$\log\left(\frac{p}{1-p}\right) = y = ThreatorNo\ Threat \tag{13}$$

Echo Dot Model:

$$y = 6.342 - 5.911x_1 - 27136\,x_2 \tag{14}$$

Smart Socket Model:

$$y = 3.081 - 0.9531x_1 - 166100\,x_2 \tag{15}$$

Smart Hub Model:

$$y = -0.4824 + 4.0263x_1 - 408.5575\,x_2 \tag{16}$$

New data is then fed into the model that corresponds with the device the data was collected from.

VI. EVALUATION OF THE PROPOSED SYSTEM

To evaluate our proposed system, we have developed a prototype application and investigated its performance. We evaluated the prototype with extensive experiments. In this section, we present how the data are analyzed and performance is measured.

A. INITIAL DATA ANALYSIS PROCESS

Before a model could be built, the IoT device raw power data needed to be:

1) cleaned
2) validated
3) organized

Cleaning and validation involved verifying that the collected data was accurate and within expected limits. Outliers were not removed as long as the data was within expected limits. Initial data collected for each scenario would often show power consumption to be over 1000 Watts when 15 Watts would have already been high. These data points were removed because they are believed to be the result of the device sensors still calibrating. Cleaning and validation were solely about ensuring the data collected was accurate. Fortunately, the only data that were removed were the initial data points.

The organization process was the process whereby the different sets of IoT power data were compiled into one of two different two-dimensional arrays:

1) The first array separated the different device states by column. This meant there were 6 columns, each consisting of 3600 data points. This was useful for summarizing and understanding the power consumption data of the different device states Idle, Active, Idle with DDoS, Active with DDoS, Idle with MitM, and Active with MitM.
2) The second array combined all the power consumption data into a single column. The second column of this array was comprised of the corresponding power spectral density (PSD) values, and the third column of this array contained known states (“Threat” / “No Threat”). The second array’s main purpose was for use in building a prediction model.

BUILDING A PREDICTION MODEL SOFTWARE

R and Microsoft Excel were used in the model building process. R has a strong foundation that supports statistical analysis. R community packages used were Spectrum, caret, and mclust. These three were integral for building our final model. R was the main software used for building the model, as well as most of our graphs. R allowed for quick reuse of code for analyzing data collected from different IoT devices.

There were instances where code reuse was not a major concern though. These were often one-off results, such as the cumulative confusion matrix. These were often accomplished much faster by just using Excel.

B. IOT DEVICE POWER PROFILING

To test the validity and long-term feasibility of our proposed system’s accuracy in power profiling, we tested different smart IoT devices and observed their power variations with and without cyber threats.

C. SUMMARY TABLE

While boxplots are wonderful for understanding and conveying information, getting an accurate value for the data shown is almost impossible. Summary tables alleviate this problem.

The summary tables show:

- Minimum and maximum values.
- 1st and 3rd quartile, which are the two sides of the box in the boxplot.
- The mean or average value.
- The median value, which is the bold line found in the boxplot.

Table 1 shows the summary tables for all device states on each of the three devices tested.

D. BOXPLOTS

An amazing resource for understanding the spread of data are boxplots. Two important features of a boxplot are the box and whiskers. The box accounts for 50% of all data collected. The whiskers on the left and right each account for 24.65%. Figure 9 shows the boxplot for the various states on all three devices tested. Looking at Figure 9, some device states show an apparent difference in power consumption, whereas other states hardly show any difference at all.

The box plots show the state-wise breakdown of power variations for IoT devices with and without attack. We observed that, although mean recognition accuracy was high, a few outliers exist.

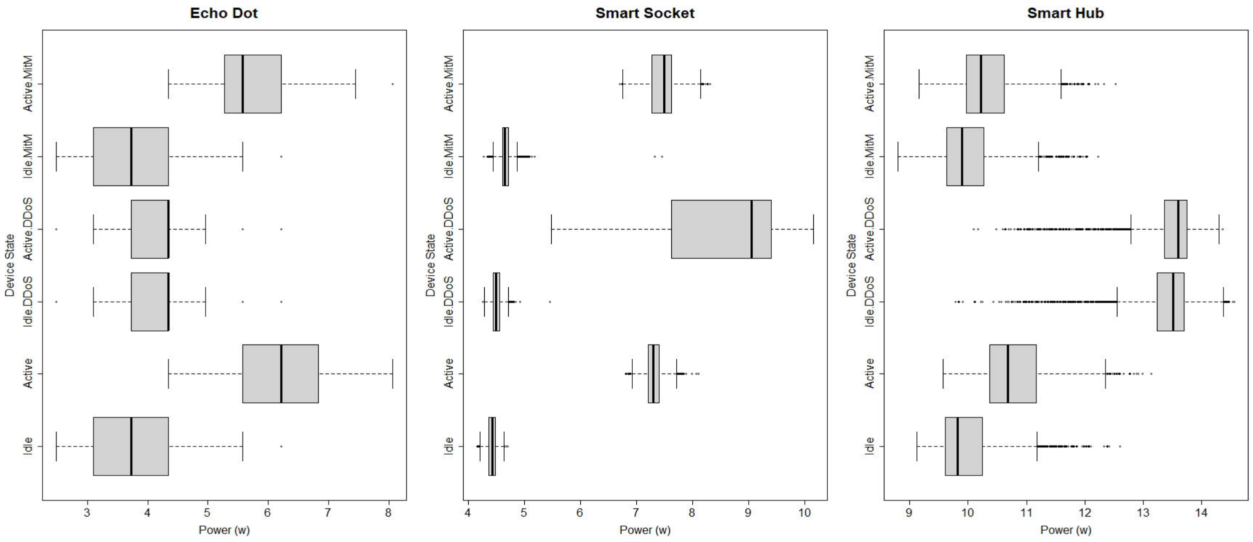


FIGURE 9. Box Plots for Power Profiling Accuracy for Security Threat Detection.

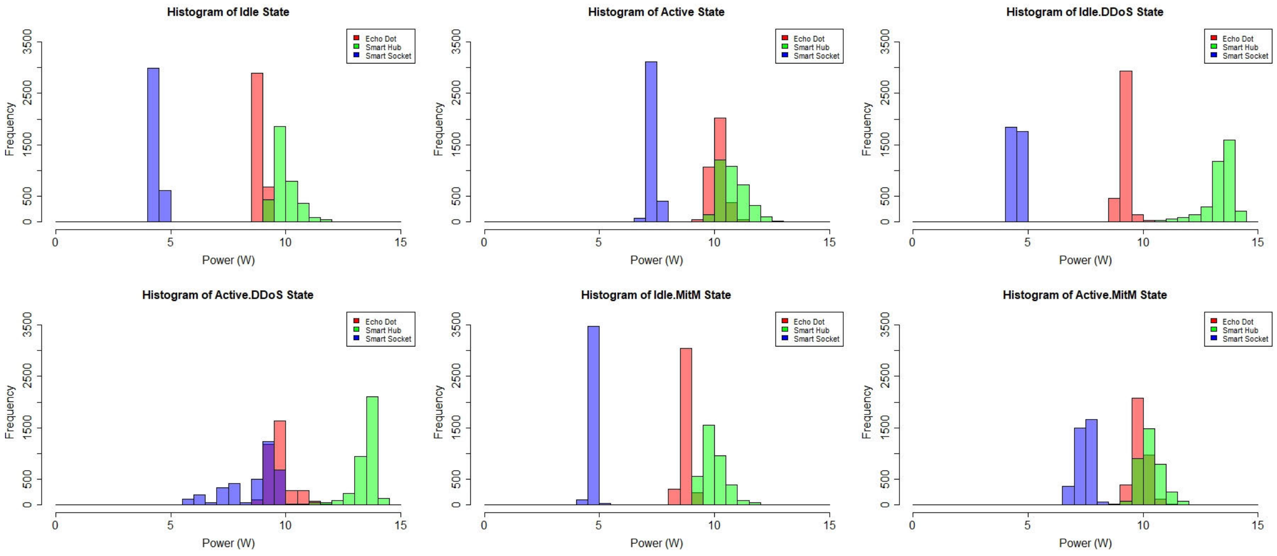


FIGURE 10. Performance of threat detection based on device power consumption grouped by idle and active state.

E. HISTOGRAM

Like boxplots, histograms show how the data is spread. However, histograms also highlight the frequency of values found. This is useful for understanding if there are only one or two values that comprise the bulk of data, or if the data is more uniformly spread. Figure 10 shows the histogram of the device state for all three devices. Figure 10 uses uniform limits for the axes. Uniform axes limits were chosen to easily compare the differences between different device states.

F. PREDICTIVE ANALYTICS FOR THREAT DETECTION

1) FLOW OF THE DIAGRAM

Figure 11 illustrates a simplified process from raw data to the finalized model. Starting with raw power, the 10 initial and end data points are removed. From there, the corresponding power spectral density is obtained.

The average value over 20 seconds is obtained. These are then merged into a single two-dimensional matrix. This step reduces the total number of data points from 21600 to just 1080. The values in this matrix are then normalized so that neither variable carries more value in building a model. A third column is appended to this matrix with the known state of the device.

At this point, the two-dimensional matrix has been finalized. This matrix is what is used to build a predictive model. Once the model is built, the results are then validated using k-fold validation.

2) ECHO DOT DATA

With a prediction accuracy of 89.1%, the best prediction results were obtained with the Echo Dot. As can be seen in Figure 13, the Echo Dot was the only IoT device tested

TABLE 1. Summary table of tested heterogeneous IoT devices

Device	State	Std. Dev.	1st Quart.	Median	Mean	3rd Quart.
Echo Dot	Idle	0.16	8.79	8.88	8.89	8.98
	Active	0.37	9.94	10.15	10.17	10.35
	Idle.DDoS	0.26	9.05	9.15	9.20	9.29
	Active.DDoS	0.53	9.41	9.62	9.73	9.88
	Idle.MitM	0.18	8.63	8.73	8.74	8.84
	Active.MitM	0.34	9.66	9.86	9.88	10.06
Smart Socket	Idle	0.08	4.37	4.43	4.43	4.48
	Active	0.16	7.21	7.30	7.32	7.41
	Idle.DDoS	0.09	4.45	4.50	4.51	4.56
	Active.DDoS	1.11	7.63	9.05	8.61	9.40
	Idle.MitM	0.12	4.61	4.66	4.67	4.72
	Active.MitM	0.28	7.28	7.50	7.45	7.63
Smart Hub	Idle	0.51	9.61	9.82	9.97	10.24
	Active	0.57	10.37	10.68	10.79	11.17
	Idle.DDoS	0.62	13.23	13.51	13.35	13.69
	Active.DDoS	0.53	13.37	13.59	13.46	13.75
	Idle.MitM	0.50	9.64	9.90	9.99	10.27
	Active.MitM	0.49	9.97	10.22	10.32	10.62

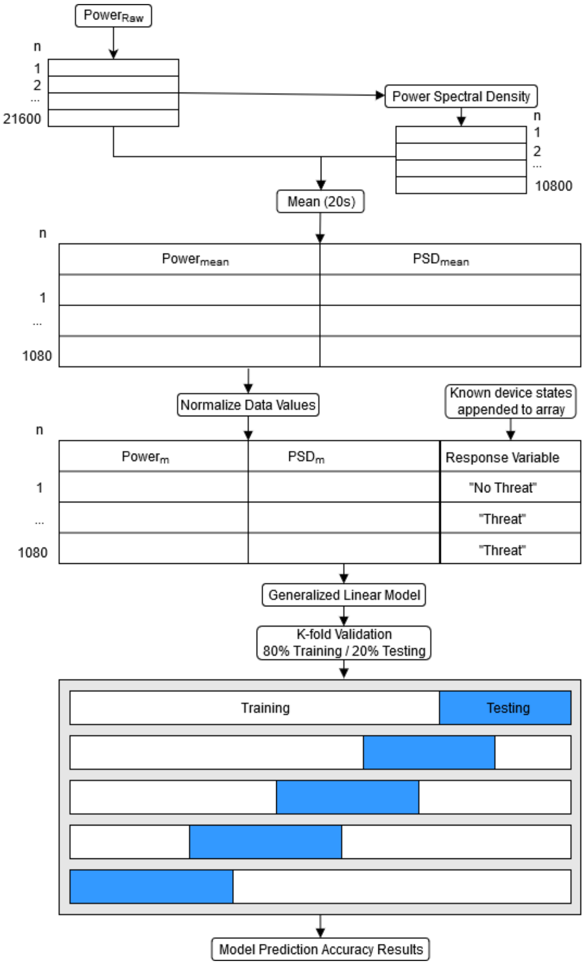


FIGURE 11. Prediction model building and analysis process.

where all threat specific prediction accuracies were above 80%. This is most likely due to the Echo Dot's heavy reliance on having a network connection to function, which is unlike either of the other two devices tested.

The Echo Dot functions by receiving user input through a variety of methods. This input is then relayed to an outside Lambda server where the bulk of the processing is done. This means any interruption in the connection to the network will become readily apparent as the Echo Dot is no longer able to function.

3) SMART SOCKET DATA

Coming in with an overall accuracy of 83.5%, the Smart Socket was our second-best device tested. The smart socket is a simple device that allows anyone to turn their socket in a smart socket. The functionality allows someone to control or check whether a device is on or off if they have an internet connection.

Unlike the Echo Dot, the smart socket does not disable the device connected to it if it loses connection to the internet. The smart socket operates on a "fail-open" system.

Figure 15 illustrates the results very well. The Left side (blue) is largely by itself, and similarly, the bottom (red) is by itself. However, there is a blue cluster nestled in with the red at the bottom.

4) SMART HUB DATA

The Samsung SmartThings Hub is an IoT device that allows different IoT devices to communicate and work together. The Smart Hub had the lowest prediction accuracy of the IoT devices tested. One thing to note is the similarity between idle/active and idle/active with MitM. We believe this similarity is a major reason our model has a low prediction accuracy.

The Smart Hub had the worst results with an overall prediction accuracy of 67.9%. Interestingly, even though it had the worst overall results, it had an unusually high threat specific prediction accuracy of 100% in detecting DDoS attacks. Empirical CDF for smart IoT devices is shown in Figure 14.

PREDICTION ACCURACY

Evaluating the quality and performance of a model was largely based on the model's ability to predict known device states. We define the prediction accuracy for each device as:

$$Accuracy_{fold} = \frac{\sum Correct\ Predictions}{\sum Predictions}$$

$$Accuracy_{device} = \frac{\sum Accuracy_{fold}}{k}$$

$$,\ where\ k\ is\ the\ number\ of\ folds$$

The final prediction results were obtained using 5-fold validation. 10-fold validation was also used. The results of 5 and 10-fold validation can be seen in Figure 12.

K-fold cross-validation is useful because it resamples the collected data to get more reliable accuracy. There is always a chance that the data split chosen could result in a high or low prediction accuracy. K-fold cross-validation lessens that chance.

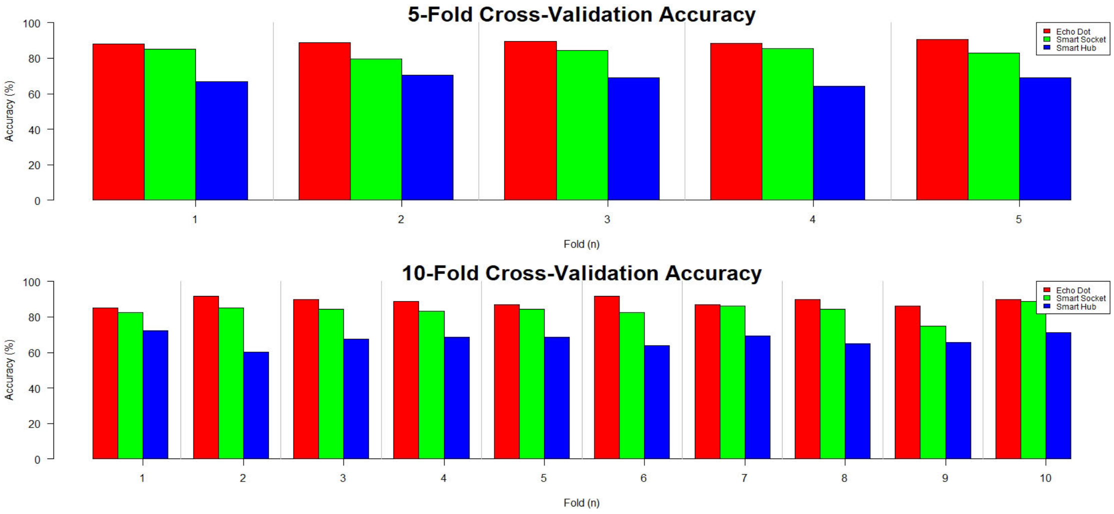


FIGURE 12. Threat detection accuracy for cross-fold analysis.

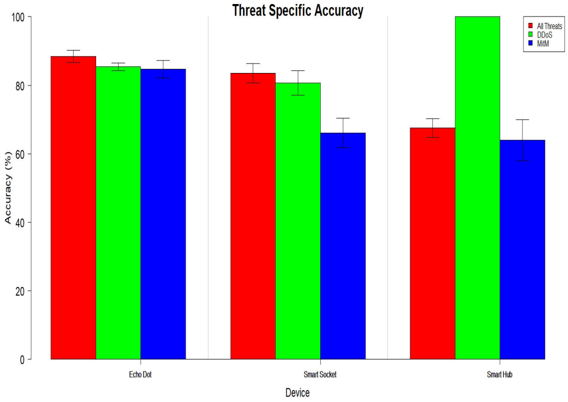


FIGURE 13. Threat specific device accuracy for threat detection.

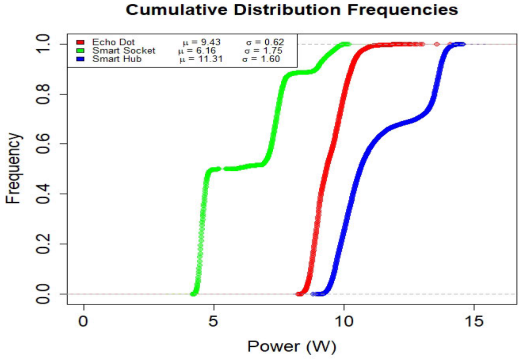


FIGURE 14. E-CDF of smart devices for accuracy analysis.

The device-specific accuracies were:
Echo Dot – 89.1%
Smart Socket – 83.5%
Smart Hub – 67.9%

The overall prediction was then determined by:

$$Accuracy_{Overall} = \frac{\sum Accuracy_{Device}}{Number\ of\ Devices}$$

The overall accuracy, shown in Figure 16, for all the devices was 80.17%.

It is important to note the variation in device-specific results. While some did very well on predicting the device state, others did not fare so well. These results demonstrate that predicting device state is possible, but doing so is device-dependent, as well as the attack type. The device-based cluster plot for threat detection accuracy is shown in Figure 15.

VII. RASPBERRY PI IMPLEMENTATIONS

A. RASPBERRY PI PYTHON IMPLEMENTATION

The system implements a Raspberry Pi Zero running Raspbian OS version 10. Due to this the device is programmed in python and uses the SciPy and NumPy libraries. The device collects about 40 points of data per second each point being between 0 and 1. These values are then converted into analog values based on the 10bit ADC of the MCP3008 chip. The square of these values is summed and then divide by the total number of samples collected. The square root of this dividend is multiplied by the respective ratio to find the Irms and Vrms respectively. With Irms and Vrms calculated the apparent power can be found by multiplying the values. The apparent power, amps, and volts are joined into one string and sent over Bluetooth to the smartphone using the PyBluez library. The shell of Thonny IDE on Raspberry Pi Zero is shown in Figure 17.

The apparent power values are then stored in an array of size 10 and another array of size 100. The first array is used to predict the state of the device, whereas the second array holds a record of the past 100 predicted states. To predict if the device is under attack a model was built for each device with three beta values each. The average power and PSD are found from the array of the last 10 seconds.

The probability is then calculated from the beta values, average power, and average PSD. The probability value is

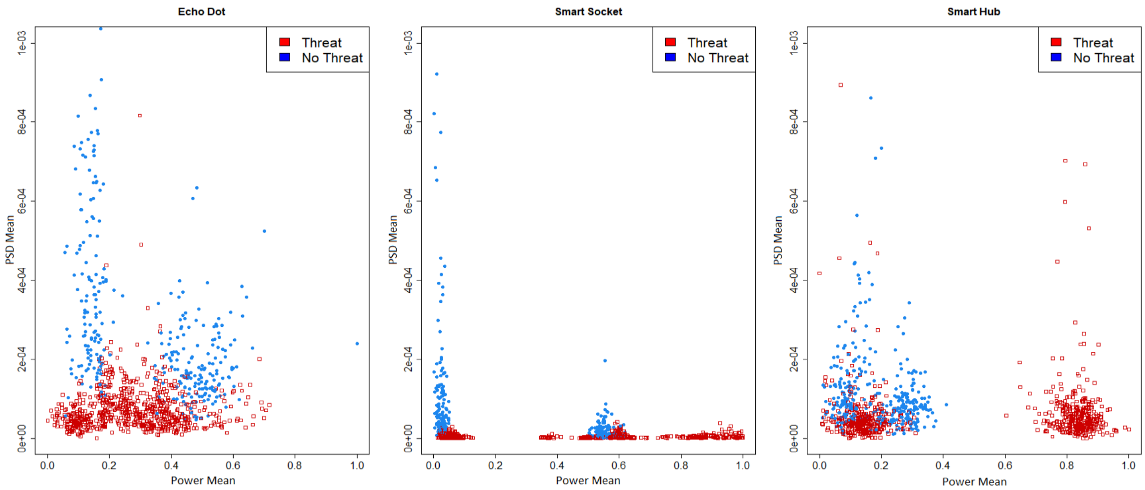
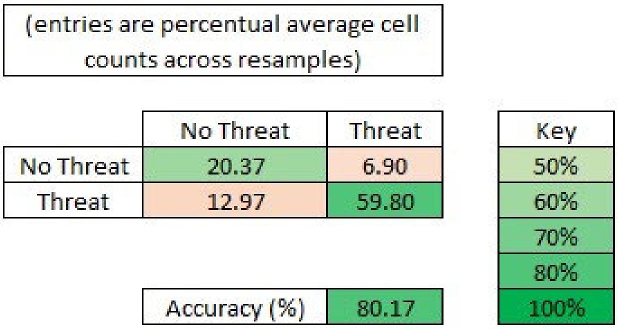


FIGURE 15. Device based cluster plot for threat detection accuracy.



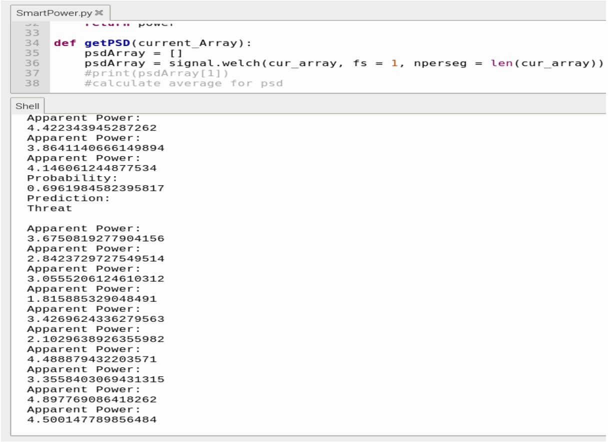
(entries are percentual average cell counts across resamples)

	No Threat	Threat
No Threat	20.37	6.90
Threat	12.97	59.80

Accuracy (%)	80.17

Key
50%
60%
70%
80%
100%

FIGURE 16. Confusion matrix for the security threat based classification accuracy.



```
    return power

def getPSD(current_Array):
    psdArray = []
    psdArray = signal.welch(cur_array, fs = 1, nperseg = len(cur_array))
    #print(psdArray[1])
    #calculate average for psd
```

```
Apparent Power:
4.422343945287262
Apparent Power:
3.8641140666149894
Apparent Power:
4.146061244877534
Probability:
0.6961984582395817
Prediction:
Threat

Apparent Power:
3.6750819277904156
Apparent Power:
2.8423729727549514
Apparent Power:
3.0555206124610312
Apparent Power:
1.815885329048491
Apparent Power:
3.4269624336279563
Apparent Power:
2.1029638926355982
Apparent Power:
4.488879432203571
Apparent Power:
3.3558403069431315
Apparent Power:
4.897769086418262
Apparent Power:
4.500147789856484
```

FIGURE 17. Shell of thonny IDE on raspberry Pi zero.

returned as a value between 0 and 1. Any probability value that is above the set threshold of 0.6 is returned as a threat, resulting in an email notification being sent via Node-RED.

B. NODE-RED

To alert users to the detected threat we implemented Node-RED. Node-RED allows for easy email notification from the Raspberry Pi Zero and starts after the boot-up of the device [53]. Figure 18 shows the Node-RED implementation for email notification. Node-RED constantly monitors the Alert.txt file that contains data on whether or not a threat has been detected.

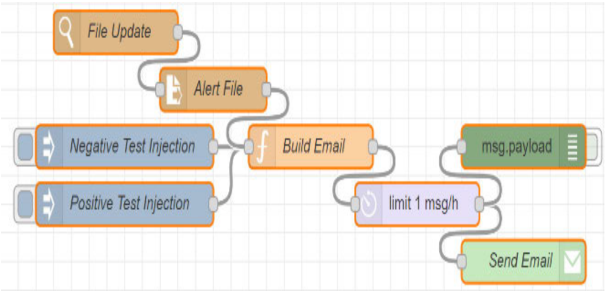


FIGURE 18. Node-RED implementation for email notification.

Once an update has been detected the file is read by the "Alert File" node and the contents are sent to the "Build Email" node. If the contents of the file show a threat, then the Email is constructed and sent. A 1msg/h delay is used as to not spam notifications. Along with this, nodes are included for debugging purposes. The nodes "Negative Test Injection" and "Positive Test Injection" are used to assure the email will be built and sent to the correct email address when a negative and positive signal is received. The node msg.payload is a debug node as well that posts the payload of the email to the debug terminal when the email is sent.

C. CONSENSUS BETWEEN MODEL AND IMPLEMENTATION

During the tests of the implementation of the prediction model, it was noticed that the predictions were not matching the actual state of the devices. This is because the PSD values obtained for the model building process were based on 21600 points of data (6 hours). The PSD values obtained for the implementation tests were based on 10 points of data

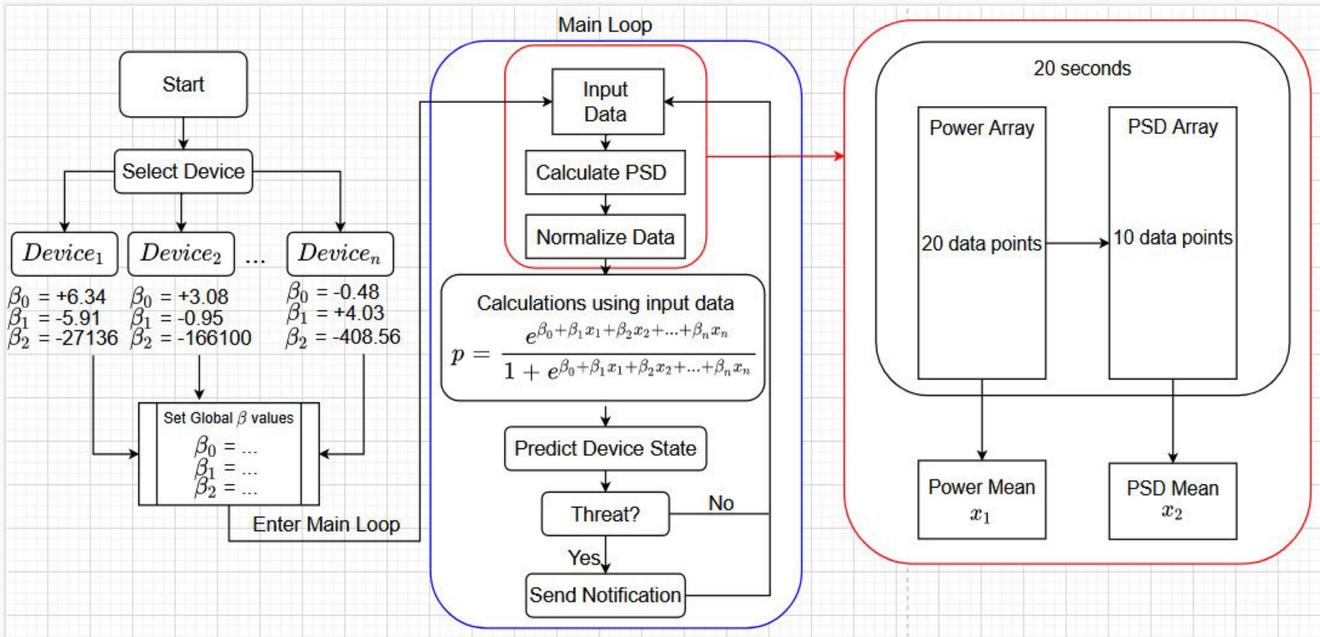


FIGURE 19. Flow diagram of software implementation of our proposed approach.

(10 seconds). Because PSD is based on the entirety of the data, the large difference in data points in how the model was built and how the model was implemented, as shown in Figure 19, leads to different PSD values.

For the implementation to reflect the predicted accuracy, the implementation needs to reflect the build model. In other words, the implementation also needs to test 6 hours of data. This is not suitable for near real-time analysis of data.

Reducing the discrepancy between the accuracy of the implemented model and the predicted accuracy will be a focus of future research.

VIII. CONCLUSION

In this research, a robust, dynamic cyber and physical threat detection approach utilizing statistical signal processing and multivariate modeling was presented. The proposed approach can detect cyber-attacks through behavioral power profiling of a wireless smart home hardware-in-the-loop (HIL) devices. This research provides a non-invasive system that allows users to better understand the security state of a CPS-based system. The results from different sensor data sets are also presented to show that this approach provides a high rate of classification correctness in power profiling for security monitoring. The system may also have multiple applications such as security analytics of a smart grid or a smart home energy management system.

In the future, we plan to include a general model, a redesign of the user interface with the device, and a focus on small data sets. Since our current model is built based on specific devices, a more general model is needed to cover all devices and draw a generalized relation between attacks and power usage. As for the user interface, the current smartphone implementation will be substituted for a web hosting implementation that would allow the user to access the device from anywhere. We plan on focusing much more on smaller data sets. In our current work, we collected data over longer time periods with greater amounts of data points. To more accurately build a model we must use smaller data sets that are more representative of our real time application.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for reviewing earlier drafts of this paper.

CONFLICT OF INTEREST

The authors declare no conflict of interest regarding this paper.

REFERENCES

[1] J.-P.-A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, "Cyber-physical systems security: Limitations, issues and future trends," *Microprocessors Microsyst.*, vol. 77, Sep. 2020, Art. no. 103201, doi: 10.1016/j.micpro.2020.103201.

[2] B. Ali and A. Awad, "Cyber and physical security vulnerability assessment for IoT-based smart homes," *Sensors*, vol. 18, no. 3, p. 817, Mar. 2018, doi: 10.3390/s18030817.

[3] N. K. Suryadevara, S. C. Mukhopadhyay, *Smart Homes: Design, Implementation and Issues*. Cham, Switzerland: Springer, 2015.

[4] M. Talal, A. A. Zaidan, B. B. Zaidan, A. S. Albahri, A. H. Alamoodi, O. S. Albahri, M. A. Alsalem, C. K. Lim, K. L. Tan, W. L. Shir, and K. I. Mohammed, "Smart home-based IoT for real-time and secure remote health monitoring of triage and priority system using body sensors: Multi-driven systematic review," *J. Med. Syst.*, vol. 43, no. 3, Mar. 2019, doi: 10.1007/s10916-019-1158-z.

[5] I. You, G. Pau, V. M. Salerno, and V. Sharma, "Special issue Internet of Things for smart homes," *Sensors*, vol. 19, no. 19, p. 4173, Sep. 2019, doi: 10.3390/s19194173.

[6] P. Lynggaard and K. Skouby, "Complex IoT systems as enablers for smart homes in a smart city vision," *Sensors*, vol. 16, no. 11, p. 1840, Nov. 2016, doi: 10.3390/s16111840.

[7] Sánchez-de-Rivera, Bordel, Alcarria, and Robles, ''Enabling efficient communications with resource constrained information endpoints in smart homes,'' *Sensors*, vol. 19, no. 8, p. 1779, Apr. 2019, doi: 10.3390/s19081779.

[8] Z. Chang, ''IoT device security: Looking out risks amd threats to smart homes,'' Trend Micro Res., Irving, TX, USA, Jul. 2019.

[9] *The IoT in 2030: 24 Billion Connected Things Generating $1.5 Trillion*, IoT Bus. News, Reading, U.K., May 2020.

[10] *Wisekey IoT Early Warning System is Now Allowing National COVID-19 App Development Teams to Build on top of the 'Foresight' Platform*, Market Watch, New York, NY, USA, Aug. 2020.

[11] S. Saber and O. Menes, *Artificial Intelligence and the Future for Smart Homes*, document IFC 78, Feb. 2020.

[12] F. S. Dantas Silva, E. Silva, E. P. Neto, M. Lemos, A. J. Venancio Neto, and F. Esposito, ''A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios,'' *Sensors*, vol. 20, no. 11, p. 3078, May 2020, doi: 10.3390/s20113078.

[13] J. Galeano-Brajones, J. Carmona-Murillo, J. F. Valenzuela-Valdés, and F. Luna-Valero, ''Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: An experimental approach,'' *Sensors*, vol. 20, no. 3, p. 816, Feb. 2020, doi: 10.3390/s20030816.

[14] S. Kazmi, N. Javaid, M. J. Mughal, M. Akbar, S. H. Ahmed, and N. Alrajeh, ''Towards optimization of Metaheuristic algorithms for IoT enabled smart homes targeting balanced demand and supply of energy,'' *IEEE Access*, vol. 7, pp. 24267–24281, 2019.

[15] J., King and A. I. Awad, ''A distributed security mechanism for resource-constrained IoT devices,'' *Informatica*, vol. 40, pp. 133–143, May 2016.

[16] M., Miller, *The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities are Changing the World*. Indianapolis, IN, USA: Que, 2015.

[17] I. Froiz-Míguez, T. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, ''Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes,'' *Sensors*, vol. 18, no. 8, p. 2660, Aug. 2018, doi: 10.3390/s18082660.

[18] M. Nolich, D. Spoladore, S. Carciotti, R. Buqi, and M. Sacco, ''Cabin as a home: A novel comfort optimization framework for IoT equipped smart environments and applications on cruise ships,'' *Sensors*, vol. 19, no. 5, p. 1060, Mar. 2019, doi: 10.3390/s19051060.

[19] K. Blake, ''Coronavirus impacts global smart home market, slashing 2020 device revenue outlook by $19.5 billion,'' Market Insight, Noida, Uttar Pradesh, Tech. Rep., Mar. 2020.

[20] A. Molina Zarca, J. B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, and P. Gouvas, ''Security management architecture for NFV/SDN-aware IoT systems,'' *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8005–8020, Oct. 2019, doi: 10.1109/JIOT.2019.2904123.

[21] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, ''IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices,'' *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: 10.1109/JIOT.2019.2935189.

[22] L. D. Xu, W. He, and S. Li, ''Internet of Things in industries: A survey,'' *IEEE Trans. Ind. Informat.*, vol. 10, no. 5, pp. 2233–2243, May 2014.

[23] S. Talari, M. Shafie-khah, P. Siano, V. Loia, A. Tommasetti, and J. Catalào, ''A review of smart cities based on the Internet of Things concept,'' *Energies*, vol. 10, no. 4, p. 421, Mar. 2017.

[24] J. Ibarra-Esquer, F. González-Navarro, B. Flores-Rios, L. Burtseva, and M. Astorga-Vargas, ''Tracking the evolution of the Internet of Things concept across different application domains,'' *Sensors*, vol. 2017, p. 17, Oct. 1379.

[25] M. Swan, ''Sensor mania! The Internet of Things, wearable computing, objective metrics, and the quantified self 2.0,'' *J. Sens. Actuators Netw.*, vol. 1, pp. 217–253, Mar. 2012.

[26] A. Gupta and R. K. Jha, ''A survey of 5G network: Architecture and emerging technologies,'' *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[27] B. L. Risteska Stojkoska and K. V. Trivodaliev, ''A review of Internet of Things for smart home: Challenges and solutions,'' *J. Cleaner Prod.*, vol. 140, pp. 1454–1464, Jan. 2017.

[28] P. K. Khatua, V. K. Ramachandaramurthy, P. Kasinathan, J. Y. Yong, J. Pasupuleti, and A. Rajagopalan, ''Application and assessment of Internet of Things toward the sustainability of energy systems: Challenges and issues,'' *Sustain. Cities Soc.*, vol. 53, Feb. 2020, Art. no. 101957.

[29] D. Brunelli, I. Minakov, R. Passerone, and M. Rossi, ''Smart monitoring for sustainable and energy-efficient buildings: A case study,'' in *Proc. IEEE Workshop Environ., Energy, Struct. Monitor. Syst. (EESMS) Process.*, Jul. 2015, pp. 186–191.

[30] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M. B. Kjárgaard, M. Srivastava, and K. Whitehouse, ''Brick: Towards a unified metadata schema for buildings,'' in *Proc. 3rd ACM Int. Conf. Syst. Energy-Efficient Built Environ.*, Nov. 2016, pp. 41–50.

[31] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, ''Green data center with IoT sensing and cloud-assisted smart temperature control system,'' *Comput. Netw.*, vol. 101, pp. 104–112, Jun. 2016.

[32] L. Pocero, D. Amaxilatis, G. Mylonas, and I. Chatzigiannakis, ''Open source IoT meter devices for smart and energy-efficient school buildings,'' *HardwareX*, vol. 1, pp. 54–67, Apr. 2017.

[33] Z. Chen, C. Ye, J. Yuan, and D. Han, ''MGF-based mutual approximation of hybrid fading: Performance of Wireless/Power line relaying communication for IoT,'' *Sensors*, vol. 19, no. 11, p. 2460, May 2019, doi: 10.3390/s19112460.

[34] R. Marin-Perez, I. Michailidis, D. Garcia-Carrillo, C. Korkas, E. Kosmatopoulos, and A. Skarmeta, ''PLUG-N-HARVEST architecture for secure and intelligent management of near-zero energy buildings,'' *Sensors*, vol. 19, no. 4, p. 843, Feb. 2019, doi: 10.3390/s19040843.

[35] O. Ardakanian, A. Bhattacharya, and D. Culler, ''Non-intrusive techniques for establishing occupancy related energy savings in commercial buildings,'' in *Proc. 3rd ACM Int. Conf. Syst. Energy-Efficient Built Environ.*, Nov. 2016, pp. 21–30.

[36] J. Majumder, A. J. Miller, C. Veilleux and A. Asif, ''Smart-power: A smart cyber-physical system to detect IoT security threat through behavioral power profiling,'' in *Proc. IEEE 44th Annu. Comput. Softw. Appl. Conf.*, Jul. 2020, pp. 1041–1049.

[37] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, ''A P2P botnet detection scheme based on decision tree and adaptive multilayer neural networks,'' *Neural Comput. Appl.*, vol. 29, no. 11, pp. 991–1004, Jun. 2018.

[38] S. Saad, ''Detecting P2P botnets through network behavior analysis and machine learning,'' in *Proc. 9th Annu. Int. Conf. Privacy, Secur. Trust*, Montreal, QC, Canada, 2011, pp. 174–180, doi: 10.1109/PST.2011.5971980.

[39] M. O'Neill, ''Insecurity by design: Today's IoT device security problem,'' *Engineering*, vol. 2, no. 1, pp. 48–49, Mar. 2016, doi: 10.1016/j.eng.2016.01.014.

[40] J. D. Bokefode, A. S. Bhise, P. A. Satarkar, and D. G. Modani, ''Developing a secure cloud storage system for storing IoT data by applying role based encryption,'' *Procedia Comput. Sci.*, vol. 89, pp. 43–50, Dec. 2016, doi: 10.1016/j.procs.2016.06.007.

[41] S. Kim and I. Lee, ''IoT device security based on proxy re-encryption,'' *J. Ambient Intell. Hum. Comput.*, vol. 9, no. 4, pp. 1267–1273, Aug. 2018, doi: 10.1007/s12652-017-0602-5.

[42] S. Kumar, Y. Hu, M. P. Andersen, R. A. Popa, and D. Culler, ''JEDI: Many-to-many end-to-end encryption and key delegation for IoT,'' in *Proc. 28th USENIX Secur. Symp.*, 2019, pp. 1–5.

[43] R. Bridges, J. Hernandez Jimenez, J. Nichols, K. Goseva-Popstojanova, and S. Prowell, ''Towards malware detection via CPU power consumption: Data collection design and analytics,'' in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2018, pp. 1680–1684, doi: 10.1109/TrustCom/BigDataSE.2018.00250.

[44] M. Park, H. Oh, and K. Lee, ''Security risk measurement for information leakage in IoT-based smart homes from a situational awareness perspective,'' *Sensors*, vol. 19, no. 9, p. 2148, May 2019, doi: 10.3390/s19092148.

[45] S. Shane, C. B. Ransford, A. Rahmati, S. Guineau, J. Sorber, K. Fu, and W. Xu, ''WattsUpDoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices,'' in *Proc. USENIX Workshop Health Inf. Technol.*, Aug. 2013, pp. 1–8.

[46] T. Wolf, H. K. Chandrikakutty, K. Hu, D. Unnikrishnan, and R. Tessier, ''Securing network processors with high-performance hardware monitors,'' *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 6, pp. 652–664, Nov. 2015.

[47] A. Manzoor, P. Porambage, M. Liyanage, M. Ylianttila, and A. Gurtov, ''DEMO: Mobile relay architecture for low-power IoT devices,'' in *Proc. IEEE 19th Int. Symp. Mobile Multimedia Networks*, Jun. 2018, pp. 14–16.

[48] J. Qadri, H. M. Chen, and J. Blasco, ''A review of significance of energy-consumption anomaly in malware detection in mobile devices,'' *Int. J. Cyber Situational Awareness*, vol. 1, no. 1, pp. 210–230, Dec. 2016.

[49] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, and M. Ismail, "Energy-efficient wireless sensor networks for precision agriculture: A review," *Sensors*, vol. 17, no. 8, p. 1781, Aug. 2017.

[50] E. Seo, H. Kim, and T.-M. Chung, "Profiling-based classification algorithms for security applications in Internet of Things," in *Proc. IEEE Int. Congr. Internet Things (ICIOT)*, Jul. 2019, pp. 138–146.

[51] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[52] S. L. Miller and D. Childers, "Power spectral density," in *Probability and Random Processes*. 2004.

[53] *Node-RED: Low-Code Programming for Event-Driven Applications*. Accessed: Nov. 6, 2020. [Online]. Available: https://nodered.org/

AKM JAHANGIR ALAM MAJUMDER (Member, IEEE) received the Ph.D. degree in computational sciences from Marquette University, Milwaukee Wisconsin, in 2016. He is currently an Assistant Professor with the Division of Mathematics and Computer Science, University of South Carolina Upstate, SC. His research explores the development of embedded Internet of Things (IoT) systems and cyber-physical-systems (CPS) with special interests on CPS design automation, model-based design, development of mobile computing technologies, and CPS security. He has developed integrated CPS technologies employing software models as well as the physical system which uses a standard smartphone in conjunction with sensors in a custom embedded CPS. From 2016 to 2018, he was a Visiting Assistant Professor with the Department of Electrical and Computer Engineering (ECE), Miami University, Oxford, OH. Before coming to Marquette, he worked as an Assistant Professor with the Department of Electrical and Electronic Engineering (EEE), Ahsanullah University of Science and Technology (AUST), Bangladesh. He was a recipient of the Richard W. Jobling Fellowship Award from Marquette University for outstanding Ph.D. student, in 2013 and 2014. He also received the Teaching Assistant of the Year Award for 2013–2014 from the Department of MSCS, Marquette University, for his well-deserved reputation as an effective and caring TA.

CHARLES B. VEILLEUX was born in Greenville, South Carolina, USA, in 1998. He is currently a Research Assistant studying computer science with the University of South Carolina at Upstate. His research interests include computer security, the Internet of Things, and automation.

JARED D. MILLER was born in Greenville, South Carolina, USA, in 1998. He is currently pursuing the bachelor's degree in computer information systems with USC Upstate. He is also a Research Assistant with the Computer Science Department, USC Upstate.