Light Auditor: Power Measurement can tell Private Data Leakage through IoT Covert Channels

Woosub Jung
William & Mary
Williamsburg, USA
wjung01@wm.edu

Kailai Cui
William & Mary
Williamsburg, USA
kcui@wm.edu

Kenneth Koltermann
William & Mary
Williamsburg, USA
khkoltermann@wm.edu

Junjie Wang
William & Mary
Williamsburg, USA
jwang51@wm.edu

ChunSheng Xin
Old Dominion University
Norfolk, USA
cxin@odu.edu

Gang Zhou
William & Mary
Williamsburg, USA
gzhou@cs.wm.edu

ABSTRACT

Despite many conveniences of using IoT devices, they have suffered from various attacks due to their weak security. Besides well-known botnet attacks, IoT devices are vulnerable to recent covert-channel attacks. However, no study to date has considered these IoT covert-channel attacks. Among these attacks, researchers have demonstrated exfiltrating users' private data by exploiting the smart bulb's capability of infrared emission.

In this paper, we propose a power-auditing-based system that defends the data exfiltration attack on the smart bulb as a case study. We first implement this infrared-based attack in a lab environment. With a newly-collected power consumption dataset, we pre-process the data and transform them into two-dimensional images through Continous Wavelet Transformation (CWT). Next, we design a two-dimensional convolutional neural network (2D-CNN) model to identify the CWT images generated by malicious behavior. Our experiment results show that the proposed design is efficient in identifying infrared-based anomalies: 1) With much fewer parameters than transfer-learning classifiers, it achieves an accuracy of 88% in identifying the attacks, including unseen patterns. The results are similarly accurate as the sophisticated transfer-learning CNNs, such as AlexNet and GoogLeNet; 2) We validate that our system can classify the CWT images in real time.

CCS CONCEPTS

• **Security and privacy** → *Malware and its mitigation.*

KEYWORDS

IoT privacy, power auditing, convolutional neural networks, covert channel

ACM Reference Format:
Woosub Jung, Kailai Cui, Kenneth Koltermann, Junjie Wang, ChunSheng Xin, and Gang Zhou. 2022. Light Auditor: Power Measurement can tell Private Data Leakage through IoT Covert Channels. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3560905.3568535

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11.
https://doi.org/10.1145/3560905.3568535

1 INTRODUCTION

Recent adversaries have exploited the vulnerabilities of IoT devices. For example, IoT botnets have become more popular among adversaries since the Mirai attack appeared in 2016. Since then, many variants have attacked vulnerable IoT devices [34]. Despite the continuous appearance of IoT botnets, a concrete solution has not been found against malicious behavior because of the weak security of IoT devices. The concerns on IoT security are related to the lack of heterogeneous solutions; IoT devices are not being controlled by just a few standard operating systems or protocols. For example, 84 different IoT devices/vendors were found to engage in the Mirai bots, which are related to more than 300 different communication protocols and platforms [3]. Furthermore, IoT devices are often not capable of deploying sophisticated detection algorithms. Thus, it is nearly impossible to deploy a universal solution to resource-constrained IoT devices.

Besides the well-known IoT botnets, the IoT security literature has also examined recent side-channel attacks that exploit covert channels using IoT devices. In this attack scenario, instead of being exploited as malicious bots, IoT devices were also used for covert-channel attacks in order to exfiltrate users' private data. For example, Maiti and Jadliwala established a potential IoT attack where adversaries exfiltrate users' private data through infrared-enabled smart bulbs [41]. This exfiltrated data such as banking accounts, authentication information, or private photos that can be observed by nearby receivers with infrared ability. Another example of IoT-related covert-channel attacks is the use of ultrasound transmission. Gao et al. [10] identified a new IoT covert-channel attack where private data can be transmitted through an ultrasound medium and observed by gyroscope sensors. Likewise, recent data exfiltration attacks have utilized IoT covert channels to leak users' private data from IoT devices [1, 7, 8, 45]. Therefore, we have to address the increasing need for protection against covert-channel attacks on IoT devices.

No thoughtful study to date, however, has considered these kinds of stealthy attacks that exploit covert channels in IoT devices. In addition, existing network-based detection methods cannot defend IoT devices against the attack since it exploits invisible channels,

and no data is transmitted through the network. To address this issue, we aim to defend against the prominent example that exploits the infrared-enabled smart bulb [41], as a case study. Accordingly, our research problem in this paper is as follows:

- How can we protect private data from the data exfiltration attack that exploits infrared emission?

While answering this question means that we are the first to tackle the emerging IoT data-exfiltration attacks, we focus on the case study in which the attacker leaks data through the infrared emission of smart bulbs. To make it possible, we face the following research challenges.

- How can we monitor the infrared channel generated by smart bulbs?
- How can we model the behavior through the covert channel?
- How well can the model identify the data exfiltration attacks?

To answer the first question, we utilize a power-auditing ability in IoT environments. Several studies have shown that power auditing techniques are effective to enhance IoT security against IoT botnets. For example, Jung et al. [29, 30] designed a deep learning classifier to identify IoT botnets via power consumption modeling. However, those studies were not designed to detect or protect against information leakage attacks through covert channels. In this paper, we monitor the power consumption data generated by the smart bulb's infrared emission in order to defend against data exfiltration attacks.

To answer the second question, we design a two-dimensional convolutional neural network (2D-CNN) classifier that can identify the data exfiltration events via infrared emissions. We first convert raw power consumption data to Continuous Wavelet Transform (CWT) [53] images as our input instances. We made this design choice because CWT images include time- and frequency-domain information, whereas raw power consumption data only contains time-domain data. Then, we tuned the hyper-parameters of the proposed CNN in order to classify behaviors of the smart bulbs, including malicious attacks.

To answer the third question, we validate whether the proposed classifier can distinguish covert-channel attacks from typical behaviors on smart bulbs. In five-fold cross-validation tests, the proposed CNN classifier achieves an accuracy of approximately 88%, which is close to other transfer-learning classifiers with a higher number of parameters. Moreover, our CNN model achieves almost the same results in identifying unseen exfiltration attacks, including different encoding schemes and different bitrates. Lastly, our CNN classifies input data 7 times faster than AlexNet and 10 times faster than GoogLeNet.

Overall, we propose the first detection approach against the infrared-based attack that leaks private data. The experimental results suggest that our system is lightweight and defends the smart bulb well from infrared covert-channel attacks, including unseen attack patterns.

The rest of the paper is organized as follows: Section 2 provides a background of covert channel attacks in IoT environments. In Section 3, we define a new threat model that uses brute-force attacks and exploits covert channels in IoT environments. Section 4 summarizes the overview of our proposed system, LightAuditor. Section 5 then introduces a power auditing method that monitors

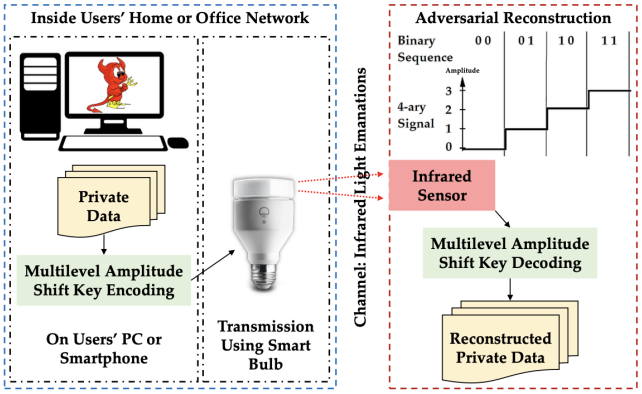


Figure 1: Target Infrared Attack: Light Ears [41]

infrared channels. Section 6 describes a CNN-based attack detection model. In Section 7, we demonstrate the performance of our classifier compared to other transfer-learning CNNs. Sections 8 and 9 provide related work and discussion. Finally. Section 10 concludes this paper.

2 BACKGROUND

In Section 2.1, we summarize covert-channel attacks that exploit out-of-band channels for data exfiltration. Section 2.2 introduces an emerging IoT attack that we aim to defend against.

2.1 Out-of-band Covert-channel Attacks

A covert channel is a communication channel that is not originally intended for data transfer but is exploited by attackers to exfiltrate data [35]. The purpose of this covert-channel attack is secret communication without modifying existing hardware. Thus, both the transmitter and receiver should agree upon a modulation and demodulation scheme so that others cannot overhear information without knowledge of this scheme. The history of covert-channel attacks began with Simmons' prisoners' problem from the 1980s. Since then, a vast number of adversaries have created clever approaches for data exfiltration through covert channels [1, 7, 8, 45].

Among the covert channels, out-of-band channels typically run on physically separated machines and share no common resources aside from a physical medium [6]. These channels have been exploited by adversaries to leak the private information of users. Since private data is being transmitted over a shared physical medium in this type of attack, adversaries have utilized either covert-acoustic,

(a) Original Image (b) Reconstructed Image at 5m

Figure 2: Exfiltrated Data Samples in the Target Attack [41]

covert-light, or covert-vibration channels. For example, if an out-of-band channel attack uses an ultrasonic medium, the transmitter and the receiver must already have the capability for ultrasonic channel establishment [21]. The attackers then communicate with each other over the shared ultrasonic channel, following any previously agreed-upon encoding schemes.

2.2 Target Covert-channel Attack: Light Ears

Recently, several out-of-band covert channel attacks exploited IoT devices' sensing capability. Among them, we chose a target attack created by Maiti and Jadliwala that exploits a covert-light channel [41], as a case study. In that paper, they proposed a malicious attack that leaks users' private data through the infrared emission ability of smart bulbs.

Figure 1 shows the core procedures of this target attack. In their covert-light attack, adversaries enter users' workstations that store sensitive information; for example, banking accounts, authentication information, or private images. Next, the adversaries transform the target information into encoded data packets. The authors then periodically change the smart bulb's infrared amplitude to the corresponding encoding values. Setting the infrared amplitude can be considered data exfiltration since any infrared-enabled receiver is able to detect and decode the amplitude changes for obtaining original private data. In Figure 2, the original image (Figure 2a) can be leaked by setting the amplitude of the smart bulb and be restructured at the distance of five meters (Figure 2b).

Note that the receiver must have the ability to receive the infrared signal and understand the transmitted encoding scheme. Although this requires an infrared-enabled receiver to interpret the encoded signals, this is a common requirement for all out-of-band covert channel attacks [6], and it is not difficult to receive and decode the data through current and future IoT devices [48].

Therefore, the need to defend against covert-channel attacks that use IoT devices has become greater as the IoT market has increased substantially. Thus far, however, there are no practical auditing methods to determine whether an IoT device is being compromised for covert communication. To tackle this situation, we design a new intrusion detection system that utilizes power auditing to monitor smart bulbs' infrared emission.

3 ATTACK MODEL: IOT DATA EXFILTRATION

In Section 3.1, we introduce a new IoT attack model that consists of a brute-force attack and a cover-channel attack for data exfiltration.

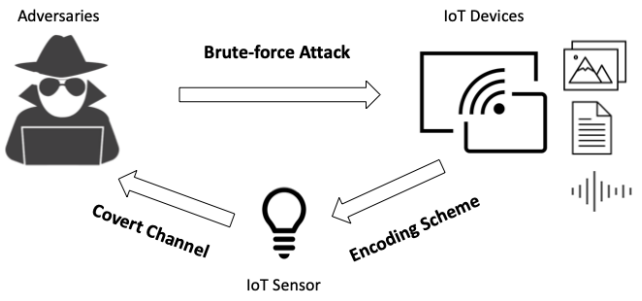


Figure 3: Attack Model Procedure — Private data can be leaked remotely via brute-force and covert-channel attacks

Table 1: Possible IoT Data Exfiltration Attacks

Intrusion Method	Covert Channel	Channel Availability	Exfiltration Bandwidth	Reference
Brute-force Attacks [55]	Acoustic	High	Medium	[19, 20]
	Ultrasonic	High	High	[10, 21]
	Infrared	High	Medium	[14, 41]
	Electromagnetic	Medium	Medium	[15–17]
	Magnetic	Medium	Medium	[13, 25]
	Thermal	Low (overnight)	Low	[18]
	Optical	Low (user absence)	High	[23, 24]

In Section 3.2, we describe our testbed environment and the attack implementation that leaks private data through infrared channels.

3.1 IoT Attack Model

As the IoT device market grows, adversaries have more chances to exploit the vulnerabilities of IoT devices. Among the emerging IoT-related attacks, private data leakage is one of the most hideous experiences that users may confront in IoT environments. Despite the existence of several IoT covert-channel attacks, it has not been fully examined how adversaries could exploit various sensors of IoT devices for data leakage, and thus this type of new attack is still considered trivial.

In this situation, we define a new IoT attack model that can leak private data through covert channels. Although the previous covert-channel attacks exploited out-of-band channels to leak private data, it was unclear how the adversaries could get into IoT devices prior to their data exfiltration parts. For example, in the covert-light channel attack [41], Maiti and Jadliwala assumed that this attack can happen where adversaries reside in the user's network. The authors then executed the attack on a Windows PC in the same network. Yet, this assumption is weak in that how the adversaries broke into the private network was unexplained.

We introduce a more practical way of the exfiltration attack in IoT environments. The new IoT attack model comprises an intrusion part and a data exfiltration method; the remote intrusion happens prior to the data exfiltration in this attack. According to the categorization of anomaly behavior [28], this new IoT attack can be classified as a temporal combination of the brute-force attack and the covert-channel attack. This attack model is also feasible because more than half of IoT botnets have utilized brute-force attacks in order to access vulnerable IoT devices [32], such as Mirai [3]. Brute-force attacks have been effective in IoT environments since a vast number of IoT devices remain in factory settings for their accounts and passwords. Therefore, it is likely that future adversaries can enter IoT devices using brute-force attacks prior to their data exfiltration attacks.

(a) Smart Bulb **(b) Infrared Capability**

Figure 4: Infrared-enabled Smart Bulb Testbed

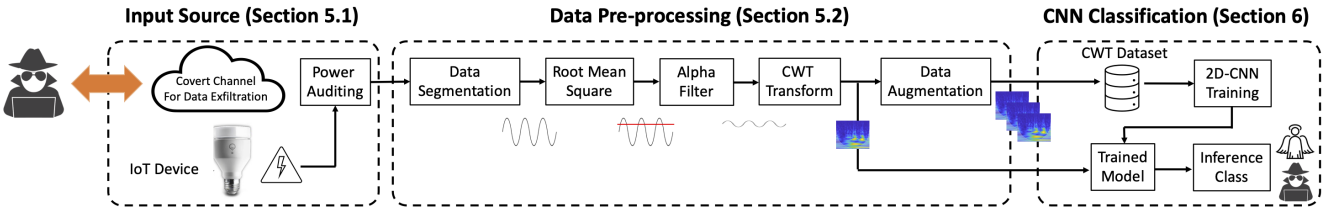


Figure 5: Overview of Light Auditor

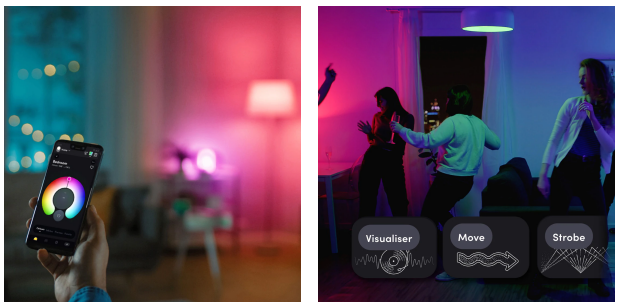
Furthermore, data exfiltration attacks have often been conducted by malicious insiders, privileged users, or former employees [11]. In any cases, the IoT data-exfiltration attack can be done between the compromised source and the outside receiver without any physical contact. Figure 3 illustrates that an attacker first enters an IoT device through brute-force attacks. The adversary can then exfiltrate users' private data through established covert channels. Table 1 also shows potential combinations of the attack cases in this new model. In this paper, we focus on the specific attack case, the covert-light channel attack, to demonstrate the feasibility of our detecting solution.

3.2 Case Study Implementation

Before developing our detecting solution, we first implemented the IoT attack model in our lab testbed setting. For prototyping the covert-channel attack, we used a smart bulb that has infrared emission capability. In the smart bulb market, the LIFX+ night vision device [40] is one of the few smart bulbs that support infrared transmission. The manufacturer provides a user application for users to remotely control the bulbs and HTTP-based APIs for developers [39] to customize its functions. Exploiting these APIs, adversaries could establish covert channels to exfiltrate private data.

Figure 4a shows the LIFX+ smart bulb connected to alternating current (AC) power. Figure 4b shows when the smart bulb emits infrared signals. Although the infrared emission can be seen by cameras, this infrared emission is invisible to the human naked eye. This means that the data exfiltration attacks involved with infrared smart bulbs are stealthy and thus hard to detect. Besides the infrared transmission, the LIFX+ smart bulb also provides several other features, such as dimmable LEDs and music visualizer features, in which the visualizer changes light colors, amplitudes, and frequencies to generate responsive light based on surrounding music. Figure 6 illustrates these features, respectively.

As described in the attack scenario in Section 3.1, adversaries can take over IoT devices and then remotely control the smart bulb to exfiltrate users' private data. In order to implement this adversary, we used a Raspberry Pi 3 device as a prototype platform, but smartphones can also serve as an adversary platform. On either platform, victims' data can be retrieved through infrared transmission.



(a) Dimmable LEDs **(b) Visualizer**

Figure 6: Smart Bulb Features

Using the HTTP APIs provided, the adversary was able to set the amplitude of the infrared light between 0 and 65, 535. We wrote a python script that encodes a target image file to Amplitude-shift Keying (ASK)-encoded data. For example, a sample picture of 128×128 pixels can be converted into a 128×128 two-dimensional array where each pixel has a value between 0 and 255. Thus, pixel value 100 would be mapped to infrared amplitude value $25,700$ ($100 \div 255 \times 65,535$). By doing so, attackers can exfiltrate private data values sequentially through the covert-light medium. The python source codes of this attack will be available for research use only.

It should also be noted that through this covert-light attack, any type of data can be transmitted, such as text or voice, whereas voice or image data tolerate bit errors better than text data. Thus, the need to address how to defend against potential exfiltration attacks still remains. In Section 7, our experiments show how well our solution identifies other unseen data patterns.

4 LIGHT AUDITOR SYSTEM OVERVIEW

In this section, we briefly introduce a system overview. Figure 5 illustrates an overview of our system, LightAuditor. In our application scenario, an IoT smart bulb is being used in the user's network. Then, we aim to detect whether the smart bulb is being exploited for data exfiltration attacks via power consumption monitoring on the infrared channel.

As an input source for detection, we measure continuous raw power consumption data of the IoT device. This requires a one-to-one connection between the IoT device and the power auditing device. Thus, we develop a power auditing device that can measure the AC data of the connected IoT device. Our power auditing device and its testbed environment will be presented in Section 5.1.

With the measured raw power traces, we conduct several data pre-processing jobs for the system to predict inference results best. In the pre-processing phase, we first segment continuous data into a window size of 4 seconds as this window can include 2 to 3 light events. Once segmented, the fixed window size data contains the AC power data of the connected IoT device. Since AC power data is noisy, it may not be easy for a classifier to train the features of the raw AC data. In order to reduce the noise in the sinusoid wave noise, we apply the Root Mean Square (RMS) and Alpha filter methods. These procedures transform the raw AC data into more stable direct current (DC)-shaped power consumption data.

Next, we convert the one-dimensional DC power data into two-dimensional image instances by utilizing the CWT technique. This

conversion makes the input data include both time- and frequency-domain features. Thus, the CWT input instances can provide better classification results than using one-dimensional time-series input. In the sensing literature, several research have also utilized the frequency-domain conversion for their learning problems, such as EEG-to-CWT [44], ECG-to-CWT [5], Vibration-to-FFT [38], and Accelerometer-to-Images [57]. The above pre-processing procedures will be explained in Section 5.2. The impact of the CWT conversion will be discussed in Section 7.

Furthermore, we increase the number of input data instances before feeding them into our classifier. This data augmentation method is known to help avoid overfitting concerns, caused by small sample datasets [31]. Since the number of collected instances in our experiments may not be enough, we doubled the size of our dataset by applying an augmentation method during training, but not testing. Our augmented dataset will be presented in Section 5.3. The benefit of this data augmentation will also be described in Section 7.

Finally, once securing the power-based CWT dataset, we design a 2D-CNN model and validate its performance in our lab environment. We chose to use CNN for our system because we utilized CWT images as input data, and CNNs have shown its effectiveness for image classification. The design choices and the impact of the design factors are shown in Section 6, accordingly. We then compare our model with a 1D-CNN approach and other well-known transfer learning CNNs, such as AlexNet and GoogLeNet, in Section 7.

5 POWER CONSUMPTION DATA PROCESSING

In this section, we propose a power-auditing method to monitor the IoT infrared channel. Section 5.1 introduces hardware setup for power auditing. Once collected, power consumption data is processed first for further analysis, as described in Section 5.2. Then, Section 5.3 illustrates the collected power-trace dataset.

5.1 Testbed Hardware Setup

In IoT environments, it is often disregarded to consider measuring stealthy channels due to weak security and constrained resources of IoT devices. However, monitoring covert channels is the key to detecting stealthy attacks. According to Carrara et al. [6], all sensing channels should be audited regularly to detect malign communication. Nonetheless, monitoring specific covert-channels is hard because it requires modification to existing IoT devices. For example, in order to monitor infrared transmission from smart bulbs, users may need an extra set-up, i.e., an infrared receiver, which is inconvenient to normal users.

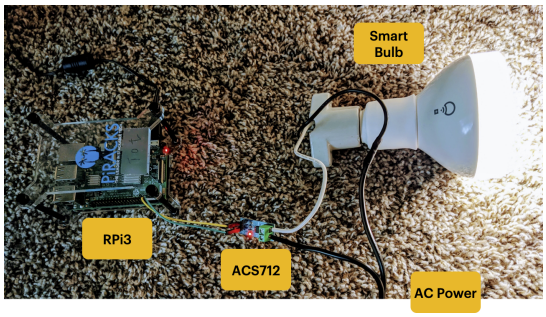


Figure 7: Power-Auditing Testbed Device

To tackle this issue, we propose to add a power-auditing feature for monitoring IoT devices, instead of building integrated solutions into existing IoT devices. In order to enable this power-auditing feature, we set up an IoT environment that can monitor the behavior of smart bulbs. A Raspberry Pi 3 device [46] is used to read and record real-time power consumption data of the connected smart bulb. As shown in Figure 7, we specifically leverage an ACS 712 current sensor [4] to measure the power consumption of IoT devices, i.e., an AC-powered smart bulb in our environment. This sensor can measure the current levels of AC-powered devices up to 30A. As such, our power-auditing prototype is universal in IoT environments because most IoT devices require less than this current level [12].

Overall, power consumption data of IoT devices can easily be measured by our proposed power-auditing hardware. This power-monitoring approach is a reasonable and practical solution for abnormal behavior detection, as validated in our experiments (Section 7). Moreover, measuring power consumption data is more practical than deploying detection algorithms into IoT devices even though our approach requires additional hardware for each IoT device. Our reasoning is that the proposed device can be considered a prototype of smart plugs [56], whose market also grows substantially [27]. In addition, smart plugs are already capable of providing and measuring power consumption data. Therefore, we leverage this market situation; in the future, most IoT devices may be connected individually by various forms of power-auditing devices [36].

5.2 Data Pre-processing Jobs

We then process the measured raw data before feeding it to our classifier. The raw data collected by our device is AC data. So, this raw signal includes various noise signals in the nature of the AC signal.

5.2.1 Root Mean Square. We first need to transform the raw data into DC data. In order to remove the AC fluctuation, we use a Root Mean Square (RMS) method. Equation 1 explains the RMS method that averages sinusoid wave data in each sine wave period. This V_{rms} value is known to be equivalent of producing the same average power as $V_{peak-peak} \times \frac{1}{2\sqrt{2}}$ of steady DC voltage [42]. As shown in Figure 8, the AC wave signals in blue sinusoid period are transformed to a constant voltage in red equal to the RMS value; we obtain the DC-shaped power consumption data after finding the RMS.

$$V_{rms} = V_{peak-peak} \times \frac{1}{2\sqrt{2}} \tag{1}$$

Figure 9 shows processed power consumption data when the IoT device is in idle. For example, Figure 9a illustrates the raw AC signal samples collected by our power-auditing device, and the sine

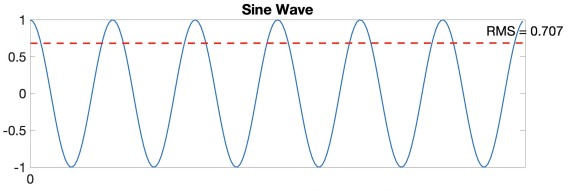
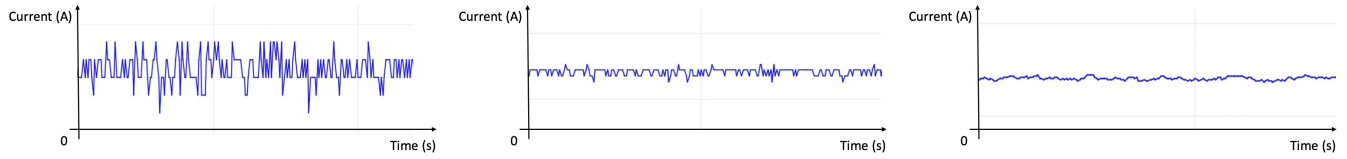


Figure 8: Root Mean Square of Sinusoidal AC Waves



(a) Raw Data (b) Root Mean Square (c) Alpha Filter ($\alpha = 0.2$)

Figure 9: Power Consumption Data from Pre-processing Steps

wave signals are already noisy even in the idle status. Figure 9b is the transformed data after applying the RMS method.

5.2.2 Alpha Filter. Despite the RMS conversion, the power traces still include spiky noises in them. Thus, we apply an alpha filter with an alpha value of 0.2 to smooth data. As shown in Equation 2, this method works as a moving average filter in which the filter takes into account α times processed signal value at the previous time and adds $(1-\alpha)$ times the raw signal value at the current time.

$$y(t) = \alpha \cdot y(t-1) + (1-\alpha) \cdot x(t) \tag{2}$$

By applying this filter, extreme spikes can be smoothed out. We set the α value to 0.2 because higher α values may lead to losing some important information about the current signals. Overall, the original raw signal is being processed into the smoothed DC signals. Figure 9c describes the filtered power consumption signals, which can now be used as our input signals for the next step.

5.2.3 Continuous Wavelet Transform. Finally, we apply CWT [53] to the filtered power consumption traces for better classification results. Our reasoning is that the converted CWT images can represent features of power consumption data in the frequency domain as well as in the time domain. Equation 3 explains how the CWT procedure works.

$$cwt(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t) \cdot \psi(\frac{t-\tau}{s}) dt \tag{3}$$

where:

τ = translation
s = scale
$\psi(t)$ = mother wavelet

Let $x(t)$ denote the filtered time-series power trace input. Then, it computes the inner products of $x(t)$ with a set of mother wavelets $\psi(t)$, where τ is the shift factor and s is the scale factor of the wavelet. In this equation, scale factors are inversely related to the frequency domain. For example, a small scale value makes a mother wavelet have a high frequency, whereas a large scale leads to a lower frequency mother wavelet. With the different scale and translation

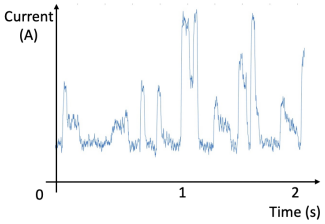
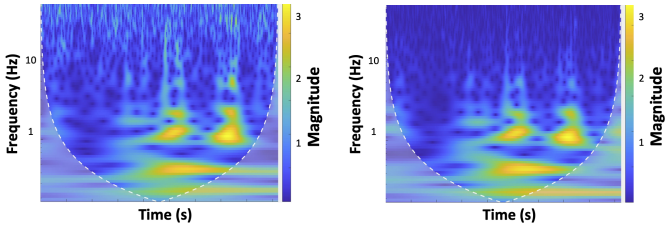


Figure 10: Filtered Power Trace Sample under Attack



(a) CWT Image w/o Alpha Filter (b) CWT Image w/ Alpha Filter

Figure 11: Converted CWT Image Samples under Attack

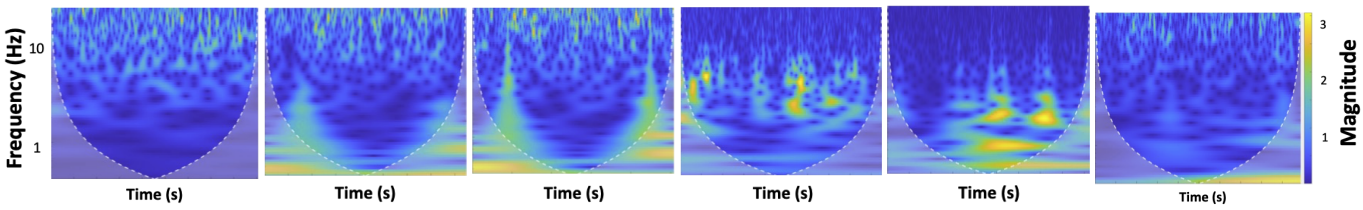
values, the CWT procedures compute convolutions between the input signals and the mother wavelets at points.

Figure 10 illustrates a source power data sample after the pre-processing. Note that, compared to Figure 9c, this sample is more spiky because it was captured under the data exfiltration attack case (Encoded infrared emission). Figure 11 then shows samples of the CWT conversion and the impact of the alpha filter in CWT images. As shown in Figure 11a, the converted CWT image contains higher amplitude values at a certain time and frequency. Figure 11b shows the CWT image with the alpha filter when α is 0.2. Since the alpha filter acts as a low-pass filter, noisy data in high frequency in Figure 11a were removed in Figure 11b. Overall, our system will benefit from using the CWT images as input data. In Section 7, we demonstrate that using the CWT input images performs better, compared to using the raw power data input.

5.3 Collected Power Trace Dataset

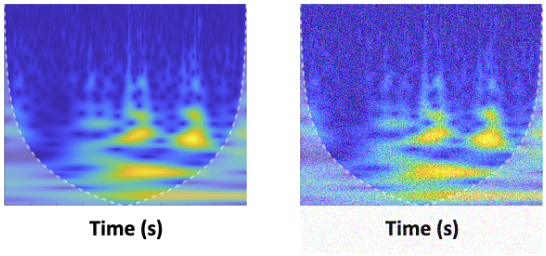
In our lab setting, we collected the power consumption data generated by a smart bulb. In total, we collected six types of different behaviors to detect abnormal behaviors. The smart bulb provides several functionalities that can be considered normal behaviors: Normal visible light, Infrared light, and Visualizer. The normal visible light is working by either a dimmable controller or an instant switch. The visualizer feature responds to background sound. As music varies, the color and its amplitude randomly change. Overall, we define five normal behaviors as follows: 1) Idle, 2) Visible Light (Dimmable), 3) Visible Light (Instant Switch), 4) Infrared Light, and 5) Visualizer.

We also define a malicious power consumption pattern of the covert-channel attack. As shown in Figure 11, the ASK-based data exfiltration attacks generate specific power consumption patterns. Both the raw power data (Figure 10) and the CWT image (Figure 11b) are distinguishable from the data of normal behaviors. This is because the frequency of amplitude change or drastic amplitude change leads to noticeable CWT image patterns.



(a) Idle (b) Light Instant Switch (c) Manual Infrared (d) Visualizer (e) Data Exfiltration (f) Dimmable Light

Figure 12: CWT Image Samples of Six Classification Classes



(a) Original Sample (b) Augmented Sample

Figure 13: Data Augmentation with Gaussian Noise

Including the Idle and Attack cases, we present CWT image samples of the six classes in Figure 12. The visualizer feature generates high-amplitude data but at different frequency levels (Figure 12d). All the Infrared (Figure 12c), the Light Switch (Figure 12b), and the Dimmable Light (Figure 12f) generate spreading amplitude patterns in CWT images. The CWT images not only include time-domain features but also have frequency-domain features in each instance. Consequently, our experiment results demonstrate that training with the CWT dataset outperforms using the raw dataset.

Furthermore, we add random noise to the CWT input images during training to make our classifier more robust. Small-size datasets often struggle with overfitting problems as the learning process more easily fits the small given inputs [51]. For each CWT image, we add Gaussian white noise with a mean (μ) of 0.01 and variance (σ) of 0.01. This means that for each pixel, we add a noise value generated by the Gaussian distribution. Figure 13 illustrates the original sample (Figure 13a) and the augmented sample with 1% Gaussian noise (Figure 13b).

By adding 1% Gaussian noise to each instance, we doubly augment the size of the original CWT dataset. Table 2 summarizes the collected dataset of power traces. In the performance evaluation of the CNN classifier, the augmented images are only used in the training dataset but not in the testing set. The data augmentation helps to achieve better classification performance, as validated in Section 7. With this newly collected dataset, we aim to see whether our CNN classifier can identify the data exfiltration attack cases well, including unseen attack patterns.

6 CNN-BASED ATTACK DETECTION

In this section, we design a 2D-CNN classifier that takes the input of CWT images to detect data exfiltration behavior on IoT devices. Section 6.1 describes the classifier design, and Section 6.2 demonstrates the impact of design factors.

6.1 2D-CNN Design

LightAuditor uses a 10-layer CNN as the classifier. The CNN takes a CWT image and passes the image through multiple layers. The layers assign parameters including weights and biases to the input and feed the output to the next layer. These parameters are "learnable" in that during training the neural network adjusts the values in order to predict inference results better. During each training round, the neural network also calculates a cost function that represents how far the predicted values are away from the actual value. Thus, the neural network aims to minimize this cost function [2].

Figure 14 shows the overview of the proposed CNN design. The input layer takes data input of size $227 \times 227 \times 3$ and passes it to the convolutional layer. This input size corresponds to processed CWT images of 227×227 pixels with RGB value. Other existing image classifiers, such as AlexNet [33] and GoogLeNet [52], also receive input images of similar sizes. For other layers, we follow the

Table 3: Hyper-parameters used in our proposed 2D-CNN

Parameter	Optimizer	Epochs	L2 Factor	Learning Rate
Value	SGDM [49]	30	0.01	0.001

Table 2: The Collected Dataset of Power Traces

Inference Class	**Description**	**Number of Instances**		**Total number of instances**
		Original	**Gaussian Noised**	
Idle	When device is in idle	600	600	1200
Visible Light (Dimmable)	Normal light controlled by dimmable switch	600	600	1200
Visible Light (Instant Switch)	Normal light controlled by instant switch	600	600	1200
Infrared Light	Infrared light controlled by user app	600	600	1200
Visualizer	Random light changes corresponding to surrounding music	600	600	1200
Data Exfiltration Attack	Infrared light changes that exfiltrate private data	600	600	1200

[1] Each instance represents a CWT image of the size $227 \times 227 \times 3$.

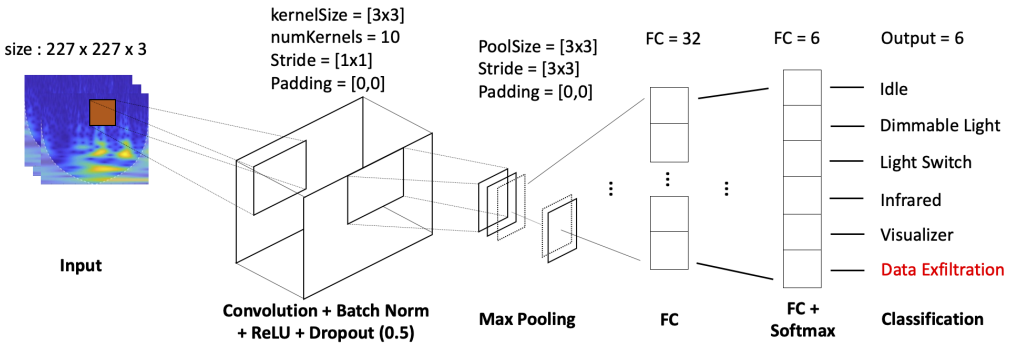


Figure 14: 2D-CNN Design for Power Consumption CWT Images

standard workflow of CNNs, tuning hyper-parameters. The impact of design choices is addressed in the next subsection. Besides, a dropout layer is used to provide regularization to the model and prevent potential overfitting. Lastly, the last layer outputs a label for each CWT input image. For all the experiments, we used the hyper-parameters presented in Table 3.

6.2 Impact of Design Factors

When we designed the CNN classifier, we considered several design factors to enhance classification performance while avoiding potential overfitting problems. Figure 15 illustrates the impact of the CNN design factors. We considered the average accuracy of five-fold cross-validation tests when choosing design factors. The details of the testing procedures are described in Section 7.

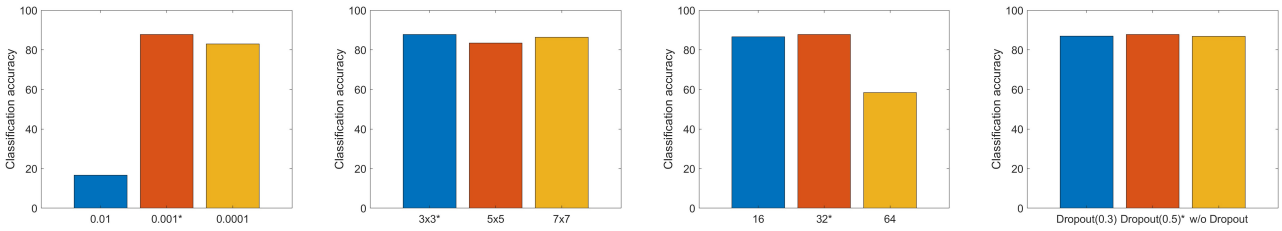
6.2.1 Learning Rate. First, we observe the impact of different learning rates in learning CWT images and kernels. The learning rate controls how much the neural network changes its parameters for each training round in order to minimize the cost function. Figure 15a shows the classification accuracy with different learning rates. High learning rates, such as 0.01, can expedite the training process but may overshoot for a minimum. In our training with a 0.01 learning rate, the model fails to converge, which results in a random classifier (Accuracy of 16% : 1/6). On the other hand, low learning rates like 0.0001 lead to slow training and do not find a minimum either in that the model takes a very few steps in each training round. Instead, in the CWT image training with the common input sizes ($227 \times 227 \times 3$), the learning rate of 0.001 promises the best results in all of our experiments.

6.2.2 Kernel Size. In our model, changing kernel sizes have limited impact on the performance. 3×3 and 7×7 are both minuscule inside the 227×227 pixel image. All three kernels observe the edge of drastic power amplitude changes in CWT images. We choose 3×3 since it gives the best accuracy.

This result is supported by Figure 10 that power patterns are local. Deeper neural networks for image classification like AlexNet usually learn different edges or color patterns on various types of images at each convolutional layer and then stack the information together. However, in CWT images, the CWT has already stacked information, so studying local patterns is good enough with shallower layers and fewer parameters than other complicated models. The experiment results in Section 7 demonstrate that our lightweight design is still good in the CWT image classification.

6.2.3 Fully Connected Layer. Figure 15c illustrates the CNN's accuracy with the different number of neurons in the fully connected layer. In general, as the number of neurons in a fully connected layer increases, there will be more feature sets to predict input instances. However, too many neurons give too many parameters to train, and small-sized datasets cannot be trained well. Our experiment shows that a fully connected layer with 32 neurons gives the best performance. When testing with 64 neurons, the CNN sometimes fails to converge, resulting in a low average cross-validation accuracy.

6.2.4 Dropout Layer. Figure 15d illustrates that regardless of the dropout layer, all experiments achieved an accuracy higher than 85%. However, we decided to add the dropout layer since we aim to build a CNN that can identify unsee data exfiltration attacks. Experiments in Section 7 show that the model with the dropout layer performs well in identifying unseen data patterns.



(a) Learning rate **(b) Convolutional Kernel size** **(c) Number of neurons in FC layer** **(d) Dropout layer rate**

Figure 15: CNN Design Factors – The asterisk represents chosen design factors

7 PERFORMANCE EVALUATION

In this section, we evaluate the performance of the system components. Section 7.1 introduces evaluation settings. Section 7.2 then demonstrates the classification performance of our CNN design compared to other baseline models. Section 7.3 shows the detection performance against unseen attack patterns. Finally, Section 7.4 evaluates system performance metrics, such as CNN inference time and bandwidth.

7.1 Evaluation Settings

We aim to validate our 2D-CNN classifier design for power CWT image analysis. To the best of our knowledge, we are the first to propose a CNN design for the CWT images derived from power consumption data. The literature has not examined CWT images for power consumption data analysis. Our results suggest that the CWT images have more information in both time- and frequency-domains, and our 2D-CNN design is suitable for analyzing the CWT power consumption data.

7.1.1 Transfer Learning Classifiers. We first consider a transfer learning method, which allows using pre-trained weights and biases for solving other classification problems. This approach is especially beneficial when the source dataset does not have enough instances to train [50]. With the pre-trained models, users can easily test their datasets and also avoid the problem of not having enough training sets. Thus, we choose two well-known image classifiers: AlexNet [33] and GoogLeNet [52] for performance comparisons. To transfer the existing classifiers to our problem, we replaced the last layers of those classifiers, and the final fully-connected layer is set to the number of our classes. Then, we retrained the inherited parameters of all layers in the modified network on our dataset. By doing this, we do not need to train the parameters from scratch.

Table 4 describes the classifier features of our model and two other transfer learning classifiers. Since the existing CNN models are often deep in layers and parameters, however, sometimes users may encounter overfitting problems or slow learning. Thus, a vast number of parameters and learning capacity are not always preferred.

On the other hand, our proposed classifier does not have an overwhelming number of parameters. In short, LightAuditor is similarly accurate as AlexNet and GoogLeNet while being 34 times smaller and 51 times less compute intensive. This also hints at appropriate design choices in IoT environments. For example, in transfer learning classifiers, features of edges in cat or dog images were also trained. This kind of feature may not be necessary for our application scenario. As illustrated in Figure 12, our potential features need to represent durations of amplitude changes or frequency ranges of power changes from the CWT images. Thus, the features that need to be extracted are relatively limited. This observation confirms that CNN classifiers for CWT images may not need to be as complicated as existing image classifiers. Section 7.2 validates our assumptions (hypotheses) on the CNN design.

Table 4: LightAuditor comparison to Transfer Learning CNNs

Classifier	Accuracy	Million Mult-Adds	Million Parameters
$GoogLeNet^T$	92.05	1550	62.3
$AlexNet^T$	93.00	930	61
LightAuditor	87.84	18	1.8

[1] Superscription T represents a transfer learning classifier.

7.1.2 1D-CNN baseline model. Another model we compared is a 1D-CNN [30] that takes into account raw power consumption data as its input. In order to match the same size as our power trace dataset, we adjusted the input and output layers of the 1D-CNN. Our input instance is 4 seconds with 220 data points under the sample rate 55Hz, and the output layer has six labels.

For other layers, we kept the original design principles of the 1D-CNN model. The kernel size is 1×16 and the stride size is 1×8 with 10 kernels, accordingly. Since this classifier receives raw power consumption data like in Figure 10, it only predicts results from time-domain features.

7.1.3 Implementation of the CNNs. In order to compare the classifier design, we used a department server that has an Nvidia RTX 3090 GPU with 24GB of memory. On this server, we first implemented and tested our proposed 2D-CNN classifier (Section 6) in Matlab. For the performance comparison, we transferred the two well-known image classifiers: AlexNet [33] and GoogLeNet [52], adjusted the input and the last layers to fit our classification problem, and retrained them. Regarding the input size, AlexNet has the same as our CWT images ($227 \times 227 \times 3$). GoogLeNet requires a slightly different input size ($224 \times 224 \times 3$), so we rescaled the CWT images to fit the input layer of GoogLeNet. Lastly, we downloaded an open-source of the 1D-CNN classifier [47] and adjusted its input and output layers in Matlab, as described in Section 7.1.2.

7.2 Performance in Classifying Bulb Behavior

Recall that the CNNs classify a CWT image into six classes: Idle, Dimmable Visible Light, Instant Switch Visible Light, Infrared Light, Visualizer, and Data Exfiltration. The performance of the CNNs is measured in accuracy, precision, recall, and F1-score. Accuracy measures how many data instances are correctly classified out of all instances. Precision measures out of all data instances that are classified into one class, how many data instances actually belong to the class. Recall measures out of all data instances that belong to a class, how many are classified correctly. F1-score then gives a harmonic mean of precision and recall with Equation 4. We calculate each model's accuracy, precision, recall, and F1-Score by averaging on scores of all six labels. From our means of calculating an average, recall is the same as accuracy, and is thus not displayed in the tables.

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

7.2.1 Cross-validation Tests. To measure the performance of each model, we ran five-fold cross-validation tests. In the cross-validation tests, we first divided the dataset into five subsets. Four of them are used together as the training dataset, and another one is held as the testing dataset. Then, we ran the data augmentation specified in Section 5.3 on the training datasets and trained the model with the augmented training sets. Next, we tested the trained model with the remaining test set, which is not augmented. We conducted the same procedure five times with different combinations, each time recording the accuracy, precision, and F1-score. For each metric, we took the average of the five runs.

Table 5: Classification Performance in Cross-validation Tests

Classifier	Accuracy (%)	Precision (%)	F1-Score
$GoogLeNet^T$	92.05	92.04	91.93
$AlexNet^T$	93.00	92.82	92.92
LightAuditor	87.84	87.02	87.57

Table 6: Classification Performance with the Original Dataset

Classifier	Accuracy (%)	Precision (%)	F1-Score
$GoogLeNet^T$	88.33	88.45	88.30
$AlexNet^T$	93.33	93.43	93.33
LightAuditor	78.05	78.44	77.96

Table 5 shows the results of cross-validation tests on three CNNs. The LightAuditor CNN achieves an average of 87.8% accuracy. Besides the classification accuracy, precision and F1-score values are also displayed. Two transferred classifiers perform slightly better than our model in all three metrics by about 5%. Such performance scores are reasonable since they are pre-trained and sophisticated models for image recognition. However, considering the number of parameters shown in Table 4, these performance differences are not significant. Overall, since the classification results of AlexNet and GoogLeNet are similar in the cross-validation tests, we will compare our CNN classifier only with AlexNet in the unseen-pattern experiments (Section 7.3).

7.2.2 Impact of Data Pre-processing. We ran another cross-validation tests with the original dataset as a training set and obtained the results in Table 6. Tables 5 and 6 demonstrate the effect of data augmentation on classification performance. As illustrated in Section 5.3, our data augmentation doubles the size of the original dataset in training. Together with the added noise, the data augmentation improves our CNN model's validation accuracy and makes the model more robust.

Interestingly, we observe that the improvement on CNN performance is the largest in our 2D-CNN. The accuracy difference is almost 10%, while the transfer learning models have an accuracy difference of less than 5%. An explanation is that the transfer learning models have already many parameters pre-trained while our CNN trains from scratch. Thus, our model benefits more significantly from the data augmentation.

Furthermore, we measure the performance metrics of the 1D-CNN model [30] with our raw power consumption dataset. The average accuracy of the classification is 80.9%, which is about 7% lower than the results from our 2D-CNN model. It is explainable that the 1D-CNN does not take into account frequency-domain features from the power consumption data, while our 2D-CNN and the other transfer learning models utilize the CWT images as input.

7.3 Performance against Unseen Patterns

In the previous subsection, we tested and classified within the collected dataset; the power consumption data used for training was collected when the smart bulb was transmitting an image. However, there are various scenarios in which adversaries can change their behavior. For example, data exfiltration attacks may leak any format of information including images, texts, and audio. In addition, the power traces when the smart bulb leaks private data under different bitrates or encoding schemes would be different in that the amplitude changes in the smart bulb's infrared signal may differ.

In order to cover those potential cases, we also evaluate the classifier with attack data unseen during training. While it is nearly impossible to consider all possible anomalies, the adversaries must leak data through the covert channel in this type of attack. As such, we choose several variations of potential unseen attack patterns that exploit the infrared channel.

7.3.1 Unseen Data File. To validate whether our classifier is versatile to identify data leakage of other data files, we used different image files unseen during training. We first recorded power consumption data when other image files were being transmitted. Then, we replaced the original testing instances with the newly-collected power consumption instances. By doing so, we can see whether our classifier is robust against unseen data leakage behavior.

Table 7 shows the performance of two CNNs in the tests. AlexNet gives a high precision of 95%. This is reasonable since AlexNet scores highest in the cross-validation tests. Nevertheless, all models still have an accuracy greater than 90%, which is similar to the cross-validation tests. In addition, our model's classification results are as good as the results from the cross-validation tests. Altogether, these evaluation results show that power consumption data on infrared emission can be used to effectively identify attacks when different files are leaked.

7.3.2 Unseen File Type. In addition to new image files, attacks leaking different file types may also generate unseen power patterns. For example, a text or audio file that includes users' private data can also be a target of adversaries. In this subsection, we also tested with power consumption data when the adversary program leaks text and audio files. That way, we can see if the classifier is able to detect malicious behavior when exfiltrating different file types.

Table 8 illustrates the classification performance when the unseen file types were exfiltrated. Still, the transfer learning classifiers' sophisticated structure gives them higher performance against unseen file types. Our CNN's performance against unseen file types also provides consistent classification results.

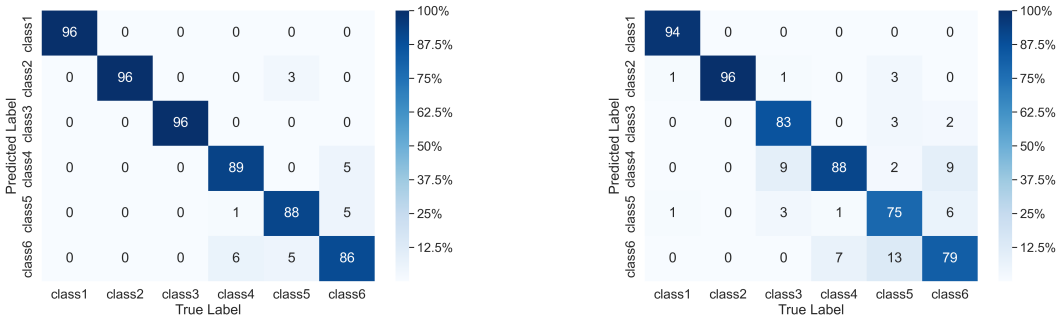
Figure 16 shows the confusion matrices of LightAuditor and AlexNet when different file type was the attack target. Our CNN identifies 79 out of the 96 instances, while AlexNet performs better than us. Yet, considering fewer computing resources, this result reflects a comparable performance in detecting unseen anomalies.

Table 7: Classification Performance against Unseen Images

Classifier	Accuracy(%)	Precision(%)	F1-Score
$AlexNet^T$	95.65	95.65	95.65
LightAuditor	90.45	90.32	90.33

Table 8: Classification Performance against Unseen Filetypes

Classifier	Accuracy(%)	Precision(%)	F1-Score
$AlexNet^T$	92.78	92.59	92.71
LightAuditor	88.54	88.43	88.47



(a) $AlexNet^T$ (b) $LightAuditor$

Figure 16: Confusion matrices of Classification Performance against Unseen File-types

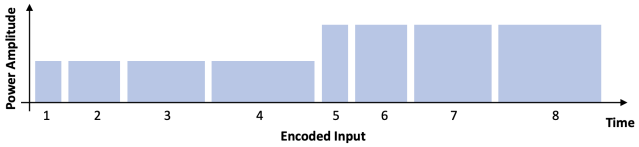


Figure 17: Data Sample using a New Encoding Scheme

7.3.3 New Encoding Scheme. Furthermore, adversaries could use a new encoding scheme such that the bulb mimics one of the legitimate patterns. So, the power consumption data become unseen to the classifier. By doing this, attackers may avoid detection. In the original attack [41], the authors used ASK-encoded data to leak image files, as shown in Figure 1. Besides the ASK encoding, we choose to utilize pulse-duration modulation [54] in which each bit is represented by a signal of a different duration. In other words, we propose an encoding scheme in which four adjacent bits are now represented by a signal of the same amplitude but a different duration based on the modulo value of four.

Figure 17 illustrates a sample signal generated by the new encoding scheme. In this way, the bulb's amplitude fluctuates less, and thus the power consumption pattern differs from that of the original attack. We then tested the performance of the classifiers against the generated unseen data. Table 9 illustrates the classification results. Despite the slight drop in the classification performance, the LightAuditor CNN is still able to identify unseen anomalies with an accuracy of about 87% under the newly designed encoding scheme.

Table 9: Classification Results against Unseen Encoding Scheme Data

Classifier	Accuracy(%)	Precision(%)	F1-Score
Original Encoding	87.84	87.02	87.57
New Encoding	86.98	86.54	86.57

Table 10: Classification Results against Unseen Bitrate Data

Classifier	Accuracy(%)	Precision(%)	F1-Score
Original Bitrate	87.84	87.02	87.57
80% Bitrate	86.53	86.98	86.57
120% Bitrate	85.59	85.35	85.37

7.3.4 Different Bitrate. Finally, we tested the impact of different bitrates at which private data is being leaked through the infrared channel. In this attack, bitrate represents the number of infrared signals emitted in a certain time unit. In the original attack [41], the smart bulb emits infrared signals every half second. We recorded the power consumption when the bulb emitted infrared at the fixed bitrate and trained the model with this data. However, if the attacker program transmits the data at a different rate, the power consumption pattern might be different. This also leads to different patterns in CWT input images in both time and frequency domains. As such, we generated new testing sets under two different bitrates: one is 80% of the original bitrate, and another is 120% of the original bitrate. This test can provide how tolerant each model is against new data patterns as well.

Table 10 shows the classification results against unseen data under different bitrates. While AlexNet still achieves better results at about 93%, our proposed CNN also classifies new anomalies with an accuracy of more than 85%. Moreover, the results are similar to the cross-validation tests. These similar results suggest that variances of unseen data under this attack are limited due to the infrared emission. Under the different bitrates or encoding schemes, the adversaries are still forced to exfiltrate data through infrared emission. Thus, regardless of unseen data patterns, the classifiers with CWT images consistently provide similar classification performance. This observation will benefit future studies that aim to defend against covert channel attacks.

7.4 System Performance Evaluation

In this subsection, we also measure system-level performance metrics. Processing or inference time matters when it comes to online inference scenarios. We used a Linux server with 32 cores and 64 GB RAM. It is a common setting for an edge server in IoT environments.

Even in IoT environments, network bandwidth should be minimized so that online inference can be done in real time. Each CWT image is a 227×227 pixel RGB image. It is then approximately 12 kB per instance. Thus, the network bandwidth should not be an issue since most networks provide at least 10 MB s^{-1} bandwidth nowadays. Under the bandwidth of 10 MB s^{-1}, network transmission time would be less than 0.0001 s. In addition, the processing time for each instance is negligible, as shown in Table 11.

Table 11: Processing Time of Data Processing Jobs

Pre-processing Job	Processing Time per Image (s)
Root Mean Square	0.000004
Alpha Filter	0.000007
CWT Transform	0.00406

Table 12: Classification Time with 600 CWT Images

Classifier	Training (s)	Testing (s)	Testing per Image (s)
$GoogLeNet^T$	5915.5	3.23	0.027
$AlexNet^T$	1229.3	2.21	0.018
LightAuditor	258.9	0.31	0.0026

What makes LightAuditor a comparable option to transfer learning CNNs is its lightweightness. As shown in Table 4, the huge number of parameters in AlexNet and GoogLeNet makes them more power- and time-consuming options. Meanwhile, the classification performance of LightAuditor is similarly accurate as other CNNs. Moreover, Table 12 shows the training times of the CNNs on our testing server. Our CNN trains 5 times faster than AlexNet and 23 times faster than GoogLeNet. The testing times give an estimate of CNN inference time. On average, our CNN classifies a CWT input image in 0.0026 seconds. This is 7 times faster than AlexNet and 10 times faster than GoogLeNet.

Overall, the total system time (processing + networking + classification) per instance in LightAuditor is less than 0.01 seconds. This estimated measurement demonstrates that the Light Auditor system is a more lightweight option and shows the potential that it can be deployed on edge devices for online classification. More discussion will be addressed in Section 9.

8 RELATED WORKS

In Section 8.1, we summarize other IoT covert-channel attacks besides the cover-light smart bulb attack. In Section 8.2, we present power-auditing sensing research for other purposes in IoT environments.

8.1 Covert-channel Attacks

The common idea of covert-channel attacks is that adversaries want to exfiltrate data without using the existing Internet network. In addition to what we have tested, attackers can exploit other devices that support covert-channel capabilities, such as acoustic and optical channels. These attacks are not easy to defend against due to the lack of proper detection channels. Thus, we recap existing related work, including some IoT data exfiltration attacks.

8.1.1 Electromagnetic. An electric current in a wire generates electromagnetic fields. A few studies have used electromagnetic emissions to exfiltrate private data. The authors in [16] controlled electromagnetic radiation to the frequency modulation (FM) radio band in order to leak data to nearby smartphones. Another study in [15] also exploited electromagnetic radiation on the CPU-memory bus of GSM phones. This project used different frequency bands to leak data, while the USBee research [17] instead utilized the USB data bus for encoding and transmitting data via RF signals.

8.1.2 Magnetic. Magnetic covert channels have also been used to leak private data based on CPU utilization. For example, the authors in ODINI [25] utilized a low-frequency magnetic field generated by CPUs. This low-frequency magnetic radiation can even propagate through metal-shielded walls. Thus, text messages successfully were exfiltrated from a faraday room. Likewise, the Magneto research [13] also utilized a CPU-generated magnetic field to build a covert channel between PCs and smartphones. Recently, Zhang et al. [58] have reduced the risk of being detected in this magnetic covert-channel attack by embedding private data in other data such as video.

8.1.3 Electric. The authors in [22] have proposed a stealthy channel by regulating CPU utilization. In this attack, a malicious program ran on PCs to intentionally generate power consumption patterns. By doing this, data can be encoded and propagated through power lines. However, the assumption is that an attacker should be equipped with power lines to connect a device.

8.1.4 Acoustic. Human beings' hearing range is limited from $20Hz$ to $18kHz$. That said, any acoustic signals beyond this range cannot be heard. Meanwhile, ultrasonics can generate acoustic signals above the human hearing span. Thus, a few studies have utilized ultrasonic transmitters. In MOSQUITO [21], the authors proposed a speaker-to-speaker communication that exploits an audio chip feature. In DeafAid [10], ultrasonic-enabled speakers are also used to transmit encoded private data without users' notice. Then, this signal can be captured by gyroscopes because gyroscope sensors react to ultrasonic sound. Besides ultrasonics, some studies have utilized a hearable sound, such as CPU cooling fans [19] and hard-drive motors [20]. In the future, it is likely for mobile devices like smartphones to include these kinds of ultrasonic capabilities.

8.1.5 Optical. Another covert channel can be made by optical signals. The main idea of this channel is that many computing devices are equipped with LED indicators, and the LEDs are controllable from software levels. Thus, adversaries can easily encode data on different patterns of LED blinks. These encoded blinks are also interceptable by drones or other local cameras. For example, LED-it-GO [24] used a drone outside of a building to capture exfiltrated blink signals from infected PCs. Likewise, the xLED [23] research exfiltrates data via seven LEDs of a router. Another work in [14] also exploited infrared (IR) signals and security cameras to exfiltrate private data. Due to the nature of the multi-bit optical signals, its bandwidth is higher than other covert channels.

8.1.6 Thermal. Some studies even utilized emitting heat from PC's components [18]. For example, using a CPU, GPU, or HDD can generate heat, and an adjacent computer can detect temperature changes by utilizing built-in thermal sensors. Through this covert channel, data can also be encoded and exfiltrated.

Even though some studies may not be practical, we clearly see that various covert channels can be built between IoT devices to exfiltrate private data. Other than the above cases, there are more covert-channel attacks that need to be watched. Thus, proper monitoring and detecting solutions are still in need. In regard to this matter, we believe that power side-channel monitoring can play a crucial role in detecting IoT data exfiltration attacks. Moreover, our proposed CNN model demonstrates the feasibility of using CWT images of power consumption data.

8.2 Power-auditing IoT Sensing

Weak security on IoT devices makes them soft targets for adversaries, and often users may not even be aware of whether their devices are infected [9]. In addition, IoT devices are not controlled by a few standard operating systems or protocols. Hence, a new research direction is needed to find a universal security solution for various IoT devices in practical deployments. Utilizing power consumption data can be one solution since it is universal and does not require modifications to the existing devices. In this vein, power consumption data have been used for side-channel detection in the last decades.

For example, several studies have utilized power-auditing techniques for botnet detection in IoT environments. Myridakis et al. [43] proposed a power monitoring circuit for IoT botnet detection. Li et al. [37] introduced an energy auditing method to infer DoS attacks, using energy meters [26]. Both studies aim to detect massive DoS attacks with spike detection approaches. On the other hand, Jung et al. [29] proposed a 1D-CNN to identify IoT botnet intrusion cases. All these works have utilized raw power consumption data for behavior detection. Instead, we proposed to use converted CWT images from power consumption data. As demonstrated in Section 7, CWT images can tell more information than using raw power traces as they contain time- and frequency-domain features. Furthermore, although power-auditing-based solutions have the potential of detecting malicious botnet attacks, no studies to date have examined covert-channel attacks on power consumption data in IoT environments. To the best of our knowledge, we are the first to propose a learning model to defend against IoT data exfiltration attacks using CWT power images.

9 DISCUSSION AND FUTURE WORK

9.1 Covert-channel Monitoring Methods

Carrara et al. [6] state that all sensing channels should be audited to detect possible malign communication. In order to meet this requirement, there can be various ways of monitoring covert channels. For example, since the smart-bulb attack exploits infrared emission to leak users' private data, using an infrared receiver can be an effective solution for this specific attack. However, it requires an extra set-up per device for normal users. Moreover, if there are other covert channels in place, such as ultrasonic speakers, another corresponding covert-channel receiver needs to be considered, which may not be preferable. Therefore, considering the increasing growth of IoT devices, a universal monitoring method is needed. Our solution uses power consumption data to monitor covert channels universally, which is more efficient and feasible in user practice.

9.2 Other Covert-channel Attacks

Although we tested different types of unseen attacks, our limitations still exist since we only considered infrared emission on a single bulb as a covert channel. This is in part because, at the time of writing, there is no other infrared-enabled smart bulb to test on the market. However, if attacks are involved in other infrared-enabled devices, the infrared emission would likely generate detectable patterns, as shown in our unseen-pattern experiments (Section 7.3). Still, we plan to enhance the classification results against unseen data as accurately as more complicated models.

Furthermore, as reviewed in Section 8.1, adversaries have also exploited other covert channels for this type of attack in the IoT environment. As such, we plan to extend the capability of our system to other covert channels based on the proposed design. We need to examine whether a channel-independent model is possible or a channel-specific model is necessary. In any case, the power modeling approach is promising because measuring power consumption is universal to any covert channels. Therefore, detecting information leakage through other channels, such as ultrasonic channel [10], remains our future work.

9.3 Edge Device Deployment

Another enhancement we plan to do is to deploy our detection system into edge devices. For proof-of-concept, we tested with the collected dataset on an offline server. Although we did not conduct an online study, we provide several system metrics in Section 7.4 to demonstrate the potential online system. Those values suggest that the proposed system does not require many resources, and thus it can be executed on an edge device with limited computing resources.

Furthermore, we plan to deploy our system in a real-world setting where over-the-shelf IoT devices are involved in attacks for real-time detection. Doing this will validate the system performance results presented in our experimental environments.

10 CONCLUSION

This paper examines the data-exfiltration attack via the infrared channel and proposes a power-auditing-based solution for anomaly detection. We first define an attack model that consists of a brute-force intrusion and a covert-channel attack. Then, in our testbed implementation, we design a power-auditing system to identify the malicious behavior of exfiltrating information through the smart bulbs. Our system design includes pre-processing procedures on the power consumption data and a 2D-CNN classifier that receives the CWT images transformed from power traces as input. The experiments demonstrate that the proposed classifier is lightweight and a comparable option to existing CNNs for power-consumption data classification.

In conclusion, this case study shows that the power auditing approach can detect data-exfiltration attacks through IoT covert channels, including unseen patterns.

ACKNOWLEDGMENTS

The authors would like to thank all the anonymous reviewers for their valuable comments and helpful suggestions. This research was supported by both Northern Virginia Commonwealth Cyber Initiative (Award Number N-4Q22-006) and Coastal Virginia Cybersecurity Innovation (Cybersecurity Dissertation Fellowship).

REFERENCES

[1] Ahmed Al-Haiqi, Mahamod Ismail, and Rosdiadee Nordin. 2014. A new sensors-based covert channel on android. *The Scientific World Journal* 2014 (2014).

[2] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*. 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the mirai botnet. In *26th USENIX security symposium (USENIX Security 17)*. 1093–1110.

[4] Arduino. 2022. ACS 712 Current Sensor. https://create.arduino.cc/projecthub/instrumentation-system/acs712-current-sensor-87b4a6.

[5] Fatima Sajid Butt, Luigi La Blunda, Matthias F Wagner, Jörg Schäfer, Inmaculada Medina-Bulo, and David Gómez-Ullate. 2021. Fall detection from electrocardiogram (ecg) signals and classification by deep transfer learning. *Information* 12, 2 (2021), 63.

[6] Brent Carrara and Carlisle Adams. 2016. Out-of-band covert channels—A survey. *ACM Computing Surveys (CSUR)* 49, 2 (2016), 1–36.

[7] Luca Caviglione, Alessio Merlo, and Mauro Migliardi. 2018. Covert channels in IoT deployments through data hiding techniques. In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 559–563.

[8] Patrick Cronin, Charles Gouert, Dimitris Mouris, Nektarios Georgios Tsoutsos, and Chengmo Yang. 2019. Covert data exfiltration using light and power channels. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*. IEEE, 301–304.

[9] N. Panwar et al. 2019. Smart Home Survey on Security and Privacy. arXiv.org [Online; accessed 2021].

[10] Ming Gao, Feng Lin, Weiye Xu, Muertikepu Nuermaimaiti, Jinsong Han, Wenyao Xu, and Kui Ren. 2020. Deaf-aid: mobile IoT communication exploiting stealthy speaker-to-gyroscope channel. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–13.

[11] Annarita Giani, Vincent H Berk, and George V Cybenko. 2006. Data exfiltration and covert channels. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, Vol. 6201. SPIE, 5–15.

[12] Maria Guerra. 2017. The Power of IoT Devices. https://www.electronicdesign.com/power-management/article/21805184/the-power-of-iot-devices.

[13] Mordechai Guri. 2021. Magneto: Covert channel between air-gapped systems and nearby smartphones via cpu-generated magnetic fields. *Future Generation Computer Systems* 115 (2021), 115–125.

[14] Mordechai Guri and Dima Bykhovsky. 2019. air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (ir). *Computers & Security* 82 (2019), 15–29.

[15] Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval Elovici. 2015. {GSMem}: Data Exfiltration from {Air-Gapped} Computers over {GSM} Frequencies. In *24th USENIX Security Symposium (USENIX Security 15)*. 849–864.

[16] Mordechai Guri, Gabi Kedma, Assaf Kachlon, and Yuval Elovici. 2014. AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. In *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*. IEEE, 58–67.

[17] Mordechai Guri, Matan Monitz, and Yuval Elovici. 2016. USBee: Air-gap covert-channel via electromagnetic emission from USB. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 264–268.

[18] Mordechai Guri, Matan Monitz, Yisroel Mirski, and Yuval Elovici. 2015. Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In *2015 IEEE 28th Computer Security Foundations Symposium*. IEEE, 276–289.

[19] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, and Yuval Elovici. 2016. Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers. *arXiv preprint arXiv:1606.05915* (2016).

[20] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, and Yuval Elovici. 2017. Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise ('DiskFiltration'). In *European symposium on research in computer security*. Springer, 98–115.

[21] Mordechai Guri, Yosef Solewicz, and Yuval Elovici. 2018. Mosquito: Covert ultrasonic transmissions between two air-gapped computers using speaker-to-speaker communication. In *2018 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 1–8.

[22] Mordechai Guri, Boris Zadov, Dima Bykhovsky, and Yuval Elovici. 2019. PowerHammer: Exfiltrating data from air-gapped computers through power lines. *IEEE Transactions on Information Forensics and Security* 15 (2019), 1879–1890.

[23] Mordechai Guri, Boris Zadov, Andrey Daidakulov, and Yuval Elovici. 2017. xled: Covert data exfiltration from air-gapped networks via router leds. *arXiv preprint arXiv:1706.01140* (2017).

[24] Mordechai Guri, Boris Zadov, and Yuval Elovici. 2017. LED-it-GO: Leaking (a lot of) Data from Air-Gapped Computers via the (small) Hard Drive LED. In *International conference on detection of intrusions and malware, and vulnerability assessment*. Springer, 161–184.

[25] Mordechai Guri, Boris Zadov, and Yuval Elovici. 2019. Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields. *IEEE Transactions on Information Forensics and Security* 15 (2019), 1190–1203.

[26] A Hindle, A Wilson, K Rasmussen, E. Jed Barlow, Joshua Charles Campbell, and Stephen Romansky. 2014. Greenminer: A hardware based mining software repositories software energy consumption framework. *dl.acm.org* (2014).

[27] Mordor Intelligence. [n.d.]. Smart Plug Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). https://www.mordorintelligence.com/industry-reports/smart-plug-market.

[28] Vikramaditya R Jakkula, Aaron S Crandall, and Diane J Cook. 2009. Enhancing anomaly detection using temporal pattern discovery. In *Advanced intelligent environments*. Springer, 175–194.

[29] Woosub Jung, Yizhou Feng, Sabbir Ahmed Khan, Chunsheng Xin, Danella Zhao, and Gang Zhou. 2021. DeepAuditor: Distributed Online Intrusion Detection System for IoT devices via Power Side-channel Auditing. *arXiv preprint arXiv:2106.12753* (2021).

[30] Woosub Jung, Hongyang Zhao, Minglong Sun, and Gang Zhou. 2019. IoT Botnet Detection via Power Consumption Modeling. In *ACM/IEEE CHASE*.

[31] Gerasimos Kalouris, Evangelia I Zacharaki, and Vasileios Megalooikonomou. 2019. Improving CNN-based activity recognition by data augmentation and transfer learning. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Vol. 1. IEEE, 1387–1394.

[32] Noam Kovartovsky. 2019. Brute Force Attacks on IoT – Here to Stay? https://www.allot.com/blog/brute-force-attacks-iot/.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[34] Mikhail Kuzin, Yaroslav Shmelev, and Vladimir Kuskov. 2018. New trends in the world of IoT threats. https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/.

[35] Butler W Lampson. 1973. A note on the confinement problem. *Commun. ACM* 16, 10 (1973), 613–615.

[36] Shih-Hsiung Lee and Chu-Sing Yang. 2017. An intelligent power monitoring and analysis system for distributed smart plugs sensor networks. *International Journal of Distributed Sensor Networks* 13, 7 (2017), 1550147717718462.

[37] F Li, Y Shi, A Shinde, and J Ye. 2019. Enhanced cyber-physical security in internet of things through energy auditing. *ieeexplore.ieee.org* (2019).

[38] Xudong Li, Jianhua Zheng, Mingtao Li, Wenzhen Ma, and Yang Hu. 2021. Frequency-Domain Fusing Convolutional Neural Network: A Unified Architecture Improving Effect of Domain Adaptation for Fault Diagnosis. *Sensors* 21, 2 (2021), 450.

[39] LIFX. 2020. HTTP API for Developers. https://api.developer.lifx.com.

[40] LIFX. 2020. Nightvision BR30. https://www.lifx.com/products/lifx-nightvision-br30.

[41] Anindya Maiti and Murtuza Jadliwala. 2019. Light ears: Information leakage via smart lights. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–27.

[42] Viorel Miron-Alexe. 2016. Comparative study regarding measurements of different AC current sensors. In *2016 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*. IEEE, 1–6.

[43] Dimitrios Myridakis, Paul Myridakis, and Athanasios Kakarountas. 2021. A Power Dissipation Monitoring Circuit for Intrusion Detection and Botnet Prevention on IoT Devices. *Computation* (2021). https://dblp.org/rec/journals/computation/MyridakisMK21

[44] Sidra Naseem, Kashif Javed, Muhammad Jawad Khan, Saddaf Rubab, Muhammad Attique Khan, and Yunyoung Nam. 2021. Integrated CWT-CNN for epilepsy detection using multiclass EEG dataset. (2021).

[45] Patrick Nelson. 2018. Using the Internet of Sound to transfer IoT data via speakers. https://www.networkworld.com/article/3324246/using-the-internet-of-sound-to-transfer-iot-data-via-speakers.html.

[46] Raspberry Pi. [n.d.]. Raspberry Pi 3 Model B. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/.

[47] Raspberry Pi. [n.d.]. Raspberry Pi 3 Model B. https://woossup.github.io/IoT-Botnet-Detection/.

[48] Kok Hang Poh, Chien Ye Tan, Nor Syafiqah Mat Ruslan, and Wan Amir Fuad Wajdi Othman. 2019. Design and implementation of simple IoT-based smart home system using arduino. *Technical Journal of Electrical Electronic Engineering and Technology* 3, 1 (2019), 1–13.

[49] Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks* 12, 1 (1999), 145–151.

[50] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging* 35, 5 (2016), 1285–1298.

[51] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[52] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer*

vision and pattern recognition. 1–9.
[53] Wikipedia. 2004. Continuous Wavelet Transform. https://en.wikipedia.org/wiki/Continuous_wavelet_transform.
[54] Wikipedia. 2006. Pulse-width modulation. https://en.wikipedia.org/wiki/Pulse-width_modulation.
[55] Wikipedia. 2018. Brute-force Attack. https://en.wikipedia.org/wiki/Brute-force_attack.
[56] Wikipedia. 2021. Smart Plug. https://en.wikipedia.org/wiki/Smart_plug.
[57] Kieran Woodward, Eiman Kanjo, and Athanasios Tsanas. 2020. Combining Deep Transfer Learning with Signal-image Encoding for Multi-Modal Mental Wellbeing Classification. *arXiv preprint arXiv:2012.03711* (2020).
[58] Juchuan Zhang, Xiaoyu Ji, Wenyuan Xu, Yi-Chao Chen, Yuting Tang, and Gang Qu. 2020. MagView: A distributed magnetic covert channel via video encoding and decoding. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 357–366.