

5-2023

Power Profiling Smart Home Devices

Kailai Cui
William & Mary

Follow this and additional works at: <https://scholarworks.wm.edu/honorsthesis>



Part of the [Digital Communications and Networking Commons](#), and the [OS and Networks Commons](#)

Recommended Citation

Cui, Kailai, "Power Profiling Smart Home Devices" (2023). *Undergraduate Honors Theses*. William & Mary. Paper 2024.

<https://scholarworks.wm.edu/honorsthesis/2024>

This Honors Thesis -- Open Access is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Undergraduate Honors Theses by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.

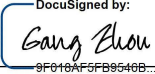
Power Profiling Smart Home Devices

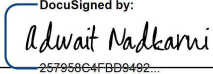
A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science in Computer Science from
William & Mary

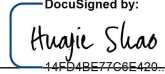
by

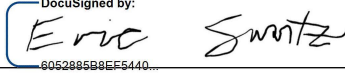
Kailai Cui

Accepted for 
DocuSigned by:
Gang Zhou
9F018AF5FB9546B...
(Honors, High Honors, Highest Honors)


DocuSigned by:
Gang Zhou
9F018AF5FB9546B...
Professor Gang Zhou, Committee Chair
Department of Computer Science


DocuSigned by:
Adwait Nadkarni
257950C4FDD9492...
Professor Adwait Nadkarni
Department of Computer Science


DocuSigned by:
Huajie Shao
14FD4BE77C6E420...
Professor Huajie Shao
Department of Computer Science


DocuSigned by:
Eric Swartz
605288588E5440...
Professor Eric Swartz
Department of Mathematics

Williamsburg, VA
May 5, 2023

Power Profiling Smart Home Devices

Undergraduate Honor Thesis by

Kailai Cui

Department of Computer Science

The College of William and Mary

May 2023

ABSTRACT

Power Profiling Smart Home Devices

Kailai Cui

In recent years, the growing market for smart home devices has raised concerns about user privacy and security. Previous works have utilized power auditing measures to infer activity of IoT devices to mitigate security and privacy threats.

In this thesis, we explore the potential of extracting information from the power consumption traces of smart home devices. We present a framework that collects smart home devices' power traces with current sensors and preprocesses them for effective inference. We collect an extensive dataset of duration $> 2\text{h}$ from 6 devices including smart speakers, smart camera and smart display. We perform different classification tasks including device identification and action classification and present accuracy and confusion matrices for each tasks. Our analysis reveals that from devices' running power traces, we can accurately identify the type of smart device being used with 93% accuracy and subsequently infer user behavior with on average 92% accuracy.

ACKNOWLEDGEMENTS

I would like to thank several people that have helped me through my undergraduate research career and, in particular, in writing this honor thesis.

First, Dr. Gang Zhou introduced me to the exciting world of research in computer science. He has taught me how to conduct research, formulate my idea and present my work. The idea of power-auditing in the smart home setting also originate from my research experience with him.

Next, Dr. Adwait Nadkarni has introduced to me about ideas in mobile security and has provided me the smart home devices for the experiments.

Next, Dr. Huajie Shao has provided valuable advice about my PhD application and about the machine learning part of the thesis.

Next, Dr. Eric Swartz, a mathematics professor, introduced me to my first experience in academic research. Though my thesis is outside of his research area, he has given valuable writing and other advice.

Next, Woosub Jung, a PhD student in our lab, introduced me to the idea of power-auditing IoT devices and provided me the power auditing devices for the experiment.

To all these mentors, I extend my sincerest thanks for their unwavering guidance and encouragement throughout my undergraduate research journey.

Contents

Abstract	2
Acknowledgements	3
List of Figures	7
List of Tables	8
1 Introduction	9
2 Background	11
2.1 Smart Home Devices	11
2.2 Power Auditing	12
2.3 Deployment Scenarios	12
2.4 Goal of Analysis	13
3 Power Profiling System Design	14
4 Data Collection	16
4.1 Experimental Setup	16
4.2 Smart Speakers	16
4.3 Other Devices	17
4.4 Collected Dataset	18
5 Data Processing and Analysis	20
5.1 RMS and Smoothing	20
5.2 Sample Trace Analysis	21
5.2.1 Wake Word Pattern	21
5.2.2 Audio Playback	22
5.2.3 User Conversation	22
5.2.4 Idle State	23
5.2.5 Smart Camera patterns	23
5.2.6 Smart Display patterns	24
5.3 Data Segmentation Using Sliding Window	24
5.4 Feature Extraction	25

6	ML Model and Evaluation	27
6.1	Random Forest Model	27
6.2	Classifying Power Traces: a Naive Approach	27
6.3	Device First, Then Action	29
6.4	Evaluation: Device Classification	30
6.5	Action Classification Per Device	30
6.6	Comparison with State-of-the-Art	33
6.7	System Performance	34
7	Related Works	36
7.1	Utilizing Power Side-Channel Information	36
7.2	IoT Privacy Leakage	36
7.3	Understanding Smart Home Uses	37
8	Concluding Remarks	38
8.1	Summary	38
8.2	Limitations	39
8.3	Future Works	39
	References	41

LIST OF FIGURES

3.1 System Overview	14
4.1 Experiment Setup Devices	17
4.2 Smart Voice Assistants in the Experiment	17
4.3 Other Devices in the Experiment	18
5.1 Power Consumption Data from Pre-processing Steps	21
5.2 Power Consumption Data of Wake Word Detection	22
5.3 Power Consumption Patterns of Amazon Echo	22
5.4 Power Consumption Patterns of Nest Cam During Different Actions	23
5.5 Power Consumption Patterns of Amazon Echo Show	24
6.1 Confusion Matrix for Device-action Pair Classification	28
6.2 Workflow of the Machine Learning Models	29
6.3 Confusion Matrix for Running Device Classification	31
6.4 Confusion Matrix for Idle Device Classification	31
6.5 Confusion Matrices for Action Classifications	32

LIST OF TABLES

4.1	List of Actions Performed by Smart Speakers	17
4.2	List of Actions Performed by Smart Cameras [1]	18
4.3	List of Actions Performed by Echo Show (Besides smart speaker functions) [2]	18
4.4	List of Devices and Their Actions Whose Power Traces Are Collected	19
5.1	Important Features Extracted Using <i>Tsfresh</i>	26
6.1	5-fold Cross Validation Accuracy of All Classification Tasks	28
6.2	Accuracy V.S. LightAuditor	33
6.3	Evaluation of Feature Extraction, Model Training, and Inference Time	34

Chapter 1

Introduction

The market for smart home devices is rapidly growing, and as a result, there is an increasing interest in understanding the behavior and power consumption patterns of these devices. Previous works [3, 4] have demonstrated the capabilities of power auditors in identifying Internet of Things (IoT) device actions, and with the possibility of embedding these power auditors into smart plugs [5, 6], large-scale power auditing in smart home settings could soon become a reality.

Given this context, this thesis aims to explore the research problem: **What information can we gain from the power traces of smart home devices?** To address this problem, we set out to profile the power consumption traces of smart home devices and answer the following three research questions:

1. How can we model the power consumption data collected from smart home devices?
2. What information can we infer from the data and why is it important?
3. How well can the model make these inferences?

To answer the first research question, we designed a power auditing framework that uses a current sensor to record and calculate the power consumption of smart home devices. We analyze power patterns and also employ data processing, data segmentation, and feature extraction to prepare the data for machine learning (ML)-based classification.

To answer the second research question, we observe patterns in the power consumption patterns corresponding to different device functionalities and states.

We use an ML model to infer the device type and activity based on the power consumption data. We argue the importance of these inferences from two perspectives: better understanding of smart home device mechanisms and the associated privacy risks, which includes inferring usage patterns and enabling better recommendation targeting for IoT manufacturer for example.

To answer the third research question, we evaluate the model’s performance on the dataset using accuracy and confusion matrices across multiple classification tasks. We also include system performance metrics to estimate the system’s scalability, demonstrating the practicality of our approach in a real-world setting.

The rest of the thesis is organized as follows: Chapter 2 provides a background and motivation for the research. Chapter 3 gives an overview of the power profiling system. Chapter 4 writes about the experimental setup and data collection procedures. Chapter 5 explains how we process the data and explore the patterns in the power traces. Chapter 6 explains the ML model and shows the evaluation on the dataset. Chapter 7 summarizes the related works in relevant research directions. Chapter 8 summarizes the contributions and outlines some limitations and directions for future work.

Chapter 2

Background

In this chapter, we provide an overview of the context and motivation for our research on power profiling smart home devices. We discuss the characteristics of smart home devices with respect to their power consumption patterns, review previous works on power auditing, and explore potential deployment scenarios and their implications.

2.1 Smart Home Devices

Smart home devices, such as smart speakers and security cameras have become increasingly popular due to their convenience and capabilities.

Smart speakers like Amazon Echo, Google Home, and Apple HomePod offer a wide range of functionalities, from answering queries to controlling other smart devices. When a user calls a voice assistant using its wake word, the recorded voice command is sent to a cloud server for processing, and the device reacts based on the server's response [7]. Power consumption patterns during these interactions might correspond to network access, sound playback, or increased processing requirements [8].

The Amazon Echo Show is a smart display that combines the voice assistant capabilities of Amazon's Echo series with a touchscreen display [2]. In addition to the functionalities of smart speakers, Echo Show can perform various tasks, such as streaming video content, displaying weather, news, and calendar information. This leads to distinct power pattern, which is dependent on the display content.

Smart cameras offer a range of advanced features and functionalities, such as real-time video streaming, motion detection, two-way audio communication,

night vision capabilities, and integration with other smart home devices [1]. Power consumption patterns might correspond to recording, motion detection, etc.

2.2 Power Auditing

Previous works have explored power auditing as a method for anomaly detection to mitigate botnet and information leakage attacks [4, 3, 9]. By monitoring the power consumption of devices, researchers have been able to identify abnormal patterns indicative of malicious activities or unauthorized accesses [10]. However, this thesis focuses on profiling the power consumption patterns of smart home devices under normal usage to better understand their characteristics and develop methods for device identification and behavior analysis. We use a current sensor connected in series with the devices' power supply to record the current passing through, and thus model the devices' power traces.

2.3 Deployment Scenarios

Assuming that smart plugs become ubiquitous in the future, then by integrating our power profiling framework into these smart plugs (as in [5, 6]), we can analyze the power consumption patterns of various devices.

The power auditing of smart home devices, while offering valuable insights into device usage patterns and energy consumption, raises privacy concerns due to the potential for sensitive information to be inferred from power traces.

One privacy risk is the possibility of tracking users' daily routines and habits. Researchers conduct user studies to obtain such information legally [11]. However, by analyzing power consumption patterns, companies like Amazon might infer when a user is at home, when they leave for work, or when they go to bed. This information could be used for targeted advertising, or in more nefarious scenarios, be sold to third parties without user consent.

Another concern is the potential for eavesdropping on user interactions with their smart devices [12]. For instance, if a voice assistant's power trace reveals an

increase in energy consumption during a specific time frame, it could indicate that the user was engaging with the device, possibly revealing sensitive information or personal preferences.

Additionally, power traces may reveal the type and number of devices in a household, which could be used to build detailed profiles of users and their lifestyles. Users are unaware of privacy risks from inferring based on side-channel information [13]. This information could be exploited by companies to push targeted marketing content or shared with other entities, such as insurance providers, who could use the data to adjust premiums based on perceived risk.

2.4 Goal of Analysis

Understanding the how privacy can be breached is essential in mitigating such risks. Utilizing power traces to gain sensitive information relies on the ability to analyze and interpret patterns in the data. In the analyses, we aim to achieve the following objectives.

Detecting user’s calls: As the device’s power consumption increases when processing voice commands, analyzing these patterns can reveal instances of user interaction. We aim to profile the power pattern of different voice-assistant-built-in devices responding to user call.

Device identification: By analyzing the power consumption patterns of various smart home devices, it is possible to build a profile of each device’s unique power usage characteristics. ML models can then be trained to classify devices based on these patterns. We aim to identify a device based on its power consumption data.

Device action classification: Different actions performed by a device, such as audio playback or user interaction, often result in distinct power consumption patterns. By training machine learning models to recognize these patterns, it becomes possible to classify device actions based on power traces. We aim to classify the device activity once the device is known.

Chapter 3

Power Profiling System Design

In this chapter, we give a brief overview of the power profiling framework. As shown in Figure 3.1, The system collects power consumption data of smart home devices and run preprocessing steps that prepare the data for effective inference. Then, we aim to infer the smart home device type and behavior based on the data.

Data Collection: To collect power consumption data, we use a sensor that measures the current passing through the circuit connected to the smart home device. The sensor collects current data over a specified period, capturing the device’s current patterns.

Data Preprocessing: The collected current data is processed to calculate the power consumption using the root mean square (RMS) method. Further preprocessing tasks, such as applying a smoothing filter, are performed to enhance the quality of the data by reducing noise and fluctuations.

Data Segmentation: We employ a sliding window approach to segment the preprocessed data, transforming the univariate time series into dataset instances. Each instance represents the power measurement of the device within a 2-second

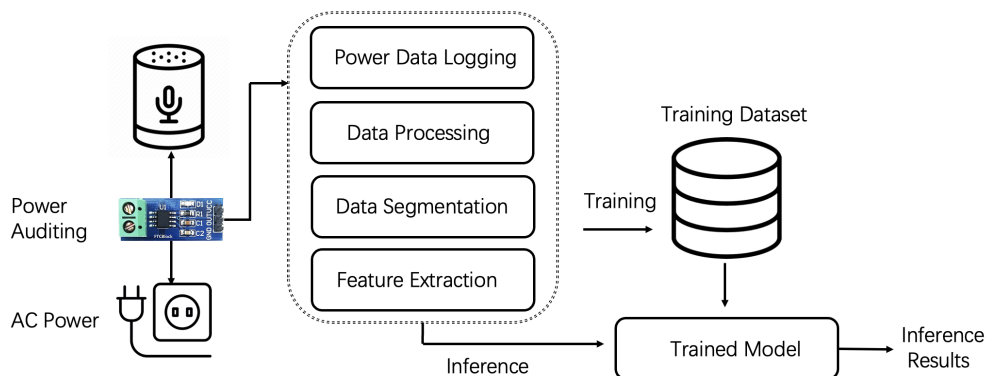


Figure 3.1: System Overview

window, providing a granular view of the energy consumption patterns.

Feature Extraction: Relevant features are extracted from the segmented data to effectively classify devices and infer their behaviors based on power consumption data. The feature extraction process aims to capture the unique characteristics of each device's power consumption, which serve as input for the subsequent classification model. Various statistical and spectral features, such as mean, standard deviation, and spectral entropy, are extracted to represent the energy consumption patterns.

Model Development and Validation: We formulate a time series classification problem to distinguish between different devices and recognize their operational patterns based on the extracted features. To this end, we employ a machine learning model based on random forest, which is well-suited for handling high-dimensional and noisy data [14].

The model's performance is evaluated using 5-fold cross-validation, to ensure its robustness and generalization capabilities, demonstrating its effectiveness in accurately identifying device types and characterizing their behaviors.

Chapter 4

Data Collection

In this chapter, we discuss the experimental setup and the devices under test.

4.1 Experimental Setup

The experimental setup consists of an ACS712 current sensor [15] that measures the current passing through the circuit connected to the smart home devices, an Arduino Uno [16] microcontroller board that reads and processes the current reading, and an M2 Macbook Pro for recording and processing the data. The ACS712 sensor is connected in series with the device’s power supply and measures the current at 55 Hz (Figure 4.1). As the connected devices perform various actions, the power consumption data is collected by ACS712, processed by Arduino Uno, and output to the laptop computer.

In the experiment, we simultaneously start logging data with Arduino script and recording from a camera, then perform various actions on the devices. The recording allows us to find the exact time that the device responds to voice commands, which is useful for data labeling and cleaning.

4.2 Smart Speakers

In the experiment, we collect power consumption data of several popular smart speakers. As listed in Figure 4.2, they are (from left to right) Apple HomePod, Google Home, Amazon Echo and Amazon Echo Dot. The experiments involve performing various actions on the devices as listed in Table 4.1. For the “user conversation” action, we asked some common daily-life questions [17]. The power

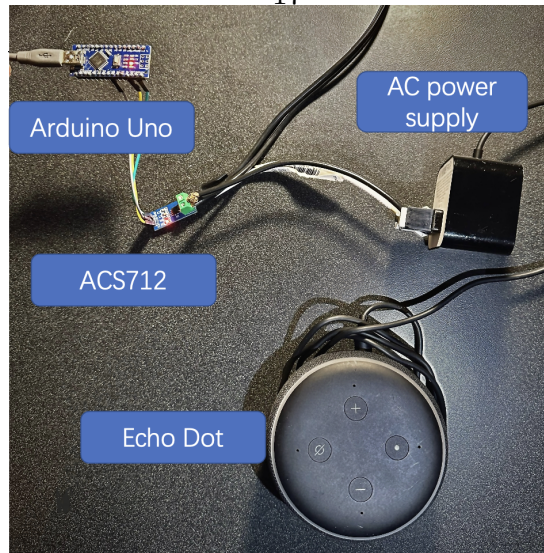


Figure 4.1: Experiment Setup Devices

Action	Description
Idle	When the device does not perform any task and is standing by for user call
Audio Playback	When the device is playing music or other audio content
Wake Word	When the device responds to user saying wake word ("Hey Siri")
User Interaction	When the device interacts with the user (such as asking about weather)

Table 4.1: List of Actions Performed by Smart Speakers

consumption data is recorded during these actions to analyze the devices' power patterns and understand how different functionalities affect the pattern.

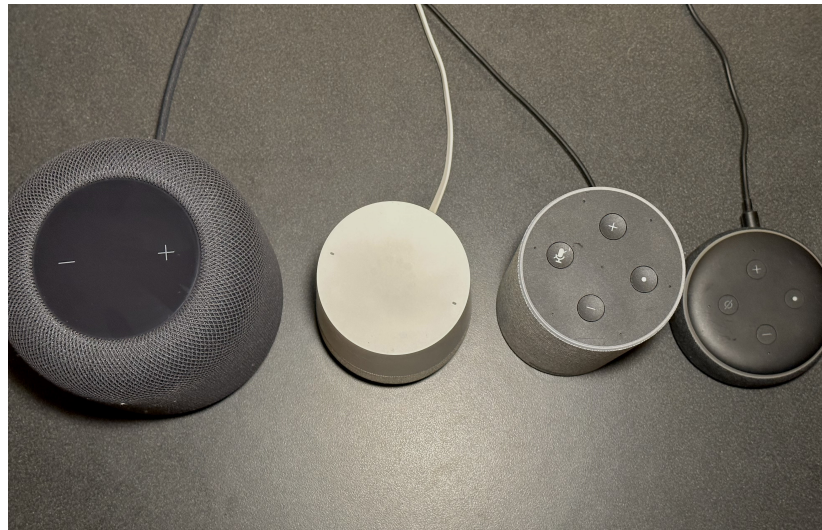


Figure 4.2: Smart Voice Assistants in the Experiment

4.3 Other Devices

In addition to smart speakers, we also collect data from other smart home devices including Amazon Echo Show and Google Nest Security Cam. These devices offer

Action	Description
Recording	Capturing and storing video footage of the monitored area
Motion Detection	Detecting movement within the field of view and initiating recording or alerts
Voice Broadcasting	Enabling remote voice communication through the camera

Table 4.2: List of Actions Performed by Smart Cameras [1]

Action	Description
Idle Display	Standing by and displaying weather, news, calendar, etc.
Video playback	When the device is playing video

Table 4.3: List of Actions Performed by Echo Show (Besides smart speaker functions) [2]

different functionalities, allowing us to explore a broader range of power consumption patterns in smart home devices. Similar to the voice assistant experiments, we perform various actions (listed in Table 4.2 and 4.3) on these devices and record their power consumption data to analyze the patterns during operation.



Figure 4.3: Other Devices in the Experiment

4.4 Collected Dataset

For the four smart speakers Amazon Echo Dot, Amazon Echo, Google Home, and Apple HomePod, we collect power consumption patterns corresponding to four different actions: Idle, Audio Playback, Wake Word Detection, and User Conversation. This results in a dataset with 16 distinct classes of instances, each representing a unique combination of device and action. The power traces from Echo Show include an action class of video playback. The power traces from Google Nest Security Cam have three classes of actions.

Device	Collected power traces
Echo	idle, audio playback, wake word, interaction
Echo Dot	same as above
Google Home	same as above
Apple HomePod	same as above
Echo Show	home screen, video playback, audio playback, interaction
Nest Cam	recording, motion detection, voice broadcasting

Table 4.4: List of Devices and Their Actions Whose Power Traces Are Collected

For all the power traces we collected, we check with the recorded video as the ground truth to mark the timestamps when each action actually started and ended. Then we manually cut out the interval when the device is not performing the labeled action. For all the actions mentioned above, each has duration that adds up to about 5 minutes.

Chapter 5

Data Processing and Analysis

In this chapter, we first describe the calculation of power from the current data, followed by the application of a smoothing filter. Next, we discuss the observed patterns in the sample traces and the segmentation of the data using a sliding window approach.

5.1 RMS and Smoothing

To calculate the power consumption from the collected current data, we use the root mean square (RMS) method, which provides a measure of the average power consumed by the devices [18]. The power is calculated using the following formula:

$$P = V \times I_{RMS} \quad (5.1)$$

where P is the power, V is the constant voltage supply, and I_{RMS} is the root mean square of the AC current value.

To further enhance the quality of the data and reduce noise, we apply an alpha filter, which is a moving average filter, to smooth the power data [19]. The alpha filter is defined as:

$$y(t) = \alpha \cdot y(t - 1) + (1 - \alpha) \cdot x(t) \quad (5.2)$$

where $y(t)$ is the smoothed power value at time t , $x(t)$ is the raw power value at time t , and α is a parameter between 0 and 1 controlling the degree of smoothing. This filter helps eliminate high-frequency fluctuations, providing a clearer representation of the underlying power patterns. Figure 5.1 shows the effect of

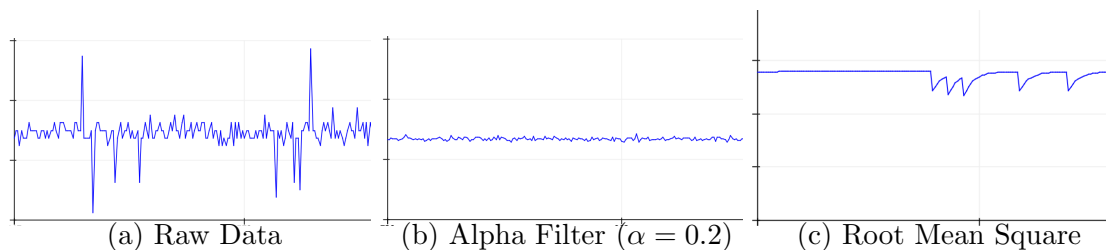


Figure 5.1: Power Consumption Data from Pre-processing Steps

alpha filter and RMS.

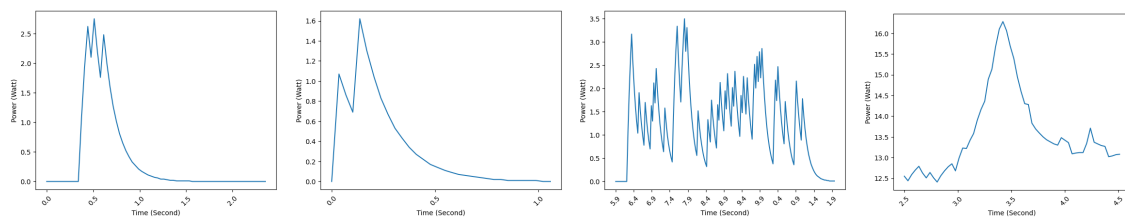
5.2 Sample Trace Analysis

Upon analyzing some sample traces, we observe distinct patterns in the power consumption data corresponding to different device functionalities and states. These patterns serve as the basis for our feature extraction and classification model, enabling the identification of device types and behaviors. We first visualize the traces of some devices and discuss their the patterns and features.

5.2.1 Wake Word Pattern

Wake word detection is the process of identifying a specific keyword or phrase, such as “Alexa” for Amazon Echo devices, “Hey Google” for Google Home devices, or “Hey Siri” for Apple HomePod devices. When the voice assistant detects its wake word, it starts listening and processing voice commands from the user [7].

In our sample traces, we observe distinct power consumption patterns across different devices when the wake word is detected (Figure 5.3). This pattern typically consists of an initial spike in current consumption, followed by a period of sustained elevated consumption as the device processes the command. The spike in consumption can be attributed to the activation of the device’s microphone and the increased processing required to analyze the user’s voice command. The sustained elevated consumption results from the device’s continuous listening and processing of the command until it completes the requested action or reverts to standby mode.



(a) Amazon Echo (b) Amazon Echo Dot (c) Google Home (d) Apple HomePod

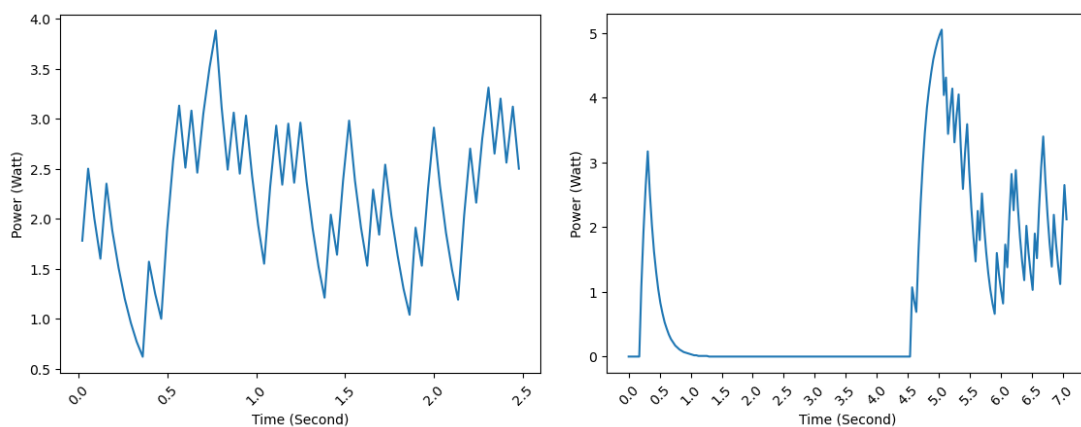
Figure 5.2: Power Consumption Data of Wake Word Detection

5.2.2 Audio Playback

Smart speakers are capable of playing news, music, podcasts, and other audio content. As shown in Figure 5.3a, the power consumption pattern during audio playback features periodic fluctuations corresponding to the audio content being played. From more traces we observe that these fluctuations are directly related to the speaker’s output volume and the complexity of the audio signal.

5.2.3 User Conversation

When a user engages in a conversation with the smart speaker, the power consumption patterns can vary depending on the duration and nature of the interaction. During a conversation, the smart speaker continuously listens for user commands while providing responses or performing requested actions. The power consumption pattern in Figure 5.3b exhibits a mix of listening, processing, and audio playback events, with spikes corresponding to the device’s microphone activation and response generation.



(a) Audio Playback

(b) User Conversation

Figure 5.3: Power Consumption Patterns of Amazon Echo

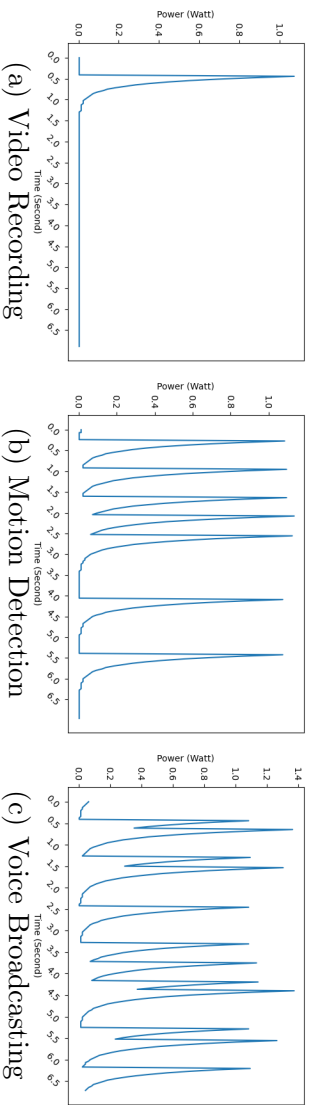


Figure 5.4: Power Consumption Patterns of Nest Cam During Different Actions

5.2.4 Idle State

In the idle state, the device does not perform any tasks and remains in standby mode, waiting for user interaction. For most of the duration of idle state, the power consumption remains stable at 0. However, some devices such as Google Home and Apple HomePod exhibit power consumption traces even when idle. This can be attributed to background processes, such as network connectivity maintenance and software updates, which consume power even when the device is not actively performing user-requested tasks.

5.2.5 Smart Camera patterns

We can observe distinct patterns corresponding to each action of the smart camera, Nest Cam, as shown in Figure 5.4.

Recording: The power consumption pattern during recording shows infrequent spikes, possibly representing the transmission of data to a remote server or local storage. In between these spikes, the power consumption remains stable.

Motion Detection: The power trace for motion detection displays intermittent spikes, indicating that the camera is processing video frames to identify motion events.

Voice Broadcasting: During voice broadcasting, the power consumption pattern exhibits very frequent and sharp spikes. These spikes can be attributed to the activation and transmission of audio data, with the variations in magnitude possibly reflecting changes in volume and duration of the audio broadcast.

5.2.6 Smart Display patterns

We observe the power traces for Echo Show and observe the following patterns.

Non-zero idle power consumption: Unlike smart speakers, the Echo Show’s power consumption never drops to zero due to its display, which is always on, even when the device is idle. This results in a baseline power consumption level that is consistently above zero.

Higher power level fluctuations: The Echo Show’s power consumption patterns generally exhibit higher levels of fluctuation compared to smart speakers (Figure 5.5). This is perhaps due to the fact that the display consumes more power than the speaker component alone.

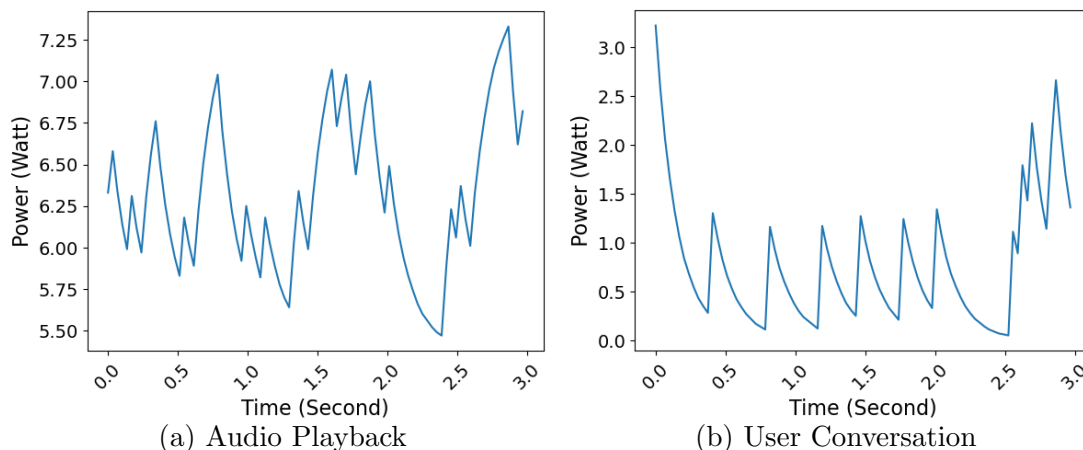


Figure 5.5: Power Consumption Patterns of Amazon Echo Show

5.3 Data Segmentation Using Sliding Window

To segment the data for further analysis, we employ a sliding window approach. The choice of window duration and overlap ratio is crucial for accurately capturing the power consumption patterns. Based on our observation that most spike patterns occur within a 2-second duration, we select a window duration of 2 seconds and an overlap ratio of 0.5. This configuration ensures that the sliding window effectively captures the rapid changes in power consumption while maintaining an adequate level of data granularity.

The sliding window segmentation process transforms the univariate time series

data into dataset instances, with each instance representing the power measurement of a device within a 2-second window. This segmented data serves as the input for the subsequent feature extraction and classification steps.

5.4 Feature Extraction

To effectively classify the power consumption patterns of smart home devices, we extract relevant features from the segmented power traces. We treat each segment as a univariate time series, consisting of sequential single data points collected at constant time intervals.

We use the *Tsfresh* [20] library to extract 777 features from each segment, including statistical features (e.g., mean, standard deviation, variance), linear trend, coefficients of Fast Fourier Transform (FFT), and Continuous Wavelet Transform (CWT).

In the training phase, *Tsfresh* automatically selects the most relevant features using statistical hypothesis tests, evaluating their importance in relation to the target variable. This process ensures that only the most informative features are retained for model training, reducing the risk of overfitting and improving the model’s generalization performance [21]. In the testing phase, features are extracted from the new, unlabeled data, and the trained model is used to make predictions about the device type and action.

Some of the important features extracted by *Tsfresh* and their definitions are presented in the Table 5.1.

By leveraging the capabilities of the *Tsfresh* library, we were able to extract and select a set of informative features that enable our machine learning model to effectively classify the smart home devices based on their power consumption patterns.

Feature	Definition
Mean	The arithmetic mean of the values in the time series.
Variance	Measures how far the values spread out from mean.
Linear trend	The slope of a straight line that best fits the time series data.
FFT coefficients	Represents the time series in the frequency domain
CWT	Decomposes the signal into different frequency components and produces a series of coefficients that describe the components' amplitude and location.
Maximum	The maximum value in the time series.
Minimum	The minimum value in the time series.

Table 5.1: Important Features Extracted Using *Tsfresh*

Chapter 6

ML Model and Evaluation

In this chapter, we discuss the choice of ML model, the model workflow and the evaluation on our dataset.

6.1 Random Forest Model

We use a Random Forest model for the classification tasks. As the dataset instances are segments of time-series with a large number of extracted features (777 in this case), Random Forest is capable of managing this complexity and avoiding overfitting. Each tree in the ensemble considers a random subset of features, which helps in capturing various aspects of the data without being overwhelmed by the high dimensionality [14].

Some previous works [9, 3, 4, 22] use deep learning techniques for identifying power traces. Our chosen machine learning model, Random Forest, is capable of delivering satisfactory results even with the limited data available (about 250 instances per device-action class), while deep learning models typically require a large amount of training data and more computational resources to achieve good performance. In a following section, we compare our performance with a deep learning based model [4].

6.2 Classifying Power Traces: a Naive Approach

A naive approach of classifying smart home devices and their actions would be to treat all device-action pairs as distinct classes and train a single model to identify both the device and the action performed in a power trace.

Task	Accuracy
Device-Action pairs:	0.8132
Device (Running)	0.9296
Device (idle)	0.7939
Echo Actions	0.8897
EchoDot Actions	0.9377
Google Actions	0.843
HomePod Actions	0.9717
Nest Actions	0.9623
EchoShow actions	0.8956

Table 6.1: 5-fold Cross Validation Accuracy of All Classification Tasks

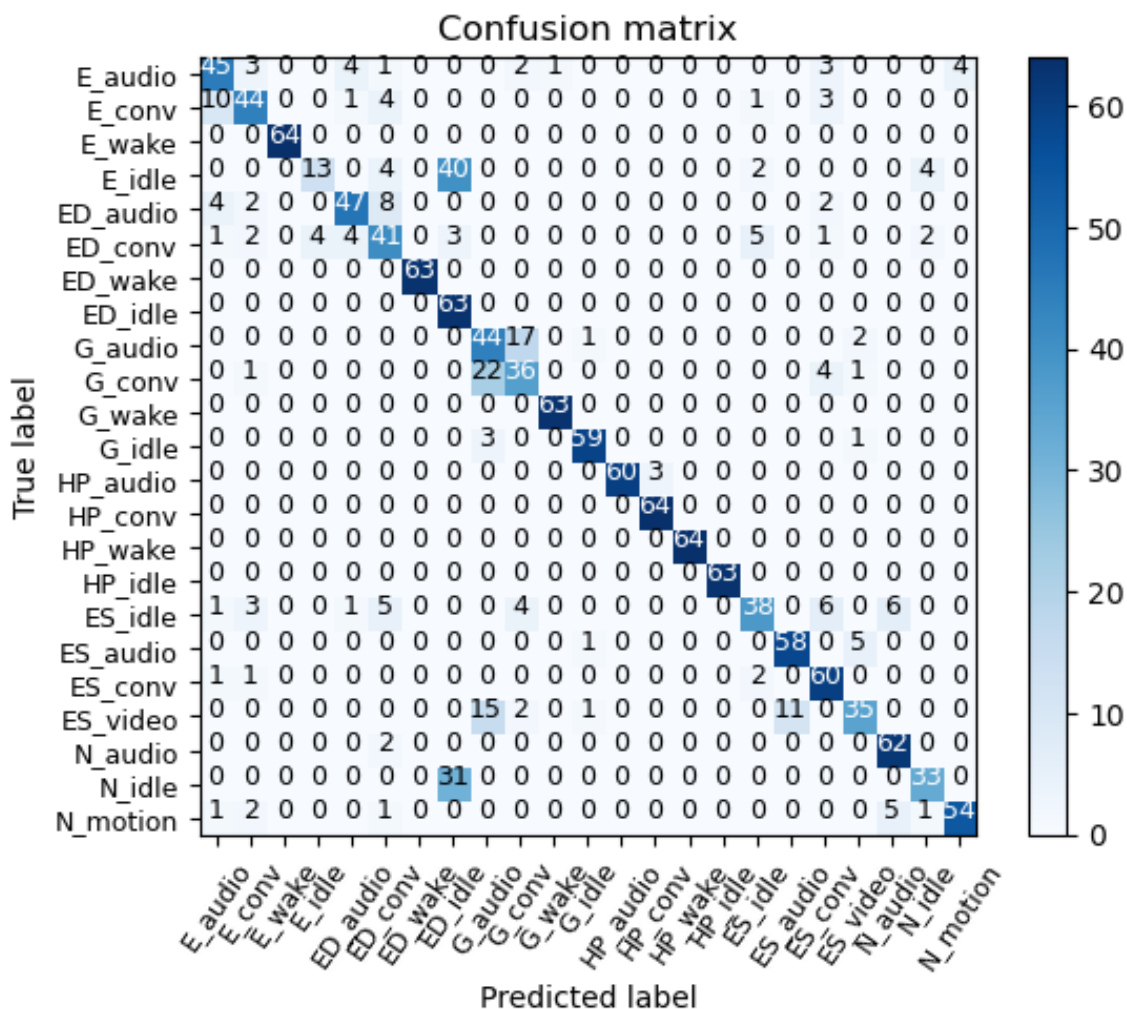


Figure 6.1: Confusion Matrix for Device-action Pair Classification

However, this approach leads to poor accuracy of 81% (Table 6.1). As demonstrated by the confusion matrix (Figure 6.1), idle power traces of several devices are easily misclassified. This is reasonable since most devices exhibit similar or zero power consumption when idle.

6.3 Device First, Then Action

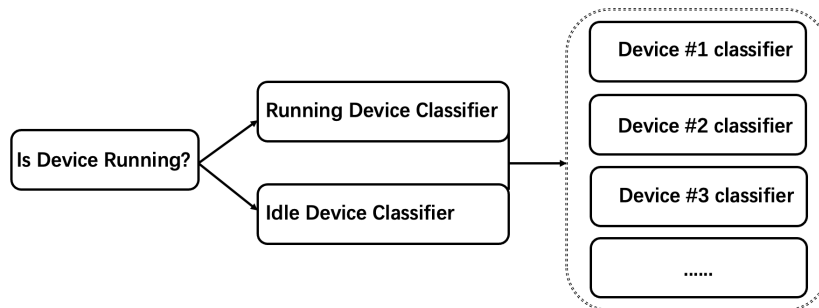


Figure 6.2: Workflow of the Machine Learning Models

Instead, we propose a two-step classification process to first identify the device and then classify the action the device is performing. We train two device classifiers: one for classifying power traces of running devices, and another for classifying devices based on their idle power traces. Additionally, we train an action classifier for each device. The workflow of this process is illustrated in Figure 6.2.

Given a power trace, we use either the running device classifier or the idle device classifier to identify the device. Determining whether a device is in a running or idle state is relatively straightforward. The rough power consumption data can provide sufficient information to differentiate between these states, without the need to examine the fine-grained collected power traces (which are collected at 55 Hz). Once the device is identified, we then use the corresponding action classifier for that device to identify the action being performed. As shown in the next sections, this two-step classification process provides a more accurate and robust method for inferring both the device type and the action performed.

6.4 Evaluation: Device Classification

We evaluate the performance of the two device classifiers and the action classifiers for each device.

The model classifies running device traces with a 92.96% accuracy. The confusion matrix for running device classification (Figure 6.3) reveals satisfactory performance in most cases. Most instances are classified correctly; however, there are notable misclassifications between Amazon Echo and Echo Dot. This may be attributed to the similar software and possibly hardware of both devices, resulting in similar power traces when running.

The model classifies running device traces with a 79.39% accuracy. The confusion matrix for idle device classification (Figure 6.4) exhibits serious misclassifications between Amazon Echo and Echo Dot. Additionally, the classifier often identifies power traces of the Nest Cam as Amazon Echo. Upon inspecting the dataset, we found that both classes display long durations of zero power consumption, which might be the reason for these misclassifications.

Thus, in the actual deployment, we can set the system to make inferences from device running power traces instead of the idle traces. Based on empirical calculation, if we can classify the device at accuracy of 93%, and we can classify the device action at accuracy 90%, then the estimated accuracy of correct device-action pair identification is $0.93 * 0.9 = 0.84$, outperforming the naive classifier.

6.5 Action Classification Per Device

Referring to confusion matrices, we discuss the model’s performance on certain devices where misclassifications occur.

Amazon Echo, Echo Dot, and Google Home: There are minor misclassifications between the conversation and audio playback actions for these devices (Figure 6.5a, 6.5b, 6.5c). This suggests that the power consumption patterns for these two activities are quite similar, making it challenging to differentiate them solely based on the power traces.

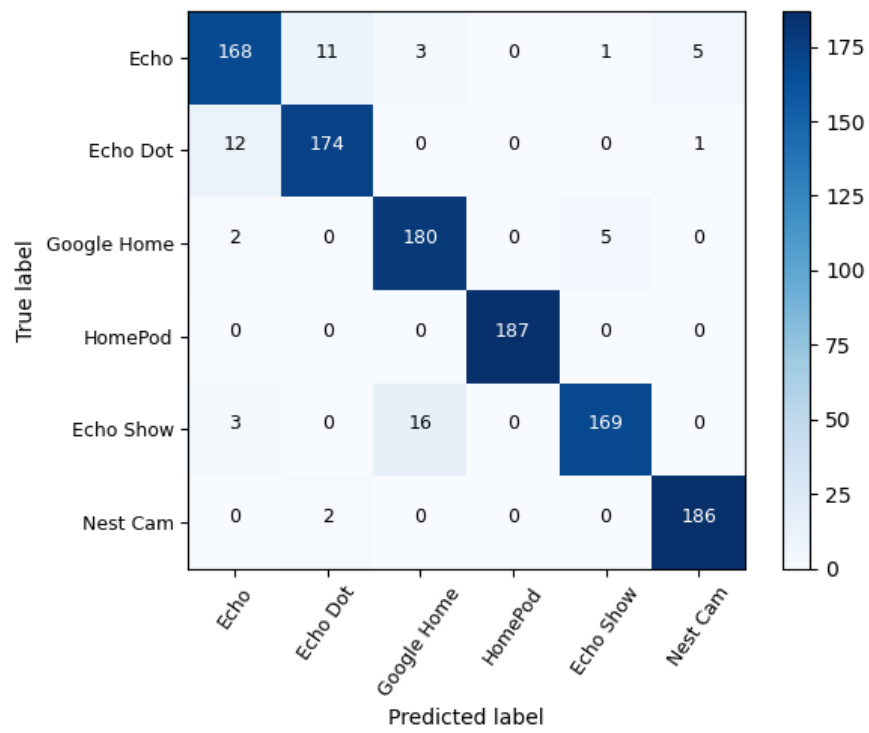


Figure 6.3: Confusion Matrix for Running Device Classification

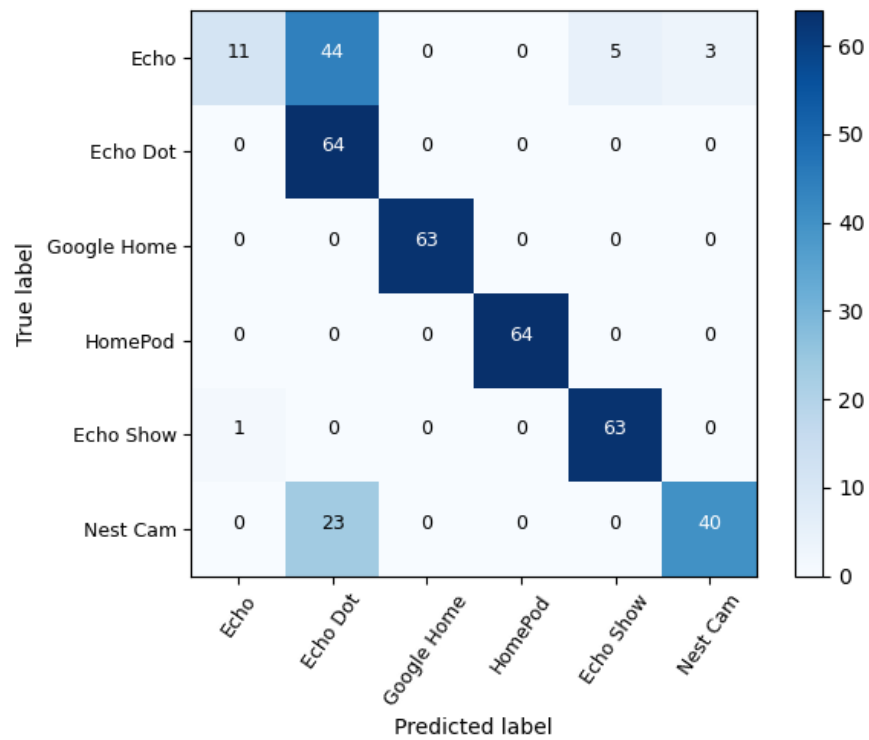
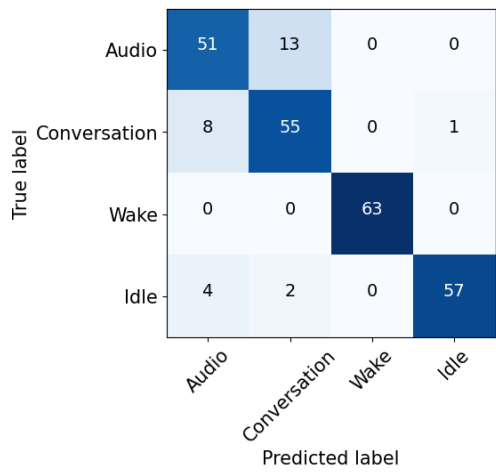
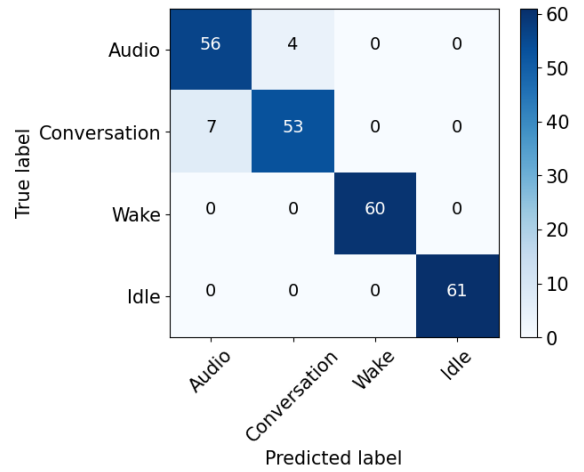


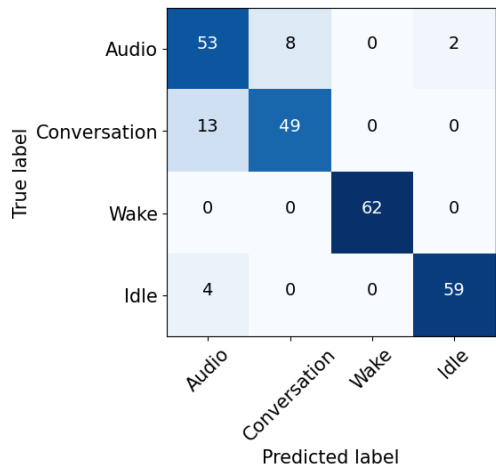
Figure 6.4: Confusion Matrix for Idle Device Classification



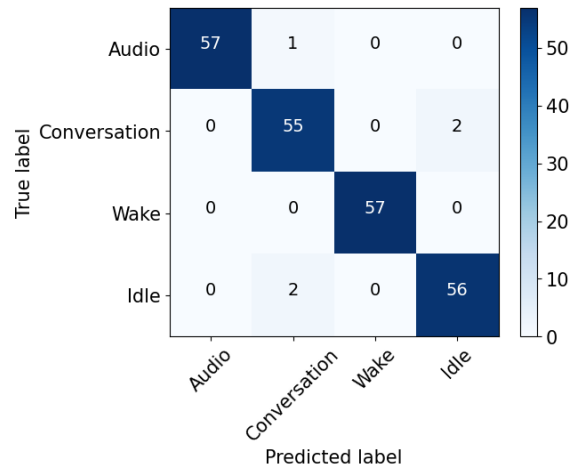
(a) Amazon Echo



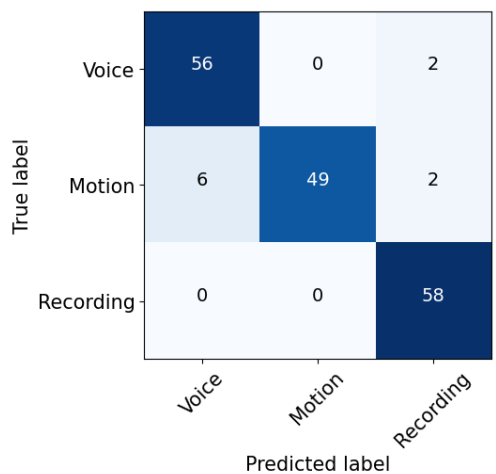
(b) Amazon Echo Dot



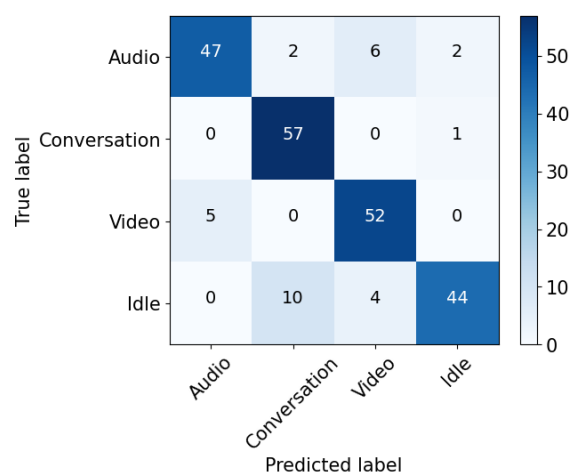
(c) Google Home



(d) Apple HomePod



(e) Nest Cam



(f) Amazon Echo Show

Figure 6.5: Confusion Matrices for Action Classifications

Model-Task	Accuracy
Light Auditor	0.8344
Ours-Classifying device	0.9296

Table 6.2: Accuracy V.S. LightAuditor

Echo Show: The most notable misclassification for the Echo Show is when the conversation action is misclassified as idle (Figure 6.5f). This can be attributed to the fact that during the idle state, the smart display is continuously on, displaying content such as weather updates and notifications. This results in an overlapping power consumption pattern between the conversation and idle states, making it harder for the classifier to distinguish between them.

For the other devices, the models identify the device actions with good accuracy.

6.6 Comparison with State-of-the-Art

We compare our proposed model with *Light Auditor*, a power-auditing-based system that identifies actions of a smart bulb [4]. The *Light Auditor* also processes time series data, but it converts data segments into 2D images using a Continuous Wavelet Transform (CWT) and applies Convolutional Neural Networks (CNN) to classify them into different behavior classes. However, this approach yielded a lower accuracy of 0.83, compared to our model’s accuracy of 0.93.

There are a couple of possible reasons for the lower performance of the *Light Auditor* approach in our case.

The dataset we collected might be too small to effectively train the deep neural network used in the *Light Auditor* model. Deep learning models typically require large amounts of data to optimize their parameters and generalize well to new instances. Our dataset’s limited size could have negatively impacted the model’s ability to learn the necessary representations for accurate cross-device identification.

The *Light Auditor* model was originally designed to identify actions of a smart

bulb, not to perform cross-device identification. This means that the model’s architecture and features may be more suited to capturing the nuances of smart bulb actions rather than differentiating between various smart devices. As a result, its performance may be suboptimal when applied to our cross-device identification task.

Given these limitations, our approach, which leverages feature extraction and a Random Forest classifier, demonstrates superior performance in accurately identifying the devices and their activities.

In addition, our model is much lighter in terms of computational resources since it does not require a GPU. Overall, our proposed model provides a viable alternative to deep learning-based approaches for power auditing systems.

6.7 System Performance

Evaluating the performance of a machine learning model is not only about its accuracy but also about its efficiency in terms of resource utilization, such as time and computational power.

Our hardware, the ACS712 sensor, generates 55 current values per second. These values are preprocessed through RMS and smoothing filters before being segmented. These processes have negligible time costs and are not throughput bottlenecks.

The time-consuming processes include feature extraction, model training, and model running. In our evaluation, 4,400 instances take 59 seconds for feature extraction and 9 seconds for model training. For 1,400 instances, it takes 2 seconds to make inferences. The estimated feature extraction, training, and inference time per instance are presented in Table 6.3.

Process	Instances	Time (s)	Time per Instance (ms)
Feature Extraction	4,400	59	13.41
Model Training	4,400	9	2.05
Model Inference	1,400	2	1.43

Table 6.3: Evaluation of Feature Extraction, Model Training, and Inference Time

Recall that each dataset instance is converted from time series by segmentation using sliding window of size 2 seconds and overlapping ratio 0.5. This means each sensor generates 1 instance per second. Then, our testbed environment (M2 MacBook Pro) allows real-time processing of power traces from $1000/13.41 = 75$ sensors, which is well above the number of socket plugs per household.

Chapter 7

Related Works

In this work, we present the first systematic study on IoT device identification using power consumption patterns, emphasizing the novelty of our approach. Furthermore, we discuss the privacy risks associated with power traces, highlighting the potential for sensitive information to be inferred from these data. With this foundation established, we proceed to review other works in the domains of power auditing, IoT privacy, and smart home usage.

7.1 Utilizing Power Side-Channel Information

A number of works aim to infer device activity based on power side-channel information. Myridakis et al. [23] proposed a circuit design for monitoring IoT devices against DOS attacks. Similarly, Li et al. [24], Majumder et al. [10], Jung et al. [3] and [9] analyzed the power consumption pattern of IoT devices to mitigate security threats. Similarly, they profile the devices' power pattern under normal usage and under attack and identify the attack pattern from power traces. In [22], Cronin et al. infer the location on the smart phone touchscreen where the user touched through charging power traces.

7.2 IoT Privacy Leakage

A number of works explore and mitigate privacy risks of IoT devices from different perspectives. Jung et al. [4] use power auditing information to mitigate information leakage attacks. Yang et al. [25] analyze smartphone activities based on USB power side channels. Sayakkara et al. [26] use electromagnetic side-channel

information to infer software behaviors of IoT devices. Zhang et al. [7] explore two types of voice skill squatting attacks on Amazon Echo that potentially leak users' information.

7.3 Understanding Smart Home Uses

One potential use scenario of our work is to profile users' usage patterns. A number of works aim to profile users for various purposes. Khan et al. [27] profile user pattern to better manage the energy use in a smart home setting. Bentley et al. [11] conduct user studies to explore long-term habits of using smart home devices. Wei et al. [28] use reinforcement learning models to profile residents' behavior in commercial buildings.

Other works have identified security and privacy issue in smart home systems. Ling et al. [29] delve into the smart plug's architecture and networking protocols. Panwar et al. [12] outline typical threats to a smart home system.

Chapter 8

Concluding Remarks

8.1 Summary

In this thesis, we explore the power consumption patterns of smart home devices and the information that can be gained from these patterns. Our contributions can be summarized in three main aspects:

First, we have developed a framework that leverages current sensors to effectively infer both the type and activity of smart home devices. This framework is built upon a solid foundation of data collection, preprocessing, feature extraction, and machine learning techniques, which allows us to analyze the power consumption patterns of various smart home devices and infer crucial information from them.

Second, we have collected an extensive power consumption dataset of more than 2 hours in total, covering different smart devices performing different actions. This dataset has provided us with valuable insights into the power consumption patterns of smart home devices and enabled us to establish a strong connection between these patterns and the devices' types and activities.

Third, our analysis has demonstrated that, by utilizing the power consumption data, we can effectively infer the type of smart device being used. Furthermore, once the device type is known, we can also make inferences about the user's behavior, which has important implications for understanding smart home usage patterns and potential privacy risks.

8.2 Limitations

However, we acknowledge that there are certain limitations to our approach. One limitation is that auditing the power consumption through circuit is not applicable to devices that run on battery power. This constraint may limit the applicability of our approach to certain smart home scenarios where battery-powered devices are prevalent.

Another limitation is that our supervised learning model requires pre-training, which means that unseen devices cannot be classified correctly unless they have been trained first [30]. This applies to most ML-based anomaly detection systems. This constraint may limit the scalability of our approach, especially when dealing with a constantly evolving landscape of smart home devices.

Another notable limitation of our model is the lack of fine-grained information to distinguish between states of devices that have very similar power patterns. For example, Google Home’s conversation and audio playback activities show several misclassifications. This indicates that our approach might not be sufficient for accurately discerning between certain device states that exhibit close power consumption patterns.

8.3 Future Works

To address the limitations in future work, first, researchers could explore alternative methods for monitoring and analyzing power consumption in battery-powered devices. One possible direction is to investigate the use of built-in battery management systems or other onboard sensors that can provide insights into the device’s power usage patterns, as researchers suggest in [31].

Another direction would be to explore the potential of combining power-auditing with other side-channel information, such as electromagnetic and acoustic data, to improve the accuracy of inferring device activities. By leveraging multiple side-channel sources such as network traffic [32, 33], it might be possible to capture more subtle differences between device states, leading to a more robust

and reliable classification performance.

To identify devices which exhibit only subtle differences in power traces, a possible direction for future work could be to explore the use of deep learning models, such as Long Short-Term Memory (LSTM) networks [34], which have shown promise in handling time series data and capturing complex temporal relationships. These models might be better equipped to identify and differentiate between similar power patterns, leading to more accurate classifications of device activities.

REFERENCES

- [1] “Nest cam (outdoor or indoor, battery).” [Online]. Available: https://store.google.com/product/nest_cam_battery?hl=en-US
- [2] Wikipedia, “Amazon Echo Show — Wikipedia, the free encyclopedia,” <http://en.wikipedia.org/w/index.php?title=Amazon%20Echo%20Show&oldid=1147832705>, 2023, [Online; accessed 19-April-2023].
- [3] W. Jung, H. Zhao, M. Sun, and G. Zhou, “Iot botnet detection via power consumption modeling,” *Smart Health*, vol. 15, p. 100103, 2020.
- [4] W. Jung, K. Cui, K. Koltermann, J. Wang, C. Xin, and G. Zhou, “Light auditor: Power measurement can tell private data leakage through iot covert channels,” in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’22. New York, NY, USA: Association for Computing Machinery, 2023, p. 518–532. [Online]. Available: <https://doi.org/10.1145/3560905.3568535>
- [5] M. S. Ahmed, A. Mohamed, R. Z. Homod, H. Shareef, A. H. Sabry, and K. Bin Khalid, “Smart plug prototype for monitoring electrical appliances in home energy management system,” in *2015 IEEE Student Conference on Research and Development (SCORed)*, 2015, pp. 32–36.
- [6] S.-H. Lee and C.-S. Yang, “An intelligent power monitoring and analysis system for distributed smart plugs sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 7, p. 1550147717718462, 2017.
- [7] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, “Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems,” in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1381–1396.
- [8] D. Giese and G. Noubir, “Amazon echo dot or the reverberating secrets of iot devices,” in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021, pp. 13–24.
- [9] W. Jung, Y. Feng, S. A. Khan, C. Xin, D. Zhao, and G. Zhou, “Deepauditor: Distributed online intrusion detection system for iot devices via power side-channel auditing,” in *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2022, pp. 415–427.

- [10] A. J. Majumder, J. D. Miller, C. B. Veilleux, and A. A. Asif, “Smart-power: A smart cyber-physical system to detect iot security threat through behavioral power profiling,” in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020, pp. 1041–1049.
- [11] F. Bentley, C. Luvogt, M. Silverman, R. Wirasinghe, B. White, and D. Lottridge, “Understanding the long-term use of smart speaker assistants,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–24, 2018.
- [12] N. Panwar, S. Sharma, S. Mehrotra, Łukasz Krzywiecki, and N. Venkatasubramanian, “Smart home survey on security and privacy,” 2019.
- [13] S. Zheng, N. Apthorpe, M. Chetty, and N. Feamster, “User perceptions of smart home iot privacy,” *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, nov 2018. [Online]. Available: <https://doi.org/10.1145/3274469>
- [14] M. Belgiu and L. Drăguț, “Random forest in remote sensing: A review of applications and future directions,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271616000265>
- [15] Arduino, “Acs 712 current sensor,” <https://create.arduino.cc/projecthub/instrumentation-system/acs712-current-sensor-87b4a6>, 2022.
- [16] T. A. Team, “Uno r3: Arduino documentation.” [Online]. Available: <https://docs.arduino.cc/hardware/uno-rev3>
- [17] A. C. Madrigal, “20 questions with google’s assistant and apple’s siri,” May 2017. [Online]. Available: <https://www.theatlantic.com/technology/archive/2017/05/20-questions-with-googles-assistant-and-apples-siri/527091/>
- [18] V. Miron-Alexe, “Comparative study regarding measurements of different ac current sensors,” in *2016 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, 2016, pp. 1–6.
- [19] Wikipedia, “Alpha beta filter — Wikipedia, the free encyclopedia,” <http://en.wikipedia.org/w/index.php?title=Alpha%20beta%20filter&oldid=1087166723>, 2023, [Online; accessed 19-April-2023].
- [20] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package),” *Neurocomputing*, vol. 307, pp. 72–77, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231218304843>

- [21] Z. Yang, I. A. Abbasi, E. E. Mustafa, S. Ali, and M. Zhang, “An anomaly detection algorithm selection service for iot stream data based on tsfresh tool and genetic algorithm,” *Security and Communication Networks*, vol. 2021, pp. 1–10, 2021.
- [22] P. Cronin, X. Gao, C. Yang, and H. Wang, “Charger-Surfing: Exploiting a power line Side-Channel for smartphone information leakage,” in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 681–698. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity21/presentation/cronin>
- [23] D. Myridakis, P. Myridakis, and A. Kakarountas, “A power dissipation monitoring circuit for intrusion detection and botnet prevention on iot devices,” *Computation*, vol. 9, no. 2, p. 19, 2021.
- [24] F. Li, Y. Shi, A. Shinde, J. Ye, and W. Song, “Enhanced cyber-physical security in internet of things through energy auditing,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5224–5231, 2019.
- [25] Q. Yang, P. Gasti, G. Zhou, A. Farajidavar, and K. S. Balagani, “On inferring browsing activity on smartphones via usb power analysis side-channel,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1056–1066, 2016.
- [26] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, “Leveraging electromagnetic side-channel analysis for the investigation of iot devices,” *Digital Investigation*, vol. 29, pp. S94–S103, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1742287619301616>
- [27] M. Khan, M. T. R. Khan, M. M. Saad, and D. Kim, “A user profile-based smart home energy management system,” in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 646–651.
- [28] P. Wei, S. Xia, R. Chen, J. Qian, C. Li, and X. Jiang, “A deep-reinforcement-learning-based recommender system for occupant-driven energy optimization in commercial buildings,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6402–6413, 2020.
- [29] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, “Security vulnerabilities of internet of things: A case study of the smart plug system,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1899–1909, 2017.
- [30] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.

- [31] L. He, Y. Lee, E. Kim, and K. G. Shin, “Environment-aware estimation of battery state-of-charge for mobile devices,” in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 227–236. [Online]. Available: <https://doi.org/10.1145/3302509.3313782>
- [32] R. A. Sharma, E. Soltanaghaei, A. Rowe, and V. Sekar, “Lumos: Identifying and localizing diverse hidden IoT devices in an unfamiliar environment,” in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 1095–1112. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity22/presentation/sharma-rahul>
- [33] J. Heo, S. Gil, Y. Jung, J. Kim, D. Kim, W. Park, Y. Kim, K. G. Shin, and C.-H. Lee, “Are there wireless hidden cameras spying on me?” in *Proceedings of the 38th Annual Computer Security Applications Conference*, ser. ACSAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 714–726. [Online]. Available: <https://doi.org/10.1145/3564625.3564632>
- [34] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM fully convolutional networks for time series classification,” *IEEE Access*, vol. 6, pp. 1662–1669, 2018. [Online]. Available: <https://doi.org/10.1109/2Faccess.2017.2779939>