Physical Side-Channel Attacks against Intermittent Devices

Muslum Ozgur Ozmen
Purdue University
mozmen@purdue.edu

Habiba Farrukh
University of California Irvine
habibaf@uci.edu

Z. Berkay Celik
Purdue University
zcelik@purdue.edu

ABSTRACT

Intermittent (batteryless) devices operate solely using energy harvested from their environment. These devices turn on when they have energy and turn off during energy scarcity. Intermittent devices have recently become increasingly popular in smart buildings, manufacturing plants, and medical implantables as they eliminate the need for battery replacement and enable green computing. Despite their growing adoption in critical applications, the privacy implications of intermittent devices remain largely unexplored.

In this paper, we introduce a novel remote side-channel attack. Our observation is that the network packet frequency of an intermittent device can be exploited to learn its turn-on/off patterns. From these patterns, we can infer the energy availability of a device, which reveals privacy-sensitive information about its operating environment, e.g., the presence or absence of individuals.

To realize our attack, we develop a three-stage hierarchical inference framework that leverages the timestamped network packet sequence of intermittent devices. Our framework automatically extracts a set of temporal features from inter-packet-arrival timings. It then employs a series of models to uncover (1) whether a target intermittent device is present in the environment, (2) its energy harvester type (e.g., vibration or water flow), and (3) its energy availability conditions (e.g., high-vibration or no-vibration).

To validate our attack's effectiveness, we conduct experiments in two environments: a smart home and a miniature manufacturing plant equipped with three intermittent devices powered by solar energy, vibration, and temperature. By analyzing their energy availability patterns, we are able to infer user activities and presence in the smart home and the robot's movement patterns in the manufacturing plant with an average accuracy of 85%. This sensitive information enables an adversary to launch domain-specific attacks, such as burglarizing a smart home when the user is asleep or timely tampering with plant sensors to cause maximum damage.

KEYWORDS

Intermittent devices, side-channel attacks, privacy

1 INTRODUCTION

Traditional IoT devices (sensors and actuators) used in diverse applications, such as smart buildings, wearables, industrial control systems (ICS), and medical implantables, rely on batteries that demand regular maintenance and replacement. To address this, there has been a growing interest in intermittent devices, which harvest energy (e.g., solar, vibration, thermal energy), thereby eliminating the need for batteries [4, 42, 56]. These devices have remarkable advantages as they (a) eliminate battery replacement costs, (b) reduce battery waste, (c) enable applications that would be otherwise impractical (e.g., changing an implantable medical device's battery may require surgery), and (d) provide redundancy for fault tolerance in safety-critical applications due to their low maintenance.

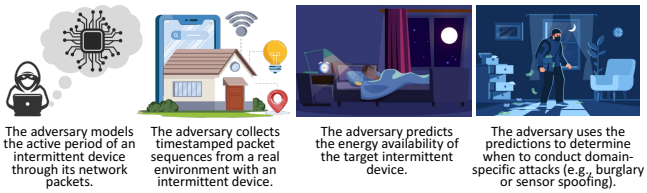


Figure 1: A motivating example for our physical side-channel attacks against intermittent devices.

Intermittent devices turn on when they have enough energy and turn off when the energy is scarce. Before turning off, they perform regular checkpoints to store run-time program states such as register, stack, and global variables in non-volatile memory [56]. Intermittent devices often have networking capabilities. When they turn on, they report sensor measurements over a low-energy communication channel (e.g., Zigbee, BLE) to a hub [38, 42, 43, 47, 49, 58]. For instance, an intermittent device harvests vibration energy from a motor (e.g., an ICS robot) and reports the motor's temperature and vibration to a hub over BLE [1]. This device turns on and reports sensor readings when the motor generates vibration. It then turns off when the motor does not generate vibration. This execution model makes this intermittent device suitable for monitoring the motor's health during the motor's operation.

While intermittent devices are increasingly used in diverse applications, their security and privacy remain unexplored. Initial works on intermittent device security are limited to RFID tags, proposing RFID tag fingerprinting and authentication to prevent counterfeits [40, 75]. A line of work focuses on extracting cryptographic keys via side-channels and malware [44, 46, 76, 85, 92]. They also propose defenses against such attacks through secure checkpoint architectures [51–53, 81] and remote attestation [25, 77]. Recent work uses compromised radio frequency harvesting devices to conduct side-channel attacks against mobile devices [67]. Yet, privacy leakages that occur due to the fluctuations in the energy availability of intermittent devices have not yet been investigated.

In this paper, we conduct the first remote side-channel attack targeting the energy availability of intermittent devices to infer privacy-sensitive information related to their operating environment. Our main observation is that an adversary who learns the times an intermittent device turns on and off (its active period) can infer the energy available to the intermittent device. The network packets an intermittent device transmits inevitably reveal if the device is on since a transmitted packet implies the device is on at that

Proceedings on Privacy Enhancing Technologies 2024(3), 461–476

https://doi.org/10.56553/popets-2024-0088

time. Thus, a remote adversary can learn the intermittent device's energy availability by observing the device's active period through its inter-packet-arrival timings (the time difference between two consecutive network packets transmitted by the device).

Such information is privacy-sensitive since the device's energy availability depends on its operating environment and user activities. To illustrate, Figure 1 shows an example where a solar-harvesting intermittent device [23] is deployed in a smart home for temperature and humidity monitoring. This device harvests energy from indoor light and sends periodic sensor measurements (e.g., every seven seconds) to an IoT hub over BLE. From this device, an adversary can infer the light conditions within the home, indicating whether the users are at home, watching TV, or sleeping. This would allow an adversary to conduct physical attacks (e.g., burglary or kidnapping) or use this information for targeted advertising. As another example, we consider a water flow harvesting device in a water treatment plant. This device periodically reports the temperature, pressure, and flow rate in pipes [33] to a control center. From this device, an adversary can infer the treatment plant's water flow rate. This knowledge allows the adversary to learn when the flow rate is high and manipulate the plant's actuators or configuration parameters at that time. For instance, an attacker injected malicious actuation commands to pumps in the Maroochy water plant and caused sewage to spill out into waterways [2]. An adversary who knows when the water flow rate is high can conduct such attacks timely to poison a higher amount of water.

Inferring privacy-sensitive information from the active period of intermittent devices has three main challenges. First, the adversary must distinguish the intermittent devices from the traditional devices with batteries since they usually co-exist in real-world environments. Second, the adversary must identify the intermittent device's energy harvester since the privacy-sensitive information that can be inferred is tightly coupled with the device's harvester. However, this is a challenging task since similar energy conditions with different harvesters may cause devices to have similar active periods. Lastly, the active period of intermittent devices is influenced by many user-configurable factors unique to intermittent devices, such as capacitor sizes and packet transmission intervals. Thus, the adversary's energy inference algorithm must integrate these factors to accurately learn the device's active period.

We address these challenges through a new side-channel attack with two phases, (1) offline intermittent device analysis and (2) online hierarchical inference. In the offline phase, we first conduct controlled experiments to uncover the causal structure between the intermittent device's active period and its user-configurable properties (e.g., firmware, capacitor size). To this end, we apply dynamic time warping on inter-packet-arrival timings to identify the intermittent device properties causally related to the device's active period. We next conduct dynamic grid testing on the intermittent device to collect a dataset that contains timestamped packet sequences while mutating the device properties causally related to its active period. This allows profiling the device's active period with the identified device properties, different energy harvesters, and energy availability conditions. Lastly, we model the intermittent device's active period through its inter-packet-arrival patterns. For this, we introduce a three-stage hierarchical classification framework to identify if an intermittent device is present in the environment, its energy harvester type, and energy availability conditions. Our framework implements an automated feature selection technique to derive temporal features from inter-packet-arrival timings.

In the online phase, we use a network sniffer to remotely collect timestamped packet sequences from devices. We separate each device's packets based on their unique identifiers (e.g., MAC addresses). Lastly, we leverage our classification framework to continuously infer the intermittent device's energy availability.

We evaluate our attack in a real smart home and (Fischertechnik) manufacturing plant testbed. Two real intermittent devices, a solar-harvesting and a temperature-harvesting device, are deployed to the smart home to monitor temperature and humidity. A vibration-harvesting device is deployed to the manufacturing plant to monitor the plant's environmental conditions. Our results show that an adversary can identify the intermittent devices in the environment with 98% accuracy and recognize their harvester with 92% accuracy. The adversary then infers (1) the light level in the smart home with 80% accuracy from the solar-harvesting device to learn if the user is sleeping or not at home, (2) the oven's temperature level with 87% accuracy from the temperature-harvesting device to profile the user's cooking activities, and (3) the robot's movement patterns with 88% accuracy from the vibration-harvesting device to learn the time the robot is operating. Our attack requires minimal effort, where 2 mins of sniffing is sufficient to infer the device's energy availability conditions. We also evaluate our attack's accuracy when the intermittent device deploys defenses that protect against network analysis attacks, such as traffic reshaping and injection. Our attack achieves, on average, only 2.7% lower accuracy as these defenses fail to hide the intermittent device's active period.

In summary, we make the following contributions:

- We introduce a new remote side-channel attack targeting the energy availability of intermittent devices. Our attack leverages the intermittent device's active period through its inter-packet-arrival timings to infer privacy-sensitive information related to the device's operating environment (e.g., user presence and robot movements).
- We develop a three-stage hierarchical inference framework that profiles an intermittent device's active period patterns to identify if an intermittent device is present in an environment, its harvester, and its energy availability conditions.
- We extensively evaluate our attack with real intermittent devices connected to three different energy harvesters (solar energy, vibration, and temperature) deployed in a smart home and a manufacturing plant[1]. Our experiments show that an adversary can infer user activities and presence in the smart home and the robot's movement patterns in the manufacturing plant with an average accuracy of 85%.
- We evaluate the feasibility of existing defense strategies against our attacks and show that they are ineffective. We then propose two defenses unique to intermittent devices to protect them against our side-channel attacks.

[1]We make our datasets available at https://github.com/purseclab/intermittent-traces to foster future research on intermittent device security and privacy.

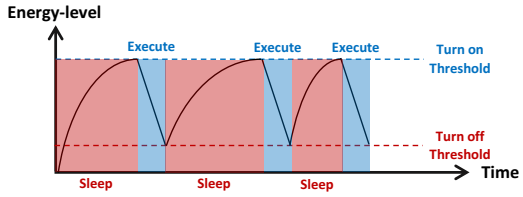


Figure 2: Illustration of intermittent device execution.

2 BACKGROUND AND DEFINITIONS

Advancements in energy harvesting enabled intermittent (self-powered, batteryless) devices that operate using energy harvested from their environment, eliminating the cost required to maintain batteries and enabling green computing [4, 56, 60].

Hardware and Execution Model. An intermittent device includes a low-power processing unit (microcontroller), sensors, an energy harvester, a power management integrated circuit (PMIC), and a low-power radio for communication. Figure 2 illustrates their execution model, where they turn on when they have enough energy to run their software and then turn off (or go into a deep sleep mode) until they have enough energy.

This execution model brings challenges in maintaining a consistent control flow and memory as the device's energy may be depleted during computation. To handle such cases, intermittent devices either restart their execution from the `main` function [20] or use *checkpoints* [57, 79]. Checkpoints involve saving the program state (register, stack, and global variable values) in non-volatile memory. Thus, the devices frequently save program states in checkpoints, and in a power failure, they continue their execution from the last checkpoint. Devices with a larger computation task use checkpoints to ensure program progress, whereas sense-and-send applications usually restart execution from the `main` function.

Power Management. Intermittent devices manage their energy usage to ensure they have enough energy to operate before turning on. For this, they use a turn-on threshold to accumulate energy without consuming any. Without this threshold, the device would turn on when the energy reaches its minimum operating voltage and discharge quickly without meaningful execution.

Energy Sources. Intermittent devices harvest three types of energy sources, (1) radiant (e.g., solar energy), (2) mechanical (e.g., wind and vibration), and (3) thermal (e.g., friction and temperature) [4].

First, intermittent devices harvest radiant energy for indoor and outdoor applications from light sources and radio frequency (RF) waves. Radiant energy harvesting devices are used in environmental monitoring and smart homes/buildings due to their high energy availability [19, 23, 26, 27, 94]. Second, they harvest mechanical energy from vibrations using piezoelectric devices. Piezoelectric harvesters are common for wearable devices and cyber-physical systems (e.g., industrial control systems and vehicles) [1, 18, 45, 48, 68]. Lastly, thermal harvesting leverages a temperature difference between two conducting materials. Thermal harvesting devices are common in industrial control systems and IoT environments with high-temperature devices (e.g., ovens) [28].

Energy Storage. The energy harvester in intermittent devices is not directly connected to the processing unit since its power output is lower than the device's operating voltage and current [4, 56]. Thus, intermittent devices use energy storage (e.g., capacitors) between the harvester and microcontroller to buffer energy over time. These energy storage units offer trade-offs in cost, lifetime, and capacity. Here, the energy capacity is critical for the device as it determines the amount of time the device needs to harvest energy before turning on and the device's operation time.

Table 1: Existing intermittent devices that periodically send sensor readings to a hub over a communication channel, their energy source, and potential information leakages.

Domain	Sensor Functionality	Energy Source	Comm.	Potential Information Leak
IoT/ICS	Vital sign sensing [47]	Heartbeat	2.4 GHz	User resting, walking or exercising
	Temp monitoring [45]	Vibration	2.4 GHz	Engine speed
	Air flow monitoring [54]	Air Flow	Zigbee	The status of AC
	Machine monitoring [1]	Vibration	BLE	Engine speed
	Machine monitoring [28]	Temp Differential	BLE	
	Temp/Pressure/Flow monitoring in pipes [33]	Water Flow	LoRa	Water usage in ICS or smart home
IoT	Temp/Hum monitoring [23]	Solar	BLE	Light status
	Contact sensor [27]	Solar	2.4 GHz	
	Inventory monitoring [94]	RF	BLE	Number of people in a home/store
	CO monitoring [19]	RF	BLE	
	Temp monitoring [26]	RF	900 MHz	
Wearable	Temp/Light/Sound monitoring [18, 66]	Vibration	2.4 GHz	User resting, walking or exercising
	Heart rate monitoring [43]	Vibration	BLE	
	Workout monitoring [68]	Vibration	BLE	
Medical	Infection detection [62]	Endocochlear Potential	2.4 GHz	Sound level
	Temp/pH/Pressure monitoring [65]	Galvanic cell	900 MHz	Dietary information

Definitions. We refer to intermittent devices as devices that rely only on harvested energy for operation instead of batteries. On the contrary, non-intermittent devices are traditional devices with batteries that do not harvest energy, such as laptops, smart TVs, IoT/ICS devices, and wearables with batteries. We use the term *active period* to refer to the difference between the two consecutive timestamps the intermittent device turns on. *Active period patterns* are sequences of active periods over time, which display differences based on the device's energy availability due to the time it takes for the device to accumulate enough energy to turn on.

3 MOTIVATION

Intermittent devices operate in bursts; they turn on when they have energy and turn off during energy scarcity. Our main observation is that *if an adversary can remotely learn when the intermittent device turns on and off, they can infer its energy availability by leveraging the device's active period as a side channel.* We refer to these attacks as *physical side-channel* attacks since they exploit the physical operation of the intermittent device in an environment.

The device's energy availability carries privacy-sensitive information about its operating environment. We present, in Table 1, a variety of intermittent devices, their use cases, communication channels, and the *potential* information leakages related to their energy availability. These devices all take sensor readings and periodically send them to a hub. For instance, a solar energy harvesting device measures the temperature and humidity of a smart home and reports them to the IoT hub over BLE [23]. From this device's energy availability, an adversary can infer the environment's light conditions and reason if the users are not at home or sleeping. As another

example, a heartbeat-harvesting device conducts vital sign sensing and sends the readings to a mobile phone over a 2.4 GHz radio [47]. From this device, an adversary can infer user activities (e.g., sleeping, exercising). Such information would allow an adversary to learn a user's routines and conduct physical attacks (e.g., burglary or kidnapping) or create user profiles for targeted ads [29, 31, 64].

In some cases, an adversary within the victim's visual range could observe the information leaked through intermittent devices. For instance, an adversary could peek through a window to see if the lights are on or use thermal cameras to observe user activities (e.g., sleeping, exercising). However, our attack removes the attacker's need to be in close proximity, allowing more stealthy attacks even against physically closed and protected environments (e.g., control rooms or offices) without directly observing the victim.

Determining the Side-Channel Leakage Source. In our initial attack prototype, we considered an adversary who leverages the intermittent device's electromagnetic (EM) emissions to learn its active period. This is because devices create higher EM emissions while they are turned on. Thus, the adversary can profile an intermittent device's active period through its EM emissions. Yet, capturing EM waves requires an adversary to deploy antennas within close proximity of the intermittent device (e.g., within millimeters) [16], which may be infeasible. Thus, for an adversary to remotely learn a device's active period, we leverage the timestamped network packet sequences transmitted from the intermittent device. Each transmitted packet indicates the device is on at that time. Thus, an adversary who sniffs the device's network packets can learn its active period through its inter-packet-arrival timings (the difference between the timestamps of two consecutive network packets observed by the adversary). We note that the intermittent device may not send a network packet in each active period. Yet, when it does send a packet, it means the device is on at that time.

3.1 Feasibility Study

We conduct a feasibility study to answer: *Do the network packets sent from an intermittent device indicate its energy availability?*

Experimental Setup. To answer this question, we conduct experiments in a real miniature manufacturing plant testbed in our laboratory, as shown in Figure 3. This testbed is a fully automated production factory [34] with four components. (1) Vacuum gripper robot (VGR) moves the workpieces. (2) High-bay warehouse (HBW) stores the workpieces. (3) Multi-processing station processes the workpieces. (4) Sorting line sorts workpieces based on their colors. The plant contains order and delivery phases. In the order phase, the workpieces are processed and placed in the warehouse. During delivery, the VGR places the workpiece at the delivery location after it is processed at the multi-processing station and sorting line.

Intermittent devices have started to be increasingly used in ICSs due to their ability to function in remote or inaccessible areas and their elimination of frequent battery replacements [87]. Following this, we connect a real vibration-harvesting intermittent device to the VGR's toothed gear. We use a Cypress intermittent device [23] with a MIDE S129 vibration harvester. Our intermittent device has similar specs to other devices in Table 1. This device transmits temperature and humidity readings to a hub over BLE every seven seconds (when it has enough energy).

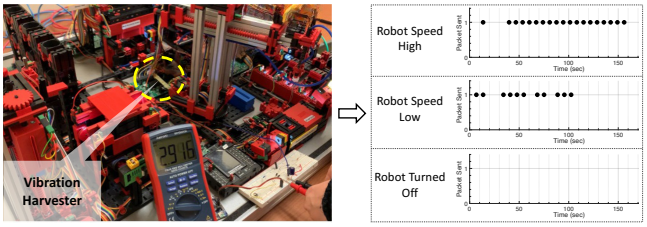


Figure 3: Feasibility experiment setup and results: We connect a vibration-harvesting intermittent device to a miniature manufacturing plant in our lab. The device's network packets indicate the vacuum gripper robot's speed.

Results. We run the manufacturing plant and collect network packet sequences from our intermittent device in three different conditions, (1) high VGR speed, (2) low VGR speed, and (3) VGR is stationary. Figure 3 shows the network packets received from the intermittent device in these conditions. The packet sequences show clear differences across different VGR conditions, where the intermittent device sends packets at a higher frequency when it has more available energy due to the vibration from VGR's high speed.

By inferring the VGR's speed, the adversary can timely conduct attacks against the manufacturing plant that would cause maximum damage. For instance, we consider the adversary aims to conduct a malicious command injection attack, causing the VGR to drop its workpiece. The adversary can leverage our physical side-channel attack to infer when the manufacturing plant operates at high speed and conduct the attack at that point to cause more severe damage to the physical machinery, environment, and workers.

These results motivate us to design a systematic side-channel attack to automatically infer an intermittent device's energy availability through temporal features extracted from its network packet sequences while accounting for the device's unique properties.

3.2 Threat Model

We consider an adversary who aims to infer privacy-sensitive information (e.g., user activity, robot states) from an intermittent device's energy availability. To this end, the adversary monitors the device's active period through the network packets it transmits.

We assume an adversary within the wireless communication range of intermittent devices. The adversary can install a sniffer once and manage it remotely to obtain the network traffic. The adversary does not intercept or inject messages but only records the traffic. This allows the adversary to remain undetected for an extended period of time. The adversary does not rely on the packet's content but only its metadata (i.e., MAC addresses and timestamps). Thus, our attack is still applicable if the device encrypts its packet content. We assume that the intermittent devices are installed within an indoor closed space (e.g., manufacturing plant, smart home, smart office). Therefore, the adversary cannot physically access or monitor the devices through other means (e.g., a camera) [30]. Lastly, similar to prior network analysis attacks against traditional IoT devices for fingerprinting them [3, 5, 8, 11, 93], we assume the adversary has access to a set of intermittent devices to collect data and train their energy inference algorithm.

3.3 Challenges

C1: Intermittent Device Identification. In real-world use cases, intermittent devices usually coexist with non-intermittent devices with batteries. Thus, the adversary must first identify the victim intermittent device(s) to conduct physical side-channel attacks.

To address this challenge, one approach would be simply identifying the victim intermittent devices by recognizing their MAC addresses since the top three bytes of a MAC address indicate the device manufacturer. Unfortunately, recognizing the manufacturer is not enough since each manufacturer usually produces both intermittent and non-intermittent devices. Another approach would be leveraging IoT device fingerprinting techniques [3, 5, 8, 11, 93]. Yet, these techniques do not consider properties unique to intermittent devices (e.g., firmware configurations, execution models, and energy storage mechanisms). Thus, they fail to fingerprint intermittent devices accurately and identify them.

C2: Energy Harvester Recognition. To infer privacy-sensitive information from the victim intermittent device, the adversary needs to infer its energy harvester. Yet, there are diverse energy harvesters that intermittent devices might be equipped with, and many devices allow users to connect different harvesters.

Recognizing a device's harvester is challenging because different harvesters yield similar active periods when their energy conditions are similar, e.g., when a solar harvester is exposed to high illumination and a temperature harvester is exposed to high temperature.

C3: Accounting for Unique Intermittent Device Properties. There are various properties unique to each intermittent device, such as their voltage thresholds, packet transmission intervals, and energy storage size. Depending on their use cases, these properties might be configured differently by each vendor at production and by users at deployment. For instance, a user may connect an additional capacitor to increase the device's energy storage. Such diverse properties impact the device's active period and inter-packet-arrival timings. Thus, these properties must be integrated into the adversary's energy inference algorithm to accurately learn the device's energy availability and desired privacy-sensitive information.

A possible solution is conducting extensive experiments with the device to model its active period. Yet, experiments with all combinations of device properties, harvesters, and energy availability conditions are infeasible. To address this, one could use a simulator. Yet, existing simulators do not precisely simulate the active periods of intermittent devices with networking capabilities [35, 36, 90].

4 SYSTEM DESIGN

4.1 Overview

Figure 4 shows our approach to conducting side-channel attacks against intermittent devices. Our attack has two phases, (1) offline intermittent device analysis and (2) online hierarchical inference.

In the offline phase (❶), first, the device property analyzer identifies the configurable intermittent device properties that are causally related to the device's active period (**C3**). Second, grid testing runs the device with different combinations of the identified device properties in various environments while the device is connected to different harvesters to collect a comprehensive timestamped packet sequence dataset. Lastly, given the collected dataset, we model the

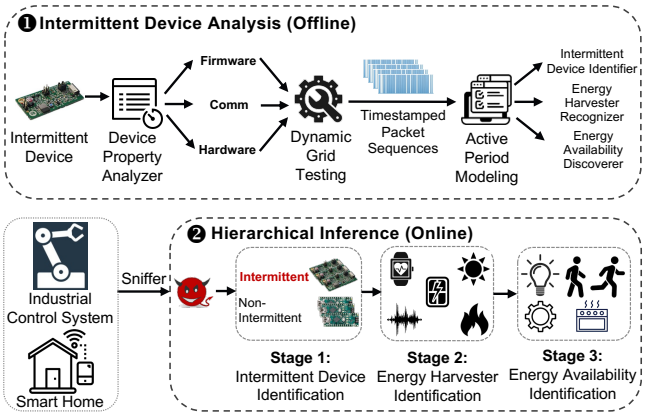


Figure 4: Overview of our attack.

device's active period patterns through its inter-packet-arrival timings with changing device properties, energy harvesters, and energy availability. This module outputs a hierarchical inference framework consisting of an intermittent device identifier (**C1**), energy harvester recognizer (**C2**), and energy availability discoverer.

In the online phase (❷), we first use a network sniffer to collect all packets in the victim environment. We then separate the timestamped packet sequences from each device through their unique identifiers (e.g., MAC addresses) and conduct a three-stage hierarchical inference using the models derived offline. We first identify if one or more intermittent devices exist in the environment ($\texttt{Stage}_1$). If an intermittent device exists, we infer which energy harvester is connected to it ($\texttt{Stage}_2$). Lastly, we continuously infer the intermittent device's energy availability over time to learn privacy-sensitive information related to its energy availability ($\texttt{Stage}_3$).

4.2 Intermittent Device Analysis

4.2.1 Device Property Analysis. Given an intermittent device, we first identify the properties that can be configured by users at deployment and are causally related to the device's active period. To model an intermittent device's active period, the adversary must conduct data collection with the device in a surrogate testbed or lab environment (since we do not assume the adversary has access to the physical environment in which the device is deployed) while accounting for its properties. Yet, intermittent devices have many properties; thus, conducting extensive data collection with all properties incurs a high time overhead. To address this, we eliminate the properties that are not causally related to the device's active period and enable scalable data collection for active period modeling.

To this end, we first analyze the intermittent device's user manuals to identify its configurable properties. We next conduct controlled experiments by individually changing each device property and collecting its timestamped packet sequences. We leverage dynamic time warping [13, 17] to measure the active period differences when a device property changes. Lastly, we eliminate the properties that do not cause a change in the device's active period.

Identifying Configurable Device Properties. Through our survey of intermittent devices, we found eight common properties in three categories that may be causally related to their active periods.

Hardware: The intermittent device's microcontroller type and sensors on board may impact the device's energy consumption. The device's energy storage size also affects its active period since it determines how long the device stays off before turning on and how long the device operates before turning off.

Firmware: We have identified three firmware parameters that may impact the intermittent device's active period, (1) packet size, (2) packet transmission interval (i.e., the duration a device waits between attempting to send two consecutive packets), and (3) voltage thresholds for changing the device state (i.e., on/off).

Communication: The intermittent device's radio module and communication protocol (e.g., BLE and Zigbee) may impact the device's active period due to their different energy consumptions.

Although several properties may impact the device's active period, not all are configurable by the user at the application level. If a property is not configurable, the adversary only needs to consider the impact of the device's default configuration on its active period. Since intermittent devices are diverse, the configurable properties also vary among them. For instance, a carbon-monoxide monitoring intermittent device [19] does not allow changing any properties. On the contrary, an environment monitoring device [83] has open-source firmware and enables changing several properties, such as the packet transmission interval. Thus, we manually study the devices' user guides to identify the configurable properties.

Controlled Experiments with Different Intermittent Device Properties. We modify each configurable property on the real intermittent device and conduct experiments to observe the changes in its active period patterns through its inter-packet-arrival timings.

The intermittent devices' user guides and manuals give hints on how to change the configurable device properties. For instance, an intermittent device's user manual mentions `app_config.h` file in the device firmware containing the configurable properties [83]. The following code block shows this device's two configurable properties, packet transmission interval and voltage threshold.

```
// Advertising Interval [ms] <100-100000>
#define APP_ADV_INTERVAL_MS   2000
// Capacitor Voltage Threshold [mV]
#define APP_VBAT_THRESHOLD    2500
```

After identifying the methods to change device properties, we conduct experiments with the intermittent device's default configurations and with each property changed individually to collect timestamped packet sequences. In these experiments, we do not change the device's harvester and keep stable energy conditions to observe only the property's impact on the device's active period.

Causal Structure Identification. We identify the causal structure between intermittent device properties and the device's active period by computing the active period distance ($\mathsf{d_{ap}}$) between the timestamped packet sequence collected with the device's default configurations and the sequence collected with a changed property. We eliminate the properties with $\mathsf{d_{ap}}$ smaller than a threshold as they are not causally related to the device's active period.

We leverage dynamic time warping [13, 17] as our $\mathsf{d_{ap}}$ metric. Compared to other metrics, such as Euclidean and Manhattan distances, dynamic time warping enables many-to-one comparisons between timestamped packet sequences, enabling the detection of shifts and shapes in the sequences. Thus, given two packet sequences, one with the default configuration ($\mathsf{D} = \{\mathsf{d_1}, \mathsf{d_2}, \ldots, \mathsf{d_n}\}$)

Algorithm 1 Dynamic Grid-Testing

Input: Intermittent device ($\mathcal{D}$), A set of device properties ($\mathsf{L_p}$), A set of energy harvesters ($\mathsf{L_h}$), A set of energy availability conditions ($\mathsf{L_c}$).
Output: Intermittent device timestamped packet sequences dataset (DS).

```
1: function DYNAMIC_GRID_TEST(D, L_p, L_h, L_c)
2:     for L ∈ L_p do
3:         for h ∈ L_h, c ∈ L_c do
4:             DS ← DS ∪ {L, h, c, D(L, h, c)}
5:         end for
6:     end for
7:     return DS
8: end function
```

and the other with the changed configuration ($\mathsf{C} = \{\mathsf{c_1}, \mathsf{c_2}, \ldots, \mathsf{c_n}\}$), we first compute a distance matrix as:

$$\begin{bmatrix} \mathtt{dist}(\mathsf{d_1},\mathsf{c_1}) & \mathtt{dist}(\mathsf{d_1},\mathsf{c_2}) & \ldots & \mathtt{dist}(\mathsf{d_1},\mathsf{c_n}) \\ \mathtt{dist}(\mathsf{d_2},\mathsf{c_1}) & \mathtt{dist}(\mathsf{d_2},\mathsf{c_2}) & & \\ \vdots & & \ddots & \\ \mathtt{dist}(\mathsf{d_n},\mathsf{c_1}) & & & \mathtt{dist}(\mathsf{d_n},\mathsf{c_n}) \end{bmatrix}$$

Here, `dist` is a standard distance metric such as Euclidean. $\mathsf{d_{ap}}$ is then defined as the minimum sum of the contiguous elements in this matrix. We compute $\mathsf{d_{ap}}$ for each intermittent device property that can be configured by the user at deployment. This module outputs the set of properties ($\mathsf{L_p}$) with $\mathsf{d_{ap}}$ larger than a threshold, as these properties impact the device's active period the most. This threshold provides a tradeoff between scalability and accuracy. A lower threshold causes more properties to be considered for active period modeling, enabling more accurate attacks but also increasing the testing overhead. We set this threshold by finding the value that provides reasonable attack accuracy and scalability in Section 5.

4.2.2 Dynamic Grid-Testing. Dynamic grid-testing takes, as input, the list of configurable intermittent device properties that are causally related to the device's active period and the list of energy harvesters and availability conditions the adversary aims to infer. It then runs the intermittent device with combinations of these inputs to collect a comprehensive timestamped packet sequence dataset from the device. This dataset enables accurate profiling of the intermittent device's active period. Algorithm 1 shows the steps of conducting dynamic grid-testing on intermittent devices.

The algorithm first generates a set of test cases by considering all combinations of device property configurations (Line 2). However, testing the device with all combinations of configurations with numerical values (e.g., packet transmission interval) is impractical. To address this, we discretize such properties by setting their configurations within their min/max ranges with equal intervals. For each energy harvester and configuration, we set the energy availability conditions to the ones we aim to infer and run the device (Lines 3-4). We collect network traffic from the intermittent device as a timestamped packet sequence while it runs for a given amount of time. We repeat this for all configurations, harvesters, and energy conditions to generate a comprehensive dataset.

4.2.3 Active Period Modeling. We model the intermittent device's active period using the timestamped packet sequence dataset generated through Algorithm 1. We develop a hierarchical classification framework that (1) distinguishes a victim intermittent device from non-intermittent devices, (2) recognizes the device's energy harvester, and (3) identifies its energy availability condition.

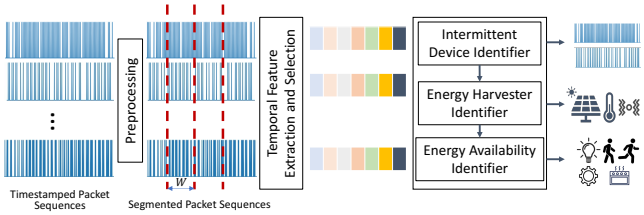


Figure 5: Active period modeling architecture.

We derive a separate model for each hierarchical stage due to two reasons. First, the intermittent device's active period has different intra-execution and inter-execution patterns. *Intra-execution patterns* are the changes in the device's active period within a short period of time, based on the amount of energy available to the device (i.e., the device's active period while it has available energy). *Inter-execution patterns* are the active period changes over an extended period of time, representing the time when the device does not have energy. While intra-execution patterns are effective in identifying the victim device and its energy availability, as shown by our experiments in Section 5, they are not suitable for recognizing the harvester. Thus, we leverage inter-execution patterns to identify the harvester. Second, to realize our attack in practice, we sequentially use each model to first eliminate the packets from non-intermittent devices and identify the intermittent device's harvester to infer privacy-sensitive information related to its energy availability.

Here, we also require a timestamped packet sequence dataset from non-intermittent devices to train our intermittent device identification model. This dataset can be collected from any smart home and industrial control system, or existing datasets [5] can be used. Figure 5 shows the general pipeline for deriving a model for each stage of our hierarchical classification framework, as detailed below.

Window-based Temporal Feature Extraction and Selection. Given the timestamped packet sequences, we first segment them into multiple samples with window size ($\mathtt{W}$). We next compute their inter-packet-arrival timings and extract time-domain features ($\mathtt{F}$) (e.g., min, max, median, length, and variance) from each sample. These features allow us to capture both intra-execution and inter-execution patterns in the inter-packet-arrival timings.

We next leverage relative mutual information (RMI) [50, 82] to measure each feature's distinctiveness in distinguishing between classes for each stage of our hierarchical model. We define RMI as:

$$\mathtt{RMI}(\mathtt{F},\mathtt{Y}) = \mathtt{H}(\mathtt{F}) + \mathtt{H}(\mathtt{Y}) - \mathtt{H}(\mathtt{F} \mid \mathtt{Y})$$

$\mathtt{Y}$ is the sequence's ground-truth label, and $\mathtt{H}(\mathtt{F})$ and $\mathtt{H}(\mathtt{Y})$ are the entropy of the features and ground-truth [50]. For each model, we separately select the top $\mathtt{k}$ features with the highest RMI for training.

Here, the window size parameter plays a crucial role in feature extraction and selection to model the device's active period. The window size should be large enough to capture the device's intra-execution and inter-execution patterns but small enough for the adversary to continuously infer the device's energy availability conditions. Thus, we determine it by conducting a grid search and selecting the optimal value that provides the highest cross-validation accuracy for each model separately.

Active Period Inference Model. Our hierarchical inference models leverage random forest classifiers [14] to identify the intermittent device, its energy harvester, and energy availability. They learn the device's active period patterns from an ensemble of decision trees, which accounts for the variance in inter-packet-arrival timings for various device properties and harvesters. Leveraging random forest for inference allows the adversary to infer the energy availability of devices with high accuracy without extensive data collection, as in the case of deep-learning-based approaches. We train our models using timestamped network packet sequences generated with our dynamic grid testing (Section 4.2.2). We perform cross-validation to tune the model parameters and determine the optimal window sizes for feature extraction and selection.

Algorithm 2 Hierarchical Inference

Input: Stream of timestamped packet sequences ($\mathcal{T}$), Intermittent device identifier ($\mathcal{M}_\mathtt{I}$), Energy harvester recognizer ($\mathcal{M}_\mathtt{H}$), Energy availability discoverer ($\mathcal{M}_\mathtt{E}$), time windows ($\mathtt{W_I}$, $\mathtt{W_H}$, $\mathtt{W_E}$).
Output: Energy availability conditions ($\mathtt{c}$).

```
1:  function HIERARCHICAL_INFERENCE(T, M_I, M_H, M_E, W_I, W_H, W_E)
2:      IntDevices = [], IntHarvesters = []
3:      test ← FILTER(T)                          ▷ Split the traffic to different devices
4:      for i ∈ test do
5:          ts ← WINDOW(T, W_I)
6:          if M_I(ts) == Intermittent then
7:              IntDevices = IntDevices ∪ i
8:          end if
9:      end for
10:     for j ∈ IntDevices do
11:         testH ← WINDOW(T, W_H)
12:         IntHarvesters = IntHarvesters ∪ M_H(testH)
13:     end for
14:     for k ∈ IntHarvesters do
15:         while T ≠ ∅ do
16:             testE ← WINDOW(T, W_E)
17:             return k, c ← M_E(testE)
18:         end while
19:     end for
20: end function
```

4.3 Hierarchical Inference

At the online stage of our attack, we first use a network sniffer (e.g., BLE, Zigbee) to remotely collect timestamped packet sequences. We then separate the timestamped packet sequences from each device through their unique identifiers, such as MAC addresses. We next leverage our active period inference models for hierarchical inference, as presented in Algorithm 2. The algorithm takes, as input, a stream of timestamped packet sequences from the sniffer, the hierarchical inference models, and time windows. It then outputs the energy availability conditions of the intermittent devices.

Victim Intermittent Device Identification. The algorithm starts with splitting the timestamped packet sequences to different devices based on their unique identifiers (e.g., MAC addresses) (Line 3). One may consider using such unique identifiers to determine the victim intermittent device. Yet, as discussed in Section 3.3, these identifiers enable the adversary to only recognize a device's manufacturer. As manufacturers usually produce both intermittent and non-intermittent devices, a matching manufacturer does not imply identifying an intermittent device. Thus, we leverage our intermittent device identifier model to distinguish intermittent devices.

For each device, our algorithm extracts a window of timestamped packet sequence of size $\mathtt{W_I}$ (Line 4-5). The intermittent device identifier then extracts a set of features from the traffic window, classifies the device as intermittent or non-intermittent (Line 6), and outputs the set of identified victim intermittent devices (Line 7).

Energy Harvester Identification. The algorithm next uses the energy harvester recognizer to infer the energy harvesters of the victim intermittent devices. For each intermittent device, it generates a window of timestamped packet sequences with the W_H parameter (Line 11). Similarly, it extracts a set of features from the timestamped packet sequences and assigns an energy harvester for each device (Line 12). This stage outputs the set of victim intermittent devices along with their energy sources.

Energy Availability Identification. The algorithm next uses energy availability discoverers to infer the energy conditions of the device, which carries privacy-sensitive information as detailed in Section 3. For each identified victim intermittent device, the algorithm divides the stream of timestamped packet sequences into windows of size W_E (Line 15-16). It next uses the energy availability discoverer derived for that harvester to extract features and infer the energy availability of the device. The algorithm continuously infers and outputs the energy availability conditions of the identified victim intermittent devices over time (Line 17).

5 EVALUATION

We evaluate our physical side-channel attacks with three energy harvesters (solar, vibration, and temperature) connected to real intermittent devices in three different real-life scenarios.

In the first scenario, we consider a smart home where the adversary aims to infer the light conditions from a solar-harvesting intermittent device, which indicates whether the users are at home or sleeping (e.g., to conduct physical attacks such as burglary or kidnapping). In the second scenario, a vibration-harvesting device is deployed in a real miniature manufacturing plant, where the adversary aims to infer the operation of a robot (e.g., to conduct sensor spoofing attacks). In the third scenario, a temperature-harvesting device is connected outside of an oven, where the adversary aims to infer whether the oven is turned on (e.g., to learn the cooking activity patterns of the residents or exploit the unintended physical app interactions the turned-on oven may trigger [70]).

Our experiments show that an adversary can accurately recover privacy-sensitive information from intermittent devices. We present our results by focusing on several research questions:

- **RQ1** Which intermittent device properties are causally related to the device's active period? (Section 5.2)
- **RQ2** What is the accuracy in each stage of our hierarchical inference? (Section 5.3)
- **RQ3** How does the accuracy change with different amounts of training data and time windows? (Section 5.3.4)
- **RQ4** How does the accuracy change if the adversary excludes an intermittent device property in grid testing? (Section 5.4)
- **RQ5** What is the accuracy of our attack against existing network analysis defenses? (Section 5.5)

5.1 Evaluation Setup

5.1.1 Intermittent Device. We use a Cypress device [23] equipped with a 32-bit 48 MHz CYBLE processor with 128 KB flash memory and 16 KB SRAM, an S6AE101A power management integrated circuit, and a BLE radio. It takes temperature and humidity readings every seven seconds and sends them to a hub over BLE. It has three 100 μF capacitors for energy storage and supports an additional capacitor. It includes built-in open-source firmware, which we do not change except its configurable parameters.

We have selected this device for three reasons: (1) It is commercially available. (2) It supports different energy harvesters, allowing us to validate our attacks with them. (3) Its specs are similar to other intermittent devices [1, 26, 27, 68, 94], providing a general idea of our attack's effectiveness on different devices. We also conduct additional experiments to confirm the transferability of our attack to a different intermittent device [83] in Section 6.

5.1.2 Energy Harvesters. We connect three harvesters, a Panasonic solar harvester, a MIDE vibration harvester, and an EverGen temperature harvester, to the intermittent devices. These harvesters cover all three energy source types outlined in Section 2. We select these harvesters as they are increasingly used in smart grids [9], wearables [86], wireless sensor networks [22, 24], and ICS [87].

5.1.3 Data Collection. We collect network packets from the intermittent devices in three datasets: (DS_1) Intermittent device analysis dataset, (DS_2) Active period modeling dataset generated through dynamic grid testing, and (DS_3) Testing dataset to measure the accuracy of our attacks in real-world scenarios. We collect DS_1 and DS_2 from a lab environment, whereas we collect DS_3 from real environments that include various noise factors, such as different devices that operate in the same frequency band as the intermittent devices. This is because, in practice, the adversary may not be able to collect packet sequences from intermittent devices in their operating environments for training active period models. Therefore, we consider an adversary who conducts our attack's offline stages in a lab environment and uses the derived models to attack intermittent devices in practice. We note that this is a stronger attack model (less advantageous for the adversary) as we do not assume physical access to the devices' operating environment.

Dataset-1. For DS_1, we identify three configurable intermittent device properties from the device's user guide, (1) including a 220 μF additional capacitor, (2) the packet transmission interval, and (3) low-voltage detection threshold. Thus, we change these device properties and collect 10 mins of network packets from the device while it is equipped with the solar harvester and under ideal energy conditions. We select the solar harvester as it enables testing the most number of properties (e.g., supports operating without an additional capacitor, while it is required for other harvesters). We use DS_1 to identify which device properties impact the inter-packet-arrival timings that guide our testing configurations for DS_2.

Dataset-2. For DS_2, we collect packets from the intermittent device with all three harvesters in different energy availability conditions and changing intermittent device properties (See Appendix Figure 10 for temperature and solar harvesting setups). We collect 30 mins of network packets with each configuration. To create different energy availability conditions, we create surrogate testbeds since we assume the adversary does not have access to the device's real operating environment. With the solar harvester, we collect packets in high ($\approx$ 500 lux), medium ($\approx$ 200 lux), low ($\approx$ 50 lux), and no illuminance settings. For the vibration harvester, we use a toothed gear connected to a DC motor. We collect packets when the toothed gear operates for 15 secs with 30 secs intervals and 25

Table 2: Real-world scenarios in our evaluation.

Exp ID	Energy Harvester	Scenario	Energy Conditions	Information Leakage
Exp_1	Solar	Smart Home	Light Off Light Low Light Med Light High	Occupants are not at home or sleeping
Exp_2	Vibration	Manufacturing Plant	No Operation Mid Operation High Operation	Workpiece-carrier's status
Exp_3	Temperature	Oven	Oven Off Oven Med Oven High	Oven's status

secs with 30 secs intervals, mimicking different operations of the manufacturing plant [34]. Lastly, with the temperature harvester, we use a hot plate and set its temperature to 40°C, 50°C, and 60°C.

Dataset-3. DS_3 represents our testing dataset, where the adversary uses the active period models learned with DS_2 to infer privacy-sensitive information in real-world environments. Table 2 shows the three real-world scenarios considered in our experiments[2].

In the first scenario, we collect 1.5 hours of timestamped packet sequences while the intermittent device is connected to the solar harvester in a smart home where the light conditions change. The adversary aims to infer the light level, indicating whether the users are at home or sleeping. The smart home also includes other devices (e.g., smart TV, laptop, phone), and therefore, the dataset includes packets from 159 unique MAC addresses.

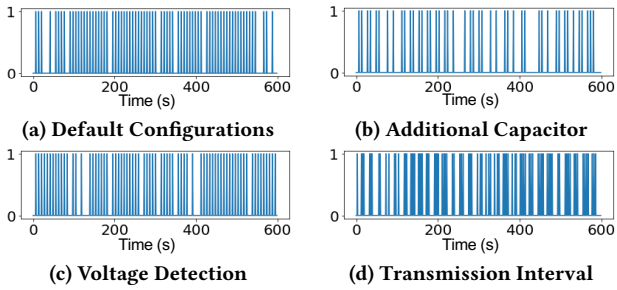
In the second scenario, we connect the vibration harvester to our manufacturing plant testbed (Detailed in Section 3.1) and collect network packets while the plant runs for 2 hours (See Appendix Figure 11 for the setup). We connect the harvester to a robot in the high-bay warehouse. This robot controls the position where a workpiece is stored, and its movements depend on the storage location. Here, the adversary can infer when the plant is operating and what the workpiece position is in the warehouse.

In the last scenario, we connect the temperature harvester to the outside of an electric oven and collect packets for 1 hour. Here, the adversary aims to infer the oven's status, if it is turned on or off.

All of our testing environments allow us to practically demonstrate our attacks since they naturally include various noises that can occur in any real-life environment, e.g., diverse physical obstacles and vibrations from the other components of the manufacturing plant. They also include noise on the communication channels since there are different devices in these environments that operate in the same frequency band as the intermittent devices. Therefore, the inter-packet-arrival timings collected from these environments are impacted by various communication factors, including channel conditions, interference, and communication quality.

5.2 Intermittent Device Analysis Results

We present the results of our intermittent device analysis performed on the DS_1 dataset. We compute the active period distance (d_{ap}) between the time-series packets collected with default device configurations and with a changed device property to determine the properties causally related to the device's active period. We found that two configurable intermittent device properties, including an additional capacitor and the packet transmission interval, significantly impact the device's active period. In contrast, the low-voltage detection threshold minimally influences it.



(a) Default Configurations (b) Additional Capacitor

(c) Voltage Detection (d) Transmission Interval

Figure 6: Time-series network packets transmitted from the intermittent device with different configurations (1 indicates packet received, 0 indicates not received).

Figure 6a presents the network packets received from the intermittent device with default configurations (no additional energy storage, disabled low-voltage interrupt, and 7 secs packet transmission intervals). When we connect an additional capacitor to the device, we observe more extended periods of time between packets (Figure 6b). The d_{ap} between the timestamped packet sequences with default configurations and with an additional capacitor is 5.20. This large distance is because the additional capacitor takes extra time to charge and incurs energy loss. We also modify the disabled low-voltage detection to generate an interrupt when the voltage is 1.75 V. Figure 6c shows that the inter-packet-arrival timings with the changed voltage detection are similar to the default configuration's timings ($d_{ap} = 0$). Lastly, we change the packet transmission interval to 4 secs instead of the default 7 secs. We observe that it causes packets to be sent in shorter bursts with more extended periods of time in between, with a d_{ap} of 4.24 (Figure 6d).

Based on these experiments, in our dynamic grid testing, we only consider energy storage and packet transmission interval changes but do not change the low-voltage detection as it does not impact the device's inter-packet-arrival timings. This decreases the adversary's data collection for active period modeling since data collection for different low-voltage detection configurations is not required.

5.3 Attack Effectiveness

We present our attack's performance in intermittent device, energy harvester, and energy availability identification for the three experimental scenarios (Exp_1, Exp_2, Exp_3). We train each model in our hierarchical framework using the active period modeling dataset (DS_2) and find each model's optimal parameters (e.g., time windows) through grid search and cross-validation. We use the models on the test dataset (DS_3) collected from real environments that include various noise factors (e.g., other devices sending packets in the same frequency band) and measure our attack's effectiveness. Table 3 shows our attack's effectiveness in each stage. Our results show that we can distinguish intermittent devices from others with 98% accuracy, identify the intermittent device's harvester with 92% accuracy, and infer its energy availability with 85% accuracy on average. We compare the performance of our attack with other classifiers in Appendix Table 7.

[2]We contacted the IRB office at our university, and they advised that IRB approval is not required because we do not collect sensitive information from human subjects.

Table 3: Effectiveness of our attack.

Experiment	Accuracy	Precision	Recall
Stage1: Intermittent Device Identification	98%	99%	98%
Stage2: Energy Harvester Identification	92%	94%	92%
Exp_1: Solar Energy Availability	80%	81%	80%
Exp_2: Vibration Energy Availability	88%	89%	88%
Exp_3: Temp Energy Availability	87%	87%	87%

5.3.1 Intermittent Device Identification. The results show that the adversary can distinguish the packets from the victim intermittent devices with 98% accuracy, 99% precision, and 98% recall when the time window parameter (W_I - used for splitting the packet sequences into windows) is set to 2 mins. We found that although the adversary correctly identifies all intermittent devices, a small subset ($\approx$ 3%) of non-intermittent devices is incorrectly classified due to their packet sequences being similar to intermittent devices.

We observe that intermittent devices have longer inter-packet-arrival timings compared to non-intermittent devices, enabling our framework to distinguish them. This is because they must accumulate energy after sending a packet to send the next one. This energy requirement prevents intermittent devices from sending packets with high frequency. This difference is reflected in our feature selection, where the features with the highest RMI were Min and Median. The less frequent packet transmissions cause a higher minimum and median inter-packet-arrival timing for intermittent devices.

Public Dataset Experiments. We conduct an additional experiment to show our attack's generalizability and ability to distinguish intermittent devices from non-intermittent IoT devices that may have similar inter-packet arrival timing patterns with intermittent devices. To this end, we leverage two public datasets collected in environments similar to our setups (e.g., smart home) for fingerprinting and classifying non-intermittent IoT devices [5, 89]. We combine our dataset collected from intermittent devices with these datasets and run our attack's intermittent device identification. We split the datasets as 80% and 20% for training and testing.

Table 4 presents the description of these two datasets and the accuracy, precision, and recall of our attack's intermittent device identification. Our attack has 100% accuracy with the NCSU dataset [5] and 96% accuracy with the UNSW dataset [89]. We found that similar to the experiments with our non-intermittent device dataset, Min and Median inter-packet-arrival timings have the highest RMI.

5.3.2 Energy Harvester Identification. Our test dataset (DS_3) includes timestamped packet sequences from the intermittent device when it is connected to solar, temperature, and vibration harvesters. We show that an adversary can identify the device's harvester (between solar, temperature, and vibration) with 92% accuracy, 94% precision, and 92% recall with 45 mins time windows (W_H). The confusion matrix illustrated in Figure 7a shows that the only misclassifications occur between solar and vibration harvesters.

We set W_H to 45 mins as it enables our framework to capture the inter-execution patterns between harvesters, yielding high accuracy. For instance, with the vibration harvester, the device sends packets in short bursts with long wait times in between due to the manufacturing plant's operation. In contrast, with the solar harvester, the device continuously sends packets over a more extended period of time since light-condition changes are less frequent. Thus,

Table 4: Our attack's effectiveness in identifying intermittent devices from non-intermittent IoT devices.

Dataset	Year	No of Packets	No of IoT Devices	Accuracy	Precision	Recall
NCSU	2022	>100K	11	100%	100%	100%
UNSW	2018	>600K	22	96%	99%	96%

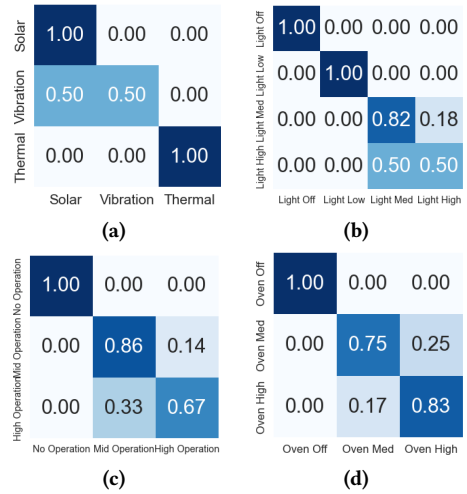


Figure 7: Confusion matrices for (a) energy harvester, (b) solar energy availability, (c) vibration energy availability, and (d) temperature energy availability identification. The x-axes indicate the predicted classes, and the y-axes indicate the ground truth.

the two features that better distinguish the energy harvesters are the max inter-packet-arrival timing and the sum of their values. The max value indicates the inter-execution patterns as it shows the amount of time when the device did not have enough energy. The sum value is also highly impacted by the inter-execution patterns since a large period of energy unavailability causes lower values.

We evaluate the effectiveness of our selected time window by comparing its performance to the harvester identification performance with a window of 2 mins. The accuracy drops to $\approx$ 70% in this case. This is because the model cannot capture the inter-execution patterns within 2 mins. Thus, it cannot distinguish between similar energy conditions from different harvesters (e.g., light-high for solar and oven-high for temperature harvester).

5.3.3 Energy Availability Identification. We measure our attack's effectiveness in inferring the energy conditions of each of the three harvesters. On average, our attack achieves 85% accuracy, 86% precision, and 85% recall with 2 mins time windows (W_E).

Exp_1. Our attack achieves 80% accuracy in identifying the smart home's light level. Through this, the adversary infers when the user is sleeping or is not at home to conduct physical attacks (e.g., kidnapping or burglary). The Median and Min features have the highest RMI in distinguishing light levels since they indicate the device's intra-execution patterns within a short time window. The confusion matrix shows that the misclassifications are mainly because light-high and light-med conditions cause similar inter-packet-arrival timing patterns (Figure 7b). To confirm this, we performed an additional test with two classes, light-on (encapsulating

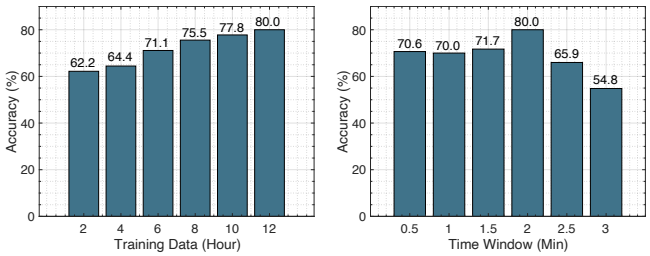


Figure 8: Attack accuracy with changing (a) amount of training data and (b) segmentation time windows (W_E).

light-high/med/low) and light-off, and found the adversary achieves 100% accuracy. This indicates a trade-off between attack accuracy and granularity, where identifying energy conditions at a finer granularity results in a decrease in accuracy.

Exp_2. Our attack achieves 88% accuracy in inferring the robot's state in the warehouse, where Figure 7c shows the confusion matrix. The intermittent device's energy availability changes based on the position in which the warehouse stores the workpiece, as the robot's movements depend on this position. This causes differences in the device's packet transmissions, where the `Length` feature, indicating the number of packets within the time window, provides the highest RMI. Through this attack, the adversary learns (1) the time that the robot is carrying an item and (2) the location where the workpieces are stored. The adversary can use this information to timely conduct a sensor spoofing or parameter injection attack [32, 71, 95], maximizing the damage to the plant and the workpieces.

Exp_3. Our attack achieves 87% accuracy in identifying if the oven is operating at a high temperature (≈ 230-250$^{\circ}C$), medium temperature (≈ 200$^{\circ}C$), or it is off. Figure 7d shows that there are misclassifications between high and med temperature settings since they sometimes cause a similar frequency in the packet sequences. Generally, with a high oven temperature, the device accumulates more energy and transmits packets at a higher frequency. We found that `Length` and `Mean` features were the most effective in distinguishing different temperature levels because a high oven temperature causes a higher number of transmitted packets and lower mean of inter-packet-arrival timings. With this attack, the adversary infers the time a person is cooking in their home. Thus, the adversary can learn the activity patterns of the residents.

5.3.4 Attack Parameters. We evaluate the impact of the two attack parameters: (a) the training data size for learning active period models and (b) the energy availability inference time window (W_E). We conduct this evaluation on the solar energy availability identification, where we learn active period models with different parameters and test on our smart home dataset (Exp_1). We vary the training data size by using 2 to 12 hours of packet sequences and change the time window sizes between 30 secs to 3 mins. Figure 8 presents our results, which show that (1) increasing training data yields higher accuracy, and (2) 2 mins time window gives the best performance, supporting our parameter selection with cross-validated results.

Accuracy with Different Amount of Training Data. Figure 8a shows the attack accuracy with different training data sizes. To vary the training data size, we randomly select subsets of DS_2. As expected, the accuracy increases with increasing training data. Yet, the accuracy becomes stable when 8 or more hours of data are used for training, which is always higher than 75%.

Accuracy with Different Time Windows. Figure 8b shows the attack accuracy with different time windows (W_E) that represent the size of the sequences we segment the packets into. The results show that 2 mins windows give the highest accuracy (80%), whereas smaller time windows give ≈ 70% accuracy. We observe that the accuracy drops when larger time windows (e.g., 3 mins) are used. This decrease occurs because the device's packet transmission interval (7 secs) becomes much smaller than the time window, making the inter-packet-arrival timing patterns less distinguishable.

Table 5: Ablation study on intermittent device properties.

	Extra Storage		Transmission Interval			
ID†	No Extra Capacitor	Extra Capacitor	7 secs	4 secs	1 sec	Accuracy
AB_1	✓	✗	✓	✓	✓	75%
AB_2	✗	✓	✓	✓	✓	71%
AB_3	✓	✓	✓	✗	✗	62%
AB_4	✓	✓	✗	✓	✗	60%
All	✓	✓	✓	✓	✓	80%

† AB_x represents the ablation settings we considered (e.g., In AB_1, we train our inference model by excluding the data collected when an extra capacitor is connected to the device). All represents that all intermittent device properties are considered in training.

5.4 Ablation Study on Device Properties

To understand the impact of device properties on our attack's accuracy, we perform an ablation study by training the energy inference model without including the data collected with certain device properties. We perform this study on the solar harvester (Exp_1), allowing us to measure the impact of adding an additional capacitor to the device and different packet transmission intervals. Table 5 shows our attack's accuracy when we train the energy inference model with datasets that include different sets of device properties.

Additional Capacitor. We exclude the additional capacitor data (AB_1) and no-additional capacitor data (AB_2) from training (DS_2) while learning the models. We test them on the Exp_1 testing dataset.

Table 5 shows excluding the additional capacitor data causes a decrease in the attack accuracy by 5%, and excluding the no-additional capacitor data causes a 9% decrease. The accuracy difference between the two ablation studies is because the testing data is collected with the default configurations of our intermittent device, which does not have an additional capacitor. Including both capacitor settings enables the highest accuracy because it allows the adversary to extract inter-packet-arrival timing patterns that are applicable regardless of the device having an additional capacitor.

Packet Transmission Interval. To evaluate the importance of changing the packet transmission intervals, we use the active period modeling data with 7 secs interval (AB_3) and 4 secs interval (AB_4).

We found that excluding different packet transmission intervals from the dataset causes a higher decrease in attack accuracy. It decreases by 18% when only 7 secs packet transmission interval is considered, and 20% with 4 secs interval. This is because the features that distinguish the light levels are those that are more impacted by the packet transmission interval configuration (e.g., min, median). Thus, including a more extensive set of packet transmission intervals significantly improves our attack's performance.

Table 6: Effectiveness of our attack under traffic injection and traffic reshaping defenses.

Experiment	No-Defense			Injection			Reshaping		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
Intermittent Device Identification	98%	99%	98%	97%	98%	97%	98%	99%	98%
Energy Harvester Identification	92%	94%	92%	83%	75%	83%	83%	75%	83%
Solar Energy Availability	80%	81%	80%	82%	78%	82%	78%	68%	78%
Vibration Energy Availability	88%	89%	88%	85%	86%	85%	85%	85%	85%
Temp Energy Availability	87%	87%	87%	87%	87%	87%	87%	87%	87%

5.5 Attack Effectiveness against Defenses

There are three main defenses against network analysis attacks: packet padding, traffic injection, and traffic reshaping [6, 7, 11, 93]. The packet padding concatenates dummy bytes to packets to confuse attacks relying on packet lengths. Since our attacks only rely on packet timings, packet padding is inherently ineffective against our attacks. Therefore, we evaluate our attack performance against two defenses, traffic injection and reshaping.

Defense Design. Traffic injection introduces dummy packets to obfuscate the real network traffic in a crowd of dummy traffic. To integrate this defense, we introduce dummy traffic to the network packets collected from our intermittent devices. We inject dummy packets while the device has high energy, as it is infeasible for the device to send dummy packets in low-energy conditions.

Traffic reshaping introduces random delays between packets to confuse attacks that rely on inter-packet-arrival timings. We sample the delays from a uniform distribution $[0, I/4]$, where I is the device's packet transmission interval. This ensures the intermittent device can accumulate enough energy to send consecutive packets while introducing randomness to the device's network traffic.

Results. Table 6 shows our attack's effectiveness without any defense and with defenses integrated. We found that our attack achieves, on average, only 2.7% lower accuracy with existing defenses as they fail to hide an intermittent device's active period.

During traffic injection, even the dummy packets indicate the device is turned on. Since our attack only profiles the active periods of devices, the adversary does not need to differentiate between dummy and real packets while conducting the attack. As shown in Table 6, this allows an adversary to achieve 97% accuracy in identifying a victim intermittent device, 83% accuracy in recognizing its harvester, and 85% accuracy in inferring its energy availability.

In traffic reshaping, each observed packet still indicates the intermittent device's active period. Thus, our attack achieves 98% accuracy in identifying an intermittent device, 83% in recognizing its harvester, and 83% accuracy in inferring its energy availability.

Overall, our evaluation shows that existing network analysis defenses, although effective against prior network analysis attacks on IoT devices for fingerprinting them [6, 7, 11, 93], are insufficient against our physical side-channel attacks against intermittent devices and highlights the need for new countermeasures.

6 DISCUSSION AND LIMITATIONS

6.1 Discussion

Attack Transferability to Different Intermittent Devices. We evaluate our attack's transferability on another intermittent device with a 32-bit 48 MHz Arm processor, 384 KB flash memory, 88 KB SRAM memory, a custom voltage regulation circuit, and a BLE radio. It includes an accelerometer and temperature, humidity, and pressure sensors. It takes periodic sensor measurements, with the default interval set as two seconds, and sends them to a hub. For energy storage, it includes a 47 μF capacitor.

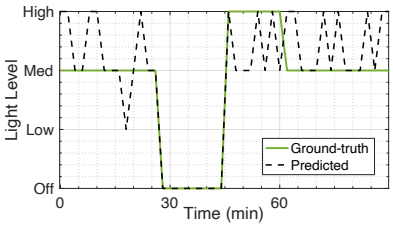


Figure 9: Real and predicted light levels in the smart home when a different intermittent device is used for testing.

With this device, we collect 1.5 hours of timestamped packet sequences in a smart home setting while the device is equipped with a solar harvester and configured with its default settings.

Our attack's accuracy is 62% when the inference model is trained with the CYBLE device and tested on the dataset collected with the ARM device. Figure 9 shows the ground-truth light levels in the smart home and those predicted with our attack. Although the accuracy decreases when the attack targets a different device, Figure 9 shows that the predictions are close to the real light levels. The adversary can discover if the light is on or off, but they cannot clearly distinguish between medium and high light levels.

Mitigation Methods. We introduce two defenses that leverage two properties critical to an intermittent device's active period, the energy storage size and packet transmission interval.

The first defense is using a larger energy storage (e.g., a supercapacitor with high capacitance). This would cause the device to turn off for a more extended period of time while it accumulates energy. When the device turns on, it also stays on for an extended period of time. This limits the adversary's ability to learn privacy-sensitive information as the adversary cannot know exactly when the energy is available. To confirm this, we have conducted preliminary experiments with a 1F supercapacitor connected to our device with the thermal harvester. We found it takes $\approx 1-2$ hours to charge the supercapacitor, depending on the temperature level. Thus, the adversary cannot infer the exact time when the energy was available (e.g., if the oven is high in the initial or the last 30 mins). Yet, this defense changes the device's operation, where the device turns off for a long time, which may not be desirable in certain applications (e.g., in vital sign sensing, frequent sensor readings are required).

The second defense is programming the device with a large packet transmission interval. Through this, the number of network packets the adversary can analyze becomes minimal, limiting its ability to recover privacy-sensitive information. For instance, we consider a device that reports sensor readings every 24 hour. Here, the device only needs to accumulate enough energy during this 24 hour, and the adversary cannot infer when the device stored its energy. Yet, similar to increasing the capacitor size, this defense also changes the device's operation, limiting its applicability.

Future work will analyze the trade-offs between our defenses and the intermittent devices' usability in different scenarios.

Other Possible Defenses. One may consider implementing different techniques to mitigate our attacks. First, the intermittent

device may randomly harvest energy to hide its energy availability patterns. However, this approach is highly undesirable since it may impair the device's normal operation [10]. Another potential defense involves leveraging MAC address randomization or MAC layer encryption techniques to prevent our attacks from separating the network traffic based on unique device identifiers. Yet, MAC address randomization is vulnerable to tracking techniques [12]. Additionally, physical-layer fingerprinting methods that leverage triangulation and radio signal properties can be integrated into our attack for traffic separation against these defenses [15, 37].

6.2 Limitations

Multiple Energy Sources. With the advancements in energy harvesting systems, intermittent devices with multiple energy sources (e.g., solar and vibration) are recently becoming popular [55, 72, 84]. Multiple energy sources allow these devices to increase their energy availability. Our physical side-channel attack can be applied to such devices by including more energy availability conditions in its dynamic grid testing based on all the harvesters the device is equipped with. Future work will evaluate our attack on intermittent devices with multiple harvesters.

Location Dependency. Attacks against certain intermittent devices may require the adversary to know the device's location. For instance, to learn the robot's position in the plant from a vibration-harvesting device, the adversary needs to know which robot this device is located at. To learn this information, the adversary can use localization methods or social engineering (e.g., phishing). Yet, attacks against most devices presented in Table 1 do not require the adversary to know the device's location as the energy source directly implies privacy-sensitive information. For instance, from a heartbeat harvesting device, the adversary can infer if the user is resting or exercising; from a solar-harvesting device, the adversary can infer if the users are at home; and from an RF harvesting device, the adversary can infer the number of people in the home without knowing the device's exact location.

7 RELATED WORK

Intermittent Device Security and Privacy. Initial works have conducted side-channel attacks against RFID tags to extract cryptographic keys and passwords [44, 46, 76]. Yet, these works are limited to RFID tags, and they do not infer privacy-sensitive information related to the energy availability of intermittent devices. A line of work has conducted physical attacks on intermittent devices to extract their checkpoints [85, 92]. To defend against them, secure checkpoints [51–53, 81] and remote attestation protocols [25, 77] have been proposed. Prior work also proposed techniques to enable secure program implementations [78] and optimize cryptographic implementations through pre-computations [91]. Recent works have focused on scheduling (e.g., multi-tenancy) [59, 73] and memory isolation to protect against malicious applications [39, 41]. Yet, none of these works consider side-channel attacks against the energy availability of intermittent devices.

Recent work uses radio frequency harvesting devices to conduct side-channel attacks against mobile devices [67]. This work assumes an adversary who compromised an RF harvesting device. The adversary then uses the deviations in the RF harvester's voltage output to infer mobile app activity in its surroundings. In contrast, we assume a remote adversary exploits the intermittent device's active period to infer its energy availability conditions. To our knowledge, this work is the first to propose remote side-channel attacks against intermittent devices to infer their energy availability.

Fingerprinting IoT Devices. Prior works fingerprint IoT devices' network behavior to infer the device types, events, and user activities in IoT environments (e.g., smart homes) [3, 5, 8, 11, 21, 61, 63, 69, 74, 80, 93, 96]. These works use diverse features such as packet sizes, directions, delays, protocol lists, flags, and ports to model network traffic characteristics of non-intermittent IoT devices with batteries. The prior work focuses on inferring privacy-sensitive information regarding the device functionality (e.g., whether a motion sensor generated a motion-detected event), whereas our attacks infer the device's energy availability, which is independent of the device functionality. Additionally, these techniques do not integrate unique intermittent device properties into their prediction algorithms and, therefore, fail to properly fingerprint the network behavior and active period of intermittent devices. To the best of our knowledge, ours is the first work that integrates the unique properties of intermittent devices (e.g., capacitor sizes, diverse harvesters) and analyzes their packet frequency to infer privacy-sensitive information regarding their energy availability.

There have also been various defenses proven effective against such attacks. These defenses use packet padding, traffic reshaping, and injection to prevent the adversary from characterizing the devices and events through the network traffic [6, 7, 11, 88, 93]. Yet, as shown in Section 5.5, such defenses are ineffective against our attacks as they fail to hide the intermittent device's active period.

8 CONCLUSIONS

In this paper, we present a novel remote side-channel attack against intermittent devices. Our attack exploits the intermittent device's active period (the times it turns on and off) from its inter-packet-arrival timings to infer its energy availability conditions. To this aim, we design a hierarchical inference framework that profiles the intermittent device's active period to discover if an intermittent device is present in an environment, recognize its harvester, and infer its energy availability conditions. We evaluate our attack in a real smart home and miniature manufacturing plant with three intermittent devices powered by solar, vibration, and temperature harvesters. Our experiments show that we can infer privacy-sensitive information (e.g., user presence, robot movements) with 85% accuracy over the three harvesters. Lastly, we show that existing defenses are insufficient to protect against our attack, and we propose two defenses that exploit unique intermittent device properties to mitigate it. Through this effort, we put forth an important step toward understanding the privacy implications of intermittent devices.

ACKNOWLEDGMENTS

We thank our shepherd and the anonymous reviewers for their comments and suggestions. We also thank Andrew Riordan, Haozhe Zhou, Tarcan Gul, and Rafael Zhu for their feedback on the earlier version of this paper. This work has been partially supported by the National Science Foundation (NSF) under grant CNS-2144645. The views expressed are those of the authors only.

REFERENCES

[1] 8Power 2023. 8power Machine Condition Monitoring. https://www.8power.com/applications/. [Online; accessed 10-Nov-2023].

[2] M. Abrams and J. Weiss. 2008. Malicious control system cyber security attack case study-Maroochy water services, Australia. In *Technical report, MITRE CORP MCLEAN VA MCLEAN.*

[3] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. 2020. Peek-a-boo: I see your smart home activities, even encrypted!. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec).*

[4] Kofi Sarpong Adu-Manu, Nadir Adam, Cristiano Tapparello, Hoda Ayatollahi, and Wendi Heinzelman. 2018. Energy-Harvesting Wireless Sensor Networks (EH-WSNs) A Review. In *ACM Transactions on Sensor Networks (TOSN).*

[5] Dilawer Ahmed, Anupam Das, and Fareed Zaffar. 2022. Analyzing the Feasibility and Generalizability of Fingerprinting Internet of Things Devices. In *Proceedings on Privacy Enhancing Technologies (PoPETs).*

[6] Ahmed Alshehri, Jacob Granley, and Chuan Yue. 2020. Attacking and protecting tunneled traffic of smart home devices. In *ACM Conference on Data and Application Security and Privacy (CODASPY).*

[7] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. 2019. Keeping the Smart Home Private with Smart (er) IoT Traffic Shaping. In *Proceedings on Privacy Enhancing Technologies (PoPETs).*

[8] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. 2017. Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic. In *arXiv preprint arXiv:1708.05044.*

[9] Omar Aragonez and Nathan Jackson. 2022. A zero power current and frequency sensor for smart grid applications. In *Smart Materials and Structures.*

[10] Abu Bakar, Alexander G Ross, Kasim Sinan Yildirim, and Josiah Hester. 2021. Rehash: A flexible, developer focused, heuristic adaptation platform for intermittently powered computing. In *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT).*

[11] Ludovic Barman, Alexandre Dumur, Apostolos Pyrgelis, and Jean-Pierre Hubaux. 2021. Every Byte Matters: Traffic Analysis of Bluetooth Wearable Devices. In *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT).*

[12] Johannes K Becker, David Li, and David Starobinski. 2019. Tracking Anonymized Bluetooth Devices. In *Proceedings on Privacy Enhancing Technologies (PoPETs).*

[13] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *AAAI Workshop on Knowledge Discovery in Databases.*

[14] Leo Breiman. 2001. Random forests. In *Machine learning.* Springer.

[15] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. 2008. Wireless device identification with radiometric signatures. In *ACM International Conference on Mobile Computing and Networking (MobiCom).*

[16] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. 2018. Screaming channels: When electromagnetic side channels meet radio transceivers. In *ACM SIGSAC Conference on Computer and Communications Security (CCS).*

[17] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. 2012. Similarity measures and dimensionality reduction techniques for time series data mining. In *Advances in data mining knowledge discovery and applications.*

[18] Salar Chamanian, Hasan Uluşan, Özge Zorlu, Sajjad Baghaee, Elif Uysal-Biyikoglu, and Haluk Külah. 2016. Wearable battery-less wireless sensor network with electromagnetic energy harvesting system. In *Elsevier Sensors and Actuators A: Physical.*

[19] Cleanspace 2023. CleanSpace Tag. https://our.clean.space/. [Online; accessed 10-January-2023].

[20] Alexei Colin and Brandon Lucia. 2016. Chain: tasks and channels for reliable intermittent programs. In *ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications.*

[21] Bogdan Copos, Karl Levitt, Matt Bishop, and Jeff Rowe. 2016. Is anybody home? inferring activity from smart home network traffic. In *IEEE Security and Privacy Workshops (SPW).*

[22] Damiano Crescini, Alessio Galli, Davide Alghisi, and Farid Touati. 2019. Ambient Monitoring WSNs with Harvesting-aware Power Management. In *International Symposium Electrical & Electronic Measurements Promote Industry 4.0.*

[23] Cypress 2023. Cypress Solar-Powered IoT Device Kit. https://www.digikey.com/en/products/detail/cypress-semiconductor-corp/S6SAE101A00SA1002/5697945. [Online; accessed 10-Nov-2023].

[24] Alfredo D'Elia, Luca Perilli, Fabio Viola, Luca Roffia, Francesco Antoniazzi, Roberto Canegallo, and Tullio Salmon Cinotti. 2016. A self-powered WSAN for energy efficient heat distribution. In *IEEE Sensors Applications Symposium (SAS).*

[25] Daniel Dinu, Archanaa S Khrishnan, and Patrick Schaumont. 2019. SIA: secure intermittent architecture for off-the-shelf resource-constrained microcontrollers. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST).*

[26] EnOcean RF 2023. EnOcean RF-powered IoT Developer Kit for Energy Harvesting Wireless Sensor Solutions. https://www.enocean.com/en/product/edk-350/. [Online; accessed 10-Nov-2023].

[27] EnOcean Solar 2023. EnOcean Solar-powered IoT Sensors. https://www.enocean.com/en/product-category/self-powered-sensors-finished-products/. [Online; accessed 10-Nov-2023].

[28] Everactive 2023. Everactive: The Self-Powered IoT Platform. https://everactive.com/. [Online; accessed 10-Nov-2023].

[29] Habiba Farrukh, Reham Mohamed, Aniket Nare, Antonio Bianchi, and Z Berkay Celik. 2023. LocIn: Inferring Semantic Location from Spatial Maps in Mixed Reality. In *USENIX Security.*

[30] Habiba Farrukh, Muslum Ozgur Ozmen, Faik Kerem Ors, and Z Berkay Celik. 2023. One key to rule them all: Secure group pairing for heterogeneous IoT devices. In *IEEE Symposium on Security and Privacy (S&P).*

[31] Habiba Farrukh, Tinghan Yang, Hanwen Xu, Yuxuan Yin, He Wang, and Z Berkay Celik. 2021. S3: Side-channel attack on stylus pencil through sensors. In *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT).*

[32] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. 2019. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In *NDSS.*

[33] Fenix 2023. The Fenix Hub. https://aquarobur.com/iot-products/sensor-nodes/. [Online; accessed 10-Nov-2023].

[34] FischerTechnik 2023. FischerTechnik Plant. https://www.fischertechnik.de/en/products/learning/training-models/554868-edu-training-factory-industry-4-0-24v-education. [Online; accessed 10-Nov-2023].

[35] Matthew Furlong, Josiah Hester, Kevin Storer, and Jacob Sorber. 2016. Realistic simulation for tiny batteryless sensors. In *International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems.*

[36] Kai Geissdoerfer, Mikołaj Chwalisz, and Marco Zimmerling. 2019. Shepherd: a portable testbed for the batteryless IoT. In *ACM Conference on Embedded Networked Sensor Systems (SenSys).*

[37] Hadi Givehchian, Nishant Bhaskar, Eliana Rodriguez Herrera, Héctor Rodrigo López Soto, Christian Dameff, Dinesh Bharadia, and Aaron Schulman. 2022. Evaluating physical-layer ble location tracking attacks on mobile devices. In *IEEE Symposium on Security and Privacy (S&P).*

[38] Rishabh Goel, Tien Pham, Phuc Nguyen, and Josiah Hester. 2023. Exploring Batteryless UAVs by Mimicking Bird Flight. In *Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet).*

[39] Michele Grisafi, Mahmoud Ammar, Kasim Sinan Yildirim, and Bruno Crispo. 2022. MPI: Memory Protection for Intermittent Computing. *IEEE Transactions on Information Forensics and Security (TIFS).*

[40] Jinsong Han, Chen Qian, Yuqin Yang, Ge Wang, Han Ding, Xin Li, and Kui Ren. 2018. Butterfly: Environment-independent physical-layer authentication for passive RFID. In *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT).*

[41] Taylor Hardin, Ryan Scott, Patrick Proctor, Josiah Hester, Jacob Sorber, and David Kotz. 2018. Application memory isolation on ultra-low-power MCUs. In *USENIX ATC.* 127–132.

[42] Josiah Hester and Jacob Sorber. 2017. The future of sensing is batteryless, intermittent, and awesome. In *ACM Conference on Embedded Network Sensor Systems (SenSys).*

[43] Qianyi Huang, Yan Mei, Wei Wang, and Qian Zhang. 2018. Toward battery-free wearable devices: The synergy between two feet. In *ACM Transactions on Cyber-Physical Systems.*

[44] Michael Hutter, Stefan Mangard, and Martin Feldhofer. 2007. Power and EM Attacks on Passive 13.56\,MHz RFID Devices. In *Cryptographic Hardware and Embedded Systems (CHES).*

[45] Geon-Tae Hwang, Venkateswarlu Annapureddy, Jae Hyun Han, Daniel J Joe, Changyeon Baek, Dae Yong Park, Dong Hyun Kim, Jung Hwan Park, Chang Kyu Jeong, Kwi-Il Park, et al. 2016. Self-powered wireless sensor node enabled by an aerosol-deposited PZT flexible energy harvester. In *Advanced Energy Materials.*

[46] Timo Kasper, David Oswald, and Christof Paar. 2009. EM side-channel attacks on commercial contactless smartcards using low-cost equipment. In *Information Security Applications.*

[47] Dong Hyun Kim, Hong Ju Shin, Hyunseung Lee, Chang Kyu Jeong, Hyewon Park, Geon-Tae Hwang, Ho-Yong Lee, Daniel J Joe, Jae Hyun Han, Seung Hyun Lee, et al. 2017. In vivo self-powered wireless transmission using biocompatible flexible energy harvesters. In *Advanced Functional Materials.*

[48] Bo-Gun Koo, Dong-Jin Shin, Dong-Hwan Lim, Min-Soo Kim, In-Sung Kim, and Soon-Jong Jeong. 2021. Properties of Car-Embedded Vibrating Type Piezoelectric Harvesting System. In *Applied Sciences.*

[49] Vito Kortbeek, Abu Bakar, Stefany Cruz, Kasim Sinan Yildirim, Przemysław Pawełczak, and Josiah Hester. 2020. Bfree: Enabling battery-free sensor prototyping with python. In *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT).*

[50] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. In *Physical review E.*

[51] Archanaa S Krishnan and Patrick Schaumont. 2022. Benchmarking And Configuring Security Levels In Intermittent Computing. In *ACM Transactions on Embedded Computing Systems (TECS).*

[52] Archanaa S Krishnan, Charles Suslowicz, Daniel Dinu, and Patrick Schaumont. 2019. Secure intermittent computing protocol: Protecting state across power loss. In *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

[53] Archanaa S Krishnan, Charles Suslowicz, and Patrick Schaumont. 2020. Secure and stateful power transitions in embedded systems. In *Journal of Hardware and Systems Security*. Springer.

[54] Feng Li, Yanbing Yang, Zicheng Chi, Liya Zhao, Yaowen Yang, and Jun Luo. 2018. Trinity: Enabling self-sustaining WSNs indoors with energy-free sensing and networking. In *ACM Transactions on Embedded Computing Systems (TECS)*.

[55] Huicong Liu, Hailing Fu, Lining Sun, Chengkuo Lee, and Eric M Yeatman. 2021. Hybrid energy harvesting technology: From materials, structural design, system integration to applications. In *Renewable and sustainable energy reviews*. Elsevier.

[56] Brandon Lucia, Vignesh Balaji, Alexei Colin, Kiwan Maeng, and Emily Ruppel. 2017. Intermittent computing: Challenges and opportunities. In *Summit on Advances in Programming Languages (SNAPL)*.

[57] Brandon Lucia and Benjamin Ransford. 2015. A simpler, safer programming and execution model for intermittent systems. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*.

[58] Dong Ma, Guohao Lan, Mahbub Hassan, Wen Hu, and Sajal K Das. 2019. Sensing, computing, and communications for energy harvesting IoTs: A survey. In *IEEE Communications Surveys & Tutorials*.

[59] Amjad Yousef Majid, Carlo Delle Donne, Kiwan Maeng, Alexei Colin, Kasim Sinan Yildirim, Brandon Lucia, and Przemysław Pawełczak. 2020. Dynamic task-based intermittent execution for energy-harvesting devices. *ACM Transactions on Sensor Networks (TOSN)*.

[60] Yuyi Mao, Jun Zhang, and Khaled B Letaief. 2016. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. In *IEEE Journal on Selected Areas in Communications (SAC)*.

[61] Samuel Marchal, Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and N Asokan. 2019. Audi: Toward autonomous IoT device-type identification using periodic communication. In *IEEE Journal on Selected Areas in Communications (SAC)*.

[62] Patrick P Mercier, Andrew C Lysaght, Saurav Bandyopadhyay, Anantha P Chandrakasan, and Konstantina M Stankovic. 2012. Energy extraction from the biologic battery in the inner ear. In *Nature biotechnology*.

[63] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2017. IoT sentinel: Automated device-type identification for security enforcement in IoT. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*.

[64] Reham Mohamed, Habiba Farrukh, Yidong Lu, He Wang, and Z Berkay Celik. 2023. Istelan: Disclosing sensitive user information by mobile magnetometer from finger touches. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*.

[65] Phillip Nadeau, Dina El-Damak, Dean Glettig, Yong Lin Kong, Stacy Mo, Cody Cleveland, Lucas Booth, Niclas Roxhed, Robert Langer, Anantha P Chandrakasan, et al. 2017. Prolonged energy harvesting for ingestible devices. In *Nature Biomedical Engineering*.

[66] Y Naruse, N Matsubara, K Mabuchi, M Izumi, and S Suzuki. 2009. Electrostatic micro power generation from low-frequency vibration such as human motion. In *Journal of Micromechanics and Microengineering*.

[67] Tao Ni, Guohao Lan, Jia Wang, Qingchuan Zhao, and Weitao Xu. 2023. Eavesdropping Mobile App Activity via Radio-Frequency Energy Harvesting. In *USENIX Security*.

[68] Nowi 2023. Smart Footware. https://www.nowi-energy.com/smart-footwear/. [Online; accessed 10-Nov-2023].

[69] TJ OConnor, Reham Mohamed, Markus Miettinen, William Enck, Bradley Reaves, and Ahmad-Reza Sadeghi. 2019. HomeSnitch: behavior transparency and control for smart home IoT devices. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

[70] Muslum Ozgur Ozmen, Xuansong Li, Andrew Chu, Z. Berkay Celik, Bardh Hoxha, and Xiangyu Zhang. 2022. Discovering Physical Interaction Vulnerabilities in IoT Deployments. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[71] Muslum Ozgur Ozmen, Ruoyu Song, Habiba Farrukh, and Z Berkay Celik. 2023. Evasion Attacks and Defenses on Smart Home Physical Event Verification. In *NDSS*.

[72] Yaokun Pang, Yunteng Cao, Masoud Derakhshani, Yuhui Fang, Zhong Lin Wang, and Changyong Cao. 2021. Hybrid energy-harvesting systems based on triboelectric nanogenerators. In *Matter*. Elsevier.

[73] Dimitris Patoukas, Kasim Sinan Yildirim, Amjad Yousef Majid, Josiah Hester, and Przemysław Pawełczak. 2018. Feasibility of multi-tenancy on intermittent power. In *Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*.

[74] Roberto Perdisci, Thomas Papastergiou, Omar Alrawi, and Manos Antonakakis. 2020. IoTfinder: Efficient large-scale identification of IoT devices via passive dns traffic analysis. In *IEEE European Symposium on Security and Privacy (EuroS&P)*.

[75] Senthilkumar Chinnappa Gounder Periaswamy, Dale R Thompson, and Jia Di. 2010. Fingerprinting RFID tags. In *IEEE Transactions on Dependable and Secure Computing (TDSC)*.

[76] Thomas Plos. 2008. Susceptibility of UHF RFID tags to electromagnetic analysis. In *The Cryptographers' Track at the RSA Conference (CT-RSA)*.

[77] Md Masoom Rabbani, Edlira Dushku, Jo Vliegen, An Braeken, Nicola Dragoni, and Nele Mentens. 2021. RESERVE: Remote Attestation of Intermittent IoT devices. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

[78] Amir Rahmati, Mastooreh Salajegheh, Daniel E Holcomb, Jacob Sorber, Wayne P Burleson, and Kevin Fu. 2012. TARDIS: Time and Remanence Decay in SRAM to Implement Secure Protocols on Embedded Devices without Clocks. In *USENIX Security*.

[79] Benjamin Ransford, Jacob Sorber, and Kevin Fu. 2011. Mementos: System support for long-running computation on RFID-scale devices. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

[80] Jingjing Ren, Daniel J Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. 2019. Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach. In *Internet Measurement Conference (IMC)*.

[81] Arman Roohi, Ronald F DeMara, Longfei Wang, and Selcuk Kose. 2017. Secure intermittent-robust computation for energy harvesting device security and outage resilience. In *IEEE SmartWorld*.

[82] Brian C Ross. 2014. Mutual information between discrete and continuous data sets. In *PloS One*.

[83] RSL10 2023. Solar Cell Multi-Sensor Platform. https://www.digikey.com/en/products/detail/onsemi/RSL10-SOLARSENS-GEVK/10261083. [Online; accessed 10-July-2023].

[84] Hanjun Ryu, Hong-Joon Yoon, and Sang-Woo Kim. 2019. Hybrid energy harvesters: toward sustainable energy harvesting. In *Advanced Materials*.

[85] Archanaa S Krishnan and Patrick Schaumont. 2018. Exploiting security vulnerabilities in intermittent computing. In *International Conference on Security, Privacy, and Applied Cryptography Engineering (SPACE)*. Springer.

[86] Raffaele Salvati, Valentina Palazzi, and Luca Roselli. 2022. IoT Wearable EH system based on Wrist Motion Kinetic Energy Harvesting. In *IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*.

[87] Philipp Schlögl. 2018. *An Energy harvesting powered sensor node for machine condition monitoring*. Ph. D. Dissertation. Wien.

[88] Akshaye Shenoi, Prasanna Karthik, Kanav Sabharwal, Li Jialin, and Dinil Mon Divakaran. 2023. iPET: Privacy Enhancing Traffic Perturbations for Secure IoT Communications. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*.

[89] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2018. Classifying IoT devices in smart environments using network traffic characteristics. In *IEEE Transactions on Mobile Computing*.

[90] Sivert T Sliper, William Wang, Nikos Nikoleris, Alexander Weddell, and Geoff Merrett. 2020. Fused: closed-loop performance and energy simulation of embedded systems. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*.

[91] Charles Suslowicz, Archanaa S Krishnan, and Patrick Schaumont. 2017. Optimizing cryptography in energy harvesting applications. In *Workshop on Attacks and Solutions in Hardware Security (ASHES)*.

[92] Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. 2020. Security in energy harvesting networks: a survey of current solutions and research challenges. In *IEEE Communications Surveys & Tutorials*.

[93] Rahmadi Trimananda, Janus Varmarken, Athina Markopoulou, and Brian Demsky. 2020. Packet-level signatures for smart home devices. In *NDSS*.

[94] Wiliot 2023. IoT Pixel. https://www.wiliot.com/product/iot-pixel#01. [Online; accessed 10-Nov-2023].

[95] Mu Zhang, Chien-Ying Chen, Bin-Chou Kao, Yassine Qamsane, Yuru Shao, Yikai Lin, Elaine Shi, Sibin Mohan, Kira Barton, James Moyne, and Z. Morley Mao. 2019. Towards Automated Safety Vetting of PLC Code in Real-World Plants. In *IEEE Symposium on Security and Privacy (S&P)*.

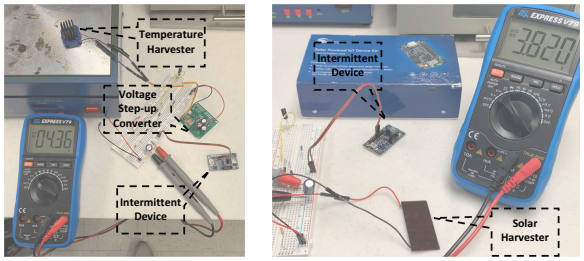
[96] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. 2018. Homonit: Monitoring smart home apps from encrypted traffic. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

A EXPERIMENTAL SETUP

Figure 10 demonstrates the lab environment setup to generate the training datasets (DS_2) for the temperature and solar harvesters. Figure 11 demonstrates our manufacturing plant testbed setup for generating the testing dataset (DS_3).

Table 7: Comparison of the effectiveness of different classifiers.

	Our Attack			SVM			Nearest Neighbor			Naive Bayes			Multi-layer Perceptron		
Experiment	**Acc.**	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**	**Acc.**	**Pre.**	**Rec.**
Stage1: Intermittent Device Identification	98%	99%	98%	94%	97%	94%	31%	94%	31%	92%	96%	92%	75%	95%	75%
Stage2: Energy Harvester Identification	92%	94%	92%	17%	3%	17%	83%	75%	83%	83%	75%	83%	83%	75%	83%
Exp_1: Solar Energy Availability	80%	81%	80%	60%	79%	60%	58%	77%	58%	62%	77%	62%	58%	67%	58%
Exp_2: Vibration Energy Availability	88%	89%	88%	88%	89%	88%	81%	81%	81%	85%	85%	85%	85%	85%	85%
Exp_3: Temp Energy Availability	87%	87%	87%	53%	55%	53%	80%	81%	80%	73%	73%	73%	80%	81%	80%



(a) (b)

Figure 10: Our experimental setups for generating the active period modeling dataset (DS_2) with a (a) temperature harvester and (b) solar harvester.

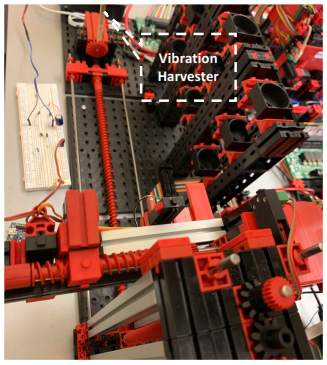


Figure 11: Our experimental setup for generating the testing dataset (DS_3) with the vibration harvester. The harvester is connected to the toothed gear that controls the high-bay warehouse position (instead of being connected to the vacuum gripper robot in Section 3.1).

B ATTACK ACCURACY WITH DIFFERENT CLASSIFIERS

Table 7 presents the accuracy, precision, and recall of different classifiers for each stage of our hierarchical inference framework compared to our random forest classifiers.