Detection of Network Scanning in the Internet of Things through Energy Consumption Analysis

Tommaso Ottali
University of Pisa
Pisa, Italy

Pericle Perazzo
University of Pisa
Pisa, Italy
pericle.perazzo@unipi.it

Alessio Vecchio
University of Pisa
Pisa, Italy
alessio.vecchio@unipi.it

***Abstract*—This study describes our experience in detecting network scanning of IoT devices through energy consumption analysis. We carried out a set of experiments where five benign activities (web requests, updates, file transfers, audio streaming, and video streaming) and one malicious activity (Nmap network scanning) were considered. Energy traces were collected using a high-accuracy power monitor. Machine learning classification using a Neural Network achieved 97.14% accuracy in identifying the malicious scanning activity. Results demonstrate that energy-based detection can effectively distinguish between normal operations and cyberattacks, offering a promising, passive security approach for IoT devices.**

I. Introduction

In recent years, the exponential spread of Internet of Things (IoT) devices has radically transformed the way we interact with technology. From smart appliances to automated industrial systems, IoT devices have become an integral part of our daily lives [1]. However, alongside these opportunities, new cybersecurity challenges are emerging [2]. The primary challenges in ensuring the security of these devices stem from their limited computational resources and prolonged periods of unattended operation, which make it more difficult to incorporate cybersecurity defense techniques or detect attacks. In fact, on a regular computer, tools such as antivirus software, firewalls, intrusion detection systems (IDS), or traffic analysis can be run in real time to identify suspicious behavior. However, these same tools are often too resource-intensive for the microcontrollers or resource-constrained CPUs that power IoT devices. This scenario calls for a radical rethinking of the approach to security in these environments.

Based on these premises, this work describes our experience in detecting attacks on IoT devices using energy consumption analysis. The underlying assumption of this approach is that every activity performed by a device—whether legitimate or malicious—results in a certain energy consumption [3]. Therefore, by accurately monitoring the device's energy profiles during normal operation and comparing them with those recorded during an attack, it is possible to identify distinctive patterns that indicate anomalous activity.

We conducted a set of experiments on an IoT device. The first five experiments simulated normal, everyday activities that a device might perform: HTTP requests, package installations, data transmission to other devices, and audio/video data transmission. In another experiment, we simulated a cyberattack through a network scanning activity using the Nmap tool—a technique commonly used to detect active devices and services within a network, often as a preliminary phase of an attack. During all experiments, the device's energy consumption was monitored in real time using high-precision tools. The collected data was then analyzed to identify any differences in energy profiles between legitimate and malicious activities. The results show that it is indeed possible to distinguish with high accuracy between normal operation and situations in which the device was under attack.

In particular, the Nmap scanning activity generated an energy consumption pattern clearly different from the other activities. This anomalous behavior, although not immediately noticeable at a functional level (the device continues to operate normally), was evident in terms of energy, suggesting that a power monitor can act as a "passive security sensor" capable of detecting intrusions.

II. Related Work

One of the first studies presenting the idea of using energy-related information for detecting malicious activity is described in [3]. The study focuses on the detection of malware on medical devices and industrial computers. Results show high levels of accuracy, ranging from 85% to 99% depending on the specific scenario considered. The medical device domain is particularly interesting since software updates and installation of anti-virus software are not always allowed by the manufacturer.

Power consumption traces are also used in [4] to detect attacks carried out by a smart camera. In this case, the smart camera is both the victim and the source of brute force attacks. The camera also carries out a TCP SYN flood attack against a server. Classification techniques, such as Support Vector Machine, k-Nearest Neighbors, and Neural Networks, are used to distinguish the normal activity periods from the anomalous ones.

Another study considered IoT devices under Distributed Denial of Service (DDoS) and Man-in-the-Middle attacks [5]. In this case, the power trace was collected using specifically designed hardware able to send information to a smartphone.

© 2024 IFIP

978-3-903176-77-5© 2025 IFIP

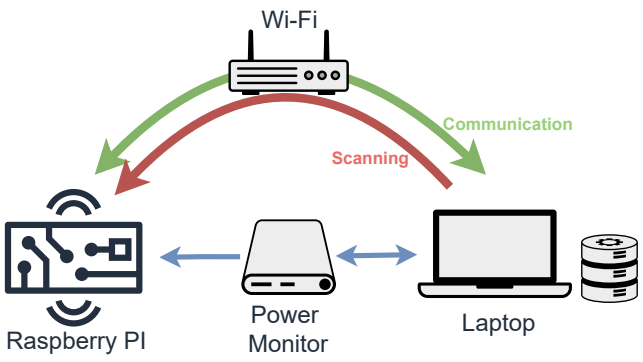

Fig. 1. Experimental setup: the Raspberry Pi is powered by the power monitor; the power monitor is controlled by the laptop and sends samples to the latter; communication and device scanning take place via Wi-Fi.

The considered smart devices included a smart camera, a smart speaker, and a smart light.

A testbed based on Single Board Computers (SBCs) was used in a capture-the-flag scenario [6]. The setup also included a number of sensors communicating with the SBC via Bluetooth. Power traces were collected and visually inspected, but not classified.

Besides cyber attacks, Shi et al. also considered some physical attacks, such as heating and power line cut, again using energy measurements in the detection phase [7].

The impact of networking protocols on energy consumption has gained attention over the last few years. The reasons are mostly due to the increased usage of battery-powered smart devices. In [8], the impact of a Wi-Fi deauthentication attack was evaluated both in terms of degraded Age of Information, a metric that measures the freshness of data [9], and in terms of wasted energy. The impact, in terms of energy consumption, of the different versions of the HTTP protocol in an IoT scenario has been evaluated in [10], [11].

More commonly, methods for detecting attacks in the IoT domain are based on the analysis of network traffic. Examples include [12]–[16].

III. Experimental Setup

In this Section, we describe the experimental setup used to collect the power traces when the device performed both benign and malicious activities. The experimental setup is depicted in Figure 1. For the experiments, we used a Raspberry Pi 3 Model B+[1], an SBC widely used in the maker, educational, and semi-professional IoT fields. The Raspberry Pi 3B+ mounts a 1.4GHz 64-bit quad-core processor, with 1GB SDRAM, and a dual-band Wi-Fi transceiver. Such a board was considered for this type of study because it is powerful enough to support a wide range of applications, yet still representative of devices with limited computational resources. Moreover, its broad hardware and software support facilitates interfacing with external measurement tools and the implementation of the considered activities. All experiments were conducted using the Raspberry Pi OS Lite operating system. Using a minimal OS like Raspberry Pi OS Lite reduces the number of background processes, avoiding the loading of unnecessary components such as desktop environments, graphical services, or window managers that are not representative of IoT contexts.

To monitor the power consumption of the Raspberry Pi, an Otii Arc Pro[2] was used. The Otii Arc Pro allows powering the device under test and, at the same time, measuring its power consumption. The Otii Arc Pro allows for measurement with an accuracy of $\pm(0.1\% + 150$ uA), enabling detailed tracking of the energy profile of the device under analysis. The sampling rate was 4000 samples per second.

The experimental setup also included a laptop, used for: i) controlling the Otii Arc and storing the power consumption traces, and ii) operating as a server for those activities requiring communication with an external host. The Raspberry Pi and the laptop were both connected to the same LAN using Wi-Fi. In particular, the Wi-Fi connection was used for the following purposes:

- Perform simulated network activities using the `scp` command, where the laptop acted as a remote server from which files were downloaded, and using the `ffmpeg` command, where the laptop received audio and video streams.
- Simulate a cyberattack via Nmap scanning launched from the laptop toward the Raspberry Pi.

The Raspberry Pi operated in headless mode, i.e. without keyboard and monitor, to avoid including the energy consumption of peripherals in the traces and to match the operational configuration of many unattended IoT devices.

IV. Data Collection

To test the feasibility of detecting a cyberattack through energy consumption analysis, six experiments were designed and conducted. The first five represent benign activities—typical operations performed in real IoT environments—while the sixth experiment simulates a malicious activity, specifically a network scan using Nmap.

A. Benign Activities

The benign activities selected for the experiment were chosen to represent realistic and common operations in the lifecycle of an IoT device. Each activity reproduces a concrete functionality that a device might perform autonomously or upon request, such as sending data, receiving updates, or transferring multimedia content. Additionally, these activities were selected to cover different types of system load, both in terms of computation and network usage. Analyzing the energy profile associated with these operations allowed us to define a reliable "energy signature" for legitimate behaviors, which can be used as a baseline to identify anomalies or suspicious activity.

[1]https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/.

[2]https://www.qoitech.com/otii-arc-pro/.

1) HTTP Requests: The first activity involved repeated execution of the `curl` command to perform HTTP GET requests to various web servers. A list of popular websites was selected, and the websites were contacted cyclically for the duration of the experiment. The payload size of the reply depends on the requested site. Requests were executed as quickly as possible—each request was sent out immediately after completing the previous one. `curl` was configured to abort the current request-response cycle if more than 10 s were needed. Overall, the goal is to simulate an IoT telemetry scenario, where the device periodically contacts a server to send data or receive information. Devices such as thermostats, environmental sensors, or smart meters use HTTP or HTTPS protocols to communicate with the cloud.

2) Software Update: The Raspberry Pi performed software update operations using the APT package manager. Software packages were selected that would not alter the system or require invasive configurations. The use of APT was managed cyclically: once all packages were downloaded and installed, they were removed so they could be re-downloaded and re-installed at the beginning of the next cycle. Over-the-air updates are common in many IoT devices, especially in managed or industrial environments. This activity simulates automatic maintenance, characterized by intensive network usage and short computational peaks due to package analysis and installation.

3) Data Transfer: File transfers were performed between the Raspberry Pi and a remote server (the laptop) using `scp`. A folder containing a 1GB file and some smaller files was copied onto the remote machine and then deleted via ssh. Secure data transfer is a frequent activity in many IoT contexts, such as uploading logs, images, or recorded videos, or synchronizing diagnostic files. The scp protocol was used because it is widely adopted and provides a good balance between simplicity and security.

4) Audio Streaming: `ffmpeg` was used to stream an MP3 file from the Raspberry Pi to the laptop. In this case, the operation is not cyclic: a 30-minute audio file was streamed. The SDP transmission phase was not included in the experiment. The SDP file contains necessary information for the recipient to correctly receive the RTP stream, such as stream type (MP3), codec (AAC), port, and other useful details. This was done because, typically, smart home devices that stream audio/video do so to the same recipient device, which already knows the characteristics of the incoming stream. On the receiving device, the command `ffplay` was executed to receive the stream. Overall, this activity mimics the behavior of IoT devices handling audio streams, such as smart speakers, intercoms, baby monitors, or voice assistants. It is common for the audio stream to be sent to another device on the same network for storage or playback, or to a remote server for analysis.

5) Video Streaming: Also in this case, `ffmpeg` was used to stream a 30-minute video file from the Raspberry Pi to the laptop using the RTP protocol. Similarly to the previous activity, the SDP transmission phase was not included in the experiment. On the laptop, `ffplay` was used to receive the video stream. This activity represents a typical use case for IoT devices with video capabilities, such as surveillance cameras, baby monitors, dash cams, or video doorbells. It is rather common for the video stream to be sent to another device for storage/viewing.

B. Malicious Activity

The Raspberry Pi was subjected to active scanning by another device using Nmap with aggressive options to simulate a malicious reconnaissance phase involving information gathering and probing of exposed services. Network scanning is one of the fundamental phases in any cyberattack operation. It is a technique used to identify which devices are active on a network, which ports are open, and which services are listening. The most widely used and powerful tool for this type of operation is Nmap (Network Mapper) [17], an open-source utility commonly used by security professionals for penetration testing, as well as by attackers with malicious intent. In the context of IoT devices, scanning represents a real and current threat. Devices connected to the Internet (such as IP cameras, environmental sensors, home or industrial automation systems) are often exposed without proper security configurations. Attackers can use Nmap to detect: i) IP address and operating system of the device, ii) open TCP/UDP ports and associated services, iii) versions of running software, iv) presence of known vulnerabilities, and v) default credentials or weak access controls. This information can then be used to plan targeted attacks such as exploit execution, credential brute-forcing, or malicious command injection.

Using Nmap scanning as the malicious activity in the experiment is therefore a realistic and coherent choice aligned with real-world attack scenarios. An Nmap-based attack does not directly compromise the target device but actively and intensively scans it, often generating anomalous traffic and system responses that may be reflected in variations in energy consumption. This characteristic makes it particularly interesting for evaluating the possibility of detecting anomalous behavior through energy profiling, without needing to analyze traffic or system logs directly.

During the experiment, Nmap scanning was executed from the laptop to the Raspberry Pi using the following parameters:

- `-T5`: Sets the timing aggressiveness to the maximum level (range T0 to T5). The command runs as fast as possible with minimal packet delay.
- `--max-rate 10000` and `--min-rate 5000`: Limit the packet sending rate between 5000 and 10000 packets per second.
- `-p-`: Scans all 65,535 TCP ports, not just the common ones. This is typical in advanced reconnaissance.
- `-sS`: Performs a SYN scan (half-open scan), which is fast because it doesn't complete the TCP handshake.
- `-sU`: Also scans UDP ports, which are often overlooked but may expose vulnerable services (DNS, SNMP, etc.).

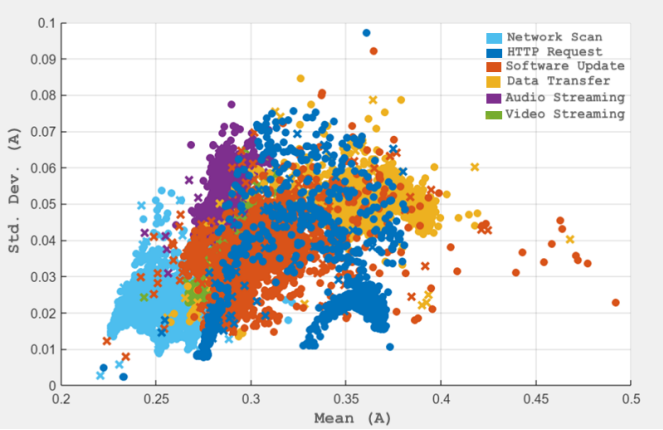

Fig. 2. Scatter plot of the different activities in the feature space (only two features, mean and standard deviation); Xes represent misclassifications, and dots represent correct classifications.

- `-Pn`: Disables initial ping, assuming the target is active—useful in networks where ICMP pings are blocked or filtered.
- `--script "default,vuln,auth,brute"`: Activates a set of predefined scripts that perform further analysis on discovered ports and services, such as checking for known vulnerabilities, authentication attempts, and brute-force attacks.

C. Features

To identify anomalies in the energy behavior of the IoT device, the current traces were segmented using non-overlapping time windows of 1 second. For each window, the following commonly used features were extracted: mean, standard deviation, kurtosis, skewness, minimum, maximum, median.

V. Results

The different activities were classified using some popular Machine Learning (ML) techniques: Neural Networks, SVMs, decision trees, and ensembles. The dataset was subjected to 5-fold cross-validation, a technique that splits the dataset into five blocks: in each iteration, one is used for testing while the remaining four are used for training. This method ensures robust model evaluation and reduces the risk of overfitting.

Table I shows the confusion matrix obtained when using Neural Networks for classification. Particular attention is given to the Network Scan class. In this case, out of 1,856 actual instances, 1,803 were correctly classified. Misclassifications were limited: the model confused 4 instances with the HTTP Request class, 27 with the Software Update class, 6 with the Data Transfer class, 8 with Audio Streaming, and 8 with Video Streaming. The scatter plot (Figure 2) shows the distribution of instances and their classification based on two extracted features: the mean and standard deviation. Network scanning, highlighted in light blue, occupies a clearly distinct region of the plane, generally concentrated in the lower-left area of the chart (for the considered features). This indicates that the energy windows related to the Nmap attack tend to have relatively lower and consistent average consumption and variability. In contrast, the five benign activities appear more overlapped and scattered across the graph.

Calculating the percentage of correct classifications, we obtain an accuracy of 97.14% for the malicious activity. The total number of errors was 53, corresponding to an error rate of 2.86% for this class. This result is highly significant: being able to identify a potentially harmful activity with over 97% accuracy suggests that energy analysis can be an effective method for detecting anomalous behavior in IoT devices.

Other ML techniques also achieved comparable results, both in terms of overall accuracy and in correctly identifying the malicious activity. In particular, models such as Ensemble (Bagged Trees) and Cubic SVM, both with a global accuracy of 95.7% (slightly lower than the neural network), demonstrated strong reliability in data classification. Focusing on the malicious activity, the Ensemble model correctly identified 1,814 out of 1,857 instances, achieving a specific accuracy of 97.69%. Similarly, the Cubic SVM correctly classified 1,816 network scanning instances, with an accuracy of 97.79%. These figures confirm that the attack produces an energy profile that is recognizable across different ML approaches.

The fact that models from different families—a neural network, an ensemble of trees, and a non-linear SVM—all achieved excellent results is a major strength. It means that the selected energy features are highly informative and that the class separability is sufficient to allow good generalization across diverse approaches. In other words, the system is not tied to a single machine learning architecture, but can be adapted to different contexts, considering computational constraints or specific requirements. This observation is particularly important in the IoT domain, where available hardware resources can vary significantly from one device to another. The ability to successfully implement lighter models paves the way for an on-board implementation of the detection system.

In conclusion, the results obtained from alternative models further strengthen the validity of the methodology, demonstrating its robustness and replicability. Regardless of the algorithm used, the energy pattern associated with the attack proves to be sufficiently distinctive to be recognized with high accuracy, a further step towards practical energy-behavioral security solutions in the IoT domain.

VI. Conclusion and Future Work

The results obtained in this study strengthen the potential of energy consumption analysis as an innovative method for detecting malicious activity in IoT environments. With an accuracy exceeding 97% in identifying Nmap scanning, the developed model highlights the ability to effectively distinguish between legitimate and suspicious behaviors, without relying on invasive techniques.

The dataset containing the power consumption traces of all the activities is publicly available [18].

However, despite these results, the approach presents some aspects that must be carefully considered. First, a device's

TABLE I
CONFUSION MATRIX

True Class	Predicted Class					
	HTTP Requests	Software Update	Data Transfer	Audio Str.	Video Str.	Network Scan
HTTP Requests	**1847**	38	1	0	2	6
Software Update	43	**1612**	52	13	54	28
Data Transfer	3	39	**1759**	4	2	9
Audio Str.	0	12	2	**1775**	13	4
Video Str.	4	56	0	11	**1726**	7
Network Scan	4	27	6	8	8	**1803**

energy consumption can be influenced by numerous external variables—environmental conditions, network instability—which may generate false positives or mask malicious activity. Moreover, the effectiveness of the approach has been demonstrated on a single type of attack (network scanning) and a single type of device; generalizing the results requires further validation across a broader range of scenarios, hardware platforms, and attack techniques. Another critical point concerns the time scale of the analysis: while the one-second windows used in the study offer a good compromise between responsiveness and accuracy, real-world environments may require more dynamic or adaptive windows capable of capturing more complex or subtle patterns.

Future work should focus on: extending the methodology to diverse attack types, including stealthy malware infection (e.g., botnet) or SSH bruteforcing; validating across different hardware platforms; integrating multimodal signals (CPU and memory usage); developing lightweight on-device classifiers; and testing in realistic environments with background noise.

ACKNOWLEDGMENT

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP J33C22002810001, partnership on "SEcurity and RIghts In the CyberSpace" (PE00000014 - program "SERICS"), and by the Italian Ministry of Education and Research (MUR) in the framework of the FoReLab project (Departments of Excellence).

REFERENCES

[1] K. Rose, S. Eldridge, L. Chapin *et al.*, "The internet of things: An overview," *The internet society (ISOC)*, vol. 80, no. 15, pp. 1–53, 2015.

[2] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517301455

[3] S. S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber, W. Xu, and K. Fu, "WattsUpDoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices," in *2013 USENIX Workshop on Health Information Technologies (HealthTech 13)*. Washington, D.C.: USENIX Association, Aug. 2013. [Online]. Available: https://www.usenix.org/conference/healthtech13/workshop-program/presentation/clark

[4] K. Nimmy, M. Dilraj, S. Sankaran, and K. Achuthan, "Leveraging power consumption for anomaly detection on iot devices in smart homes," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 10, pp. 14 045–14 056, 2023. [Online]. Available: https://doi.org/10.1007/s12652-022-04110-6

[5] A. J. Majumder, J. D. Miller, C. B. Veilleux, and A. A. Asif, "Smartpower: A smart cyber-physical system to detect iot security threat through behavioral power profiling," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 1041–1049.

[6] D. Lightbody, D.-M. Ngo, A. Temko, C. C. Murphy, and E. Popovici, "Attacks on iot: Side-channel power acquisition framework for intrusion detection," *Future Internet*, vol. 15, no. 5, 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/5/187

[7] Y. Shi, F. Li, W. Song, X.-Y. Li, and J. Ye, "Energy audition based cyber-physical attack detection system in iot," in *Proceedings of the ACM Turing Celebration Conference - China*, ser. ACM TURC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3321408.3321588

[8] P. Perazzo, M. Paladini, and A. Vecchio, "Connectivity and energy consumption of cyber-physical systems under wi-fi attack," in *2024 15th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2024, pp. 55–59.

[9] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[10] C. Caiazza, V. Luconi, and A. Vecchio, "Energy consumption of smartphones and iot devices when using different versions of the http protocol," *Pervasive and Mobile Computing*, vol. 97, p. 101871, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119223001293

[11] ——, "Measuring the energy of smartphone communications in the edge-cloud continuum: Approaches, challenges, and a case study," *IEEE Internet Computing*, vol. 27, no. 6, pp. 29–35, 2023.

[12] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for iot networks through deep learning model," *Computers and Electrical Engineering*, vol. 99, p. 107810, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790622001100

[13] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for iot application," *Discover Internet of Things*, vol. 3, no. 1, p. 5, 2023. [Online]. Available: https://doi.org/10.1007/s43926-023-00034-5

[14] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for iot intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, p. 102031, 2020, modeling and Simulation of Fog Computing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X19301625

[15] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for iot networks," in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2019, pp. 256–25 609.

[16] L. Touileb, K. Zekri, A. Bradai, Y. Pousset, and J. C. Point, "A hybrid lstm-autoencoder based approach for network anomaly detection system in iot environments," in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2024, pp. 125–130.

[17] G. F. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Sunnyvale, CA, USA: Insecure, 2009.

[18] T. Ottali, P. Perazzo, and A. Vecchio, "Detection of cyber attacks in cyberphysical systems through energy consumption analysis," Jul. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.16088505