International Journal of Information Security (2025) 24:172
https://doi.org/10.1007/s10207-025-01092-2

REGULAR CONTRIBUTION

CARNYX: A framework for vulnerability detection via power consumption analysis in embedded systems

Jorge Barredo[1,2] · Maialen Eceiza[1] · Jose Luis Flores[3] · Mikel Iturbe[2]

Accepted: 23 June 2025 / Published online: 8 July 2025

Abstract

The widespread use of the Internet of Things (IoT) has led to a surge in interconnected, resource-constrained embedded systems, which are inherently vulnerable due to limited security mechanisms. This paper presents CARNYX, a framework leveraging power consumption analysis, rooted in Side-Channel Analysis (SCA), to detect vulnerabilities in embedded systems with high accuracy. Designed for pre-deployment vulnerability detection, it offers three key advantages over existing SCA solutions: (1) detailed categorisation of specific vulnerability types beyond binary detection, (2) a methodology validated on the STM32F4 architecture and ARM Cortex-A8 with potential applicability to similar low- and medium-end systems, and (3) reliable detection in resource-constrained devices where power monitoring is practical. We evaluate CARNYX on three platforms: two low-end STM32F4-based platforms (Riscure Piñata and STM NUCLEO-144) and the medium-end ARM Cortex-A8-based BeagleBone Black, analysing 16 arithmetic and memory-related software flaws. Results demonstrate recall rates of 99.69% (Piñata), 86.88% (NUCLEO-144 with serial interface), 51.25% (NUCLEO-144 with Ethernet), and 53.67% (BeagleBone Black)-all with high precision-while measuring the effect of communication peripherals on side-channel leakage, an aspect underexplored in prior vulnerability detection studies. These results highlight CARNYX's potential to enhance security in constrained IoT devices, even in noisy environments where binary detection methods offer limited value. While validated on STM32F4 and ARM Cortex-A8, its principles may extend to other low- and medium-end systems, subject to further validation.

Keywords Embedded systems · Side-channel analysis · Hardware security · Vulnerability detection

1 Introduction

The number of Internet of Things (IoT) devices worldwide is forecast to almost double from 13.15 billion in 2023 to more than 25.44 billion IoT devices in 2030 [1]. This growth expands the attack surface due to the diverse applications of IoT devices in various sectors, including industrial, telecommunication, automotive, and medical domains. This makes them attractive targets for cyberattacks, potentially resulting in significant financial losses. Furthermore, inadequate security in IoT devices can lead to severe security consequences, such as data breaches compromising sensitive information [2].

Embedded systems are a fast-growing part of the IoT market, expected to reach $116.2 billion by 2025 [3]. However, securing these compact, single-function devices presents significant challenges. Their limited processing capabilities, memory constraints, and cost-sensitive design hinder the implementation of conventional security solutions [4]. Moreover, traditional security approaches may be ill-suited for these constrained environments [5], exposing devices to vulnerabilities, as highlighted in OWASP IoT Top 10 2018 [6].

✉ Jorge Barredo
jbarredo@ikerlan.es

Maialen Eceiza
meceiza@ikerlan.es

Jose Luis Flores
joseluis.flores@ehu.eus

Mikel Iturbe
miturbe@mondragon.edu

1 IKERLAN Technology Research Centre, Mondragon Unibertsitatea, Arrasate/Mondragón, Gipuzkoa, Spain

2 IKERLAN Technology Research Centre, Arrasate/Mondragón, Gipuzkoa, Spain

3 University of the Basque Country UPV/EHU, Donostia-San Sebastián, Gipuzkoa, Spain

Besides, embedded systems, once deployed, are difficult to monitor for security due to their complex design. This complexity hinders continuous surveillance throughout their lifecycle-from deployment to decommissioning [7], often leaving vulnerabilities undetected until exploitation. Post-deployment remediation can be costly and resource-intensive, while ignoring these issues increases cyberattack risks. Therefore, a proactive approach, emphasising security during development, becomes essential for improved system reliability.

Existing security solutions for embedded systems often prioritise post-deployment monitoring techniques [8–10], generally targeting specific applications or hardware architectures. Furthermore, they utilise conventional forensics techniques to identify limited vulnerability sets. This perspective proves inadequate for the heterogeneous landscape of embedded systems. Consequently, there is a pressing need for non-invasive external analysis techniques capable of identifying security vulnerabilities without disrupting operation. Such techniques would enable security assessment during both pre-deployment and operational phases.

Various methods address embedded security, including source code evaluation (static analysis [11]), runtime analysis (dynamic analysis [12]), and test input generation (fuzzing [13]). Physical evaluation techniques have gained attention recently [14], with Side-Channel Analysis (SCA) emerging as a promising approach. SCA leverages unintentional information leakage through physical channels to detect vulnerabilities [15, 16], showing potential beyond traditional applications in cryptographic analysis [17]. However, existing SCA solutions often evaluate limited device ranges [18–20] or target specific applications [21–23].

SCA research for embedded systems has explored multiple leakage channels, including timing, electromagnetic, and power consumption analysis. Existing power-based studies [21–23] focus on specific applications, limiting their broader applicability, and rely on pre-labelled datasets and threshold-based detection, restricting them to binary categorisation. In response, this paper proposes CARNYX, a framework leveraging power consumption analysis for SCA across low- and medium-end embedded systems. Rather than replacing conventional security methods, CARNYX complements them with a hardware-level perspective that detects vulnerabilities undetectable through software analysis. Evaluated on STM32F4-based platforms and a BeagleBone Black, it demonstrates its effectiveness across diverse configurations in a controlled environment, independent of specific applications. CARNYX advances beyond existing solutions by:

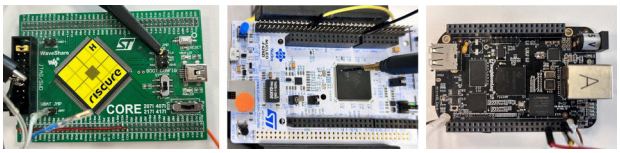

Fig. 1 Evaluated Embedded Systems: Riscure Piñata and STM NUCLEO-144 (low-end, STM32F4-based, left and centre) and BeagleBone Black (medium-end, Linux-based, right)

1. Granular categorisation of specific vulnerability types, based on unsupervised clustering, rather than binary detection.
2. Power monitoring methodology for low- and medium-end embedded systems.
3. High detection accuracy in resource-constrained environments.

Our methodology characterises common software vulnerabilities within three embedded devices: the low-end Riscure Piñata and STM NUCLEO-144, and the medium-end BeagleBone Black, as shown in Figure 1. By capturing, preprocessing, and categorising power responses to modified software execution, CARNYX identifies specific vulnerability classes with high precision. Its automation and standard interfaces enable integration into Continuous Integration/Continuous Deployment (CI/CD) pipelines for continuous security validation. The framework's source code, including three proof-of-concept implementations, is available on GitHub[1].

The remainder of this paper is organised as follows: Section 2 provides background on embedded systems and SCA; Section 3 reviews prior research; Section 4 details CARNYX's design; Section 5 presents results demonstrating high accuracy; Section 6 explores CARNYX's applicability; and Section 7 summarises findings and future directions.

2 Background

This section introduces foundational concepts for CARNYX, covering embedded systems and side-channel analysis (SCA). Section 2.1 explores embedded systems' characteristics and security challenges, while Section 2.2 examines SCA-based security monitoring with emphasis on power consumption analysis for vulnerability detection in resource-constrained environments.

[1] https://github.com/JorgeBarredo14/carnyx

2.1 Embedded Systems

Embedded systems are compact devices that perform a specific task [24], and their hardware and software resources are constrained. Their functionality is defined during their design phase, and once installed, minor or no changes can be made. This necessitates a reactive operational mode, where embedded systems make decisions based on sensor inputs. Common applications for embedded systems involve industrial control systems (e.g., programmable logic controllers, PLCs [25]) and home automation [26].

2.1.1 Taxonomy of Embedded Systems

Embedded systems can be classified according to different principles. As mentioned before, security measures in embedded systems are constrained by their hardware resources. Therefore, following Eceiza *et al.* [27] and Muench *et al.* [28], we categorise embedded systems into high-, medium-, and low-end types by operating system (OS), processing power, and memory. This taxonomy highlights hardware protection mechanisms, such as memory management units (MMUs) and memory protection units (MPUs), which shape vulnerability types.

- **High-end systems:** These feature 32-bit or 64-bit RISC multicore processors or Reconfigurable System-on-a-Chip (RSoC) designs, with at least 1 GB of RAM. They run general-purpose OSs (e.g., Busybox [27]), customised for complex tasks like operator interfaces. Equipped with MMUs for virtual memory management and often MPUs for fine-grained access control, they support hardware protections such as Data Execution Prevention (DEP) and Control Flow Integrity (CFI) [27]. Common vulnerabilities include code injection and memory corruption, mitigated by secure boot [29] and Trusted Execution Environment (TEE)-based security [30].
- **Medium-end systems:** These use 16-bit or 32-bit monocore or bicore microprocessors with 1 MB to 1 GB of RAM, running embedded OSs (e.g., FreeRTOS [27]) optimised for efficiency. Some include MMUs for process isolation, but MPUs are less common, limiting fine-grained memory protection [27]. They are prone to arithmetic and memory-related vulnerabilities, such as buffer overflows [31], mitigated by memory partitioning (where MMUs exist) and stack protection.
- **Low-end systems:** Equipped with 8-bit or 16-bit monocore processors and less than 1 MB of RAM, these typically lack an OS. MMUs are generally absent, though some include MPUs, with hardware protections like DEP or CFI being rare [27]. They face vulnerabilities such as stack/heap overflows and side-channel attacks [21, 23], addressed by software-based bounds checking and memory segment isolation (where MPUs exist).

Table 1 Hardware Protection Mechanisms by Microcontroller Families [27]

Hardware Family	MPU	MMU	DEP	CFI
ARM 1 to ARM 7	×	×	×	×
ARM Cortex R	✓	×	×	×
ARM Cortex M	Partial*	×	✓	×
PIC 10 to PIC 24	×	×	×	×
Intel MCS-51	×	×	×	×
Infineon XC88X-I	×	×	×	×
Infineon XC88X-A	×	×	×	×

* Supported by some microcontrollers in the family

Hardware protection mechanisms, particularly MMUs and MPUs, vary across microcontroller families, influencing the security of high-, medium-, and low-end systems. Table 1 illustrates these differences, highlighting the limited protections in low-end systems critical to CARNYX's application. This analysis underscores the resource-security trade-offs, enabling CARNYX to detect vulnerabilities in resource-constrained systems using power consumption analysis.

2.1.2 Security in Embedded Systems

Embedded systems often operate with constrained hardware resources, especially regarding memory and processing power, which poses a challenge for security implementations. These security solutions usually require additional computational resources, potentially impacting the performance of these devices. For example, cryptographic implementations may consume significant processing resources [4], creating a tradeoff between security and functionality.

Traditionally, development has prioritised core functionality, often overlooking security. This has enabled attackers to exploit specific vulnerabilities in these systems, such as buffer overflows in IoT firmware [32] and cryptographic implementations with side-channel leakage [33]. These vulnerabilities are particularly concerning, as embedded systems often lack runtime protection mechanisms found in general-purpose systems, such as address space layout randomization (ASLR) or stack canaries.

The landscape of embedded systems is transforming significantly. The number of deployed devices is rapidly increasing, as well as their complexity and interconnectivity [34], demanding more comprehensive security approaches. In response, regulatory bodies have introduced legal frameworks, such as the Directive (EU) 2022/2555 (NIS 2 Directive) [35]) and standards for embedded systems, including the Cyber Resilience Act (CRA) [36]. Additionally, technical

standards like ISA/IEC 62443-4-1/2 [37, 38] provide specific security guidelines for industrial automation and control systems.

These regulatory and standardisation efforts define requirements for embedded devices deployed in critical infrastructure, where security breaches can have severe consequences [39]. Non-invasive analysis techniques that can identify vulnerabilities during pre-deployment testing provide a valuable approach to meeting these requirements without adding computational burden to the target system.

2.2 Side-Channel-Based Security Monitoring

We prioritise power consumption analysis over other parameters (e.g., electromagnetic analysis) due to its correlation with hardware activity and accessibility via standard oscilloscopes, particularly in low-end systems where hardware simplicity enables accessible monitoring. Power monitoring in complex devices requires detailed manipulation of their hardware components [16], and other parameters such as electromagnetic analysis exhibit greater variability across devices.

Originally developed to evaluate leakage in cryptographic systems [40], side-channel analysis (SCA) has shown promise for securing IoT devices, particularly in malware detection [45]. Studies by Moore *et al.* [42] further confirm that power consumption signatures vary between valid and faulty executions, with observable differences during errors such as buffer overflow attacks.

In this paper, we leverage power consumption analysis for defensive anomaly detection in embedded systems, aiming to preempt vulnerabilities that could be exploited post-deployment. While this concept is not entirely novel [23, 46, 47], its application to embedded systems offers a useful extension of prior work. Furthermore, although existing studies have advanced the identification of abnormal power patterns, they often lack a comprehensive analysis of the underlying causes of these anomalies, as detailed in Table 2. This work seeks to address that limitation through a granularisation of the detected anomalies. By granularisation, we refer to the framework's ability to distinguish between multiple specific vulnerability types (16 in our evaluation) rather than providing only binary (normal/anomalous) categorisation. This approach enables precise diagnosis and targeted remediation strategies.

The theoretical foundation for power-based vulnerability detection rests on three key principles: (1) execution flow determinism in low-end embedded systems, (2) vulnerability-specific instruction pattern alterations, and (3) power consumption correlation with instruction execution patterns. Low-end embedded systems characteristically execute instructions in deterministic sequences with minimal background processing, creating consistent power signatures. When vulnerabilities are exploited, these sequences are measurably altered by specific instruction pattern changes-arithmetic vulnerabilities typically affect ALU (Arithmetic Logic Unit) operations, while memory vulnerabilities disrupt memory access patterns. Each instruction type (e.g., memory load/store, arithmetic operations) consumes distinctly different power amounts based on circuit activation patterns. This relationship between instruction types, vulnerability exploitation, and power consumption enables the identification of specific vulnerability types through power signature analysis. This foundation explains why we observe different power consumption clusters for different vulnerability categories, as detailed in our experimental results.

3 Related Works

The field of anomaly detection in embedded systems has advanced significantly, with methodologies varying by detection parameters, system architectures, and analytical techniques. Table 3 summarises key contributions, revealing gaps that CARNYX addresses through its granular vulnerability detection in embedded systems.

Studies such as Qaddori [21] detect anomalies in smart meter data but provide only general deviations, lacking specific vulnerability insights. WattsUpDoc [18] pioneered power-based malware detection, achieving 85% accuracy on medical devices using supervised learning. However, its reliance on labelled datasets limits real-world applicability. Similarly, Ding *et al.* [44] and Wang *et al.* [41] employed deep learning for IoT devices and PLCs, achieving 92.7% and 91.28% accuracy, respectively. Bai *et al.* [48] achieved 96% accuracy on Arduino platforms with real-time detection, whilst TrustGuard [43] reached 99% on BeagleBone systems using custom FPGA hardware. However, these approaches rely on binary 'normal' versus 'anomalous' categorisation, which lacks actionable information about vulnerability types.

Despite their high accuracy, current methods face fundamental limitations that constrain their diagnostic capabilities. Supervised approaches like WattsUpDoc [18] and DeepPower [44] depend on pre-labelled datasets, which are typically binary due to the complexity of labelling multiple vulnerability types. Deep learning methods [41, 48] achieve high accuracy but obscure feature interpretation, hindering differentiation between anomaly types, while their computational demands exceed resource-constrained environments. Moreover, threshold-based approaches like TrustGuard [43] produce binary decisions that limit detailed vulnerability insights for targeted remediation.

CARNYX addresses these limitations by introducing granular vulnerability categorisation through unsupervised clustering, identifying 16 distinct vulnerability types from power consumption patterns without requiring pre-labelled

Table 2 Methodological Comparison of Side-Channel Analysis Approaches

Feature	Traditional SCA	Threshold Anomaly Detection	Supervised SCA	Unsupervised SCA
Methodology	DPA, CPA attacks [40]	Threshold detection [18]	Classifier detection [41]	Density-based clustering
Categorisation	Binary (leak/secure)	Binary (normal/anomaly)	Fixed anomaly classes	Granular
Use Case	Cryptanalysis	Runtime monitoring	Application detection	Pre-deployment testing
Input Data	Reference traces	Baseline traces	Labeled datasets	Unlabeled traces
Compute Load	High (alignment, stats)	Low-moderate (threshold)	High (training)	Moderate (preprocessing)
Frameworks	CEMA [33], CPA [42]	WattsUpDoc [18], TrustGuard [43]	DeepPower [44], DDCM [41]	CARNYX (this work)

Table 3 Comparison of related works and CARNYX

Method	Parameter	Device Under Test	Categorisation Approach
WattsUpDoc [18]	Power Consumption	Schweitzer SEL3354	Binary (normal/anomalous)
Ding *et al.* [44]	Power Consumption	D-Link, Xiaofang IP Cameras	Binary (normal/anomalous)
Wang *et al.* [41]	Power Consumption	Arduino Mega 2560, Siemens PLC	Binary (normal/anomalous)
Bai *et al.* [48]	Power Consumption	Arduino UNO	Binary (normal/anomalous)
TrustGuard [43]	Power Consumption	BeagleBone Black	Binary (normal/anomalous)
CARNYX	Power Consumption	Riscure Piñata, STM NUCLEO-144, BeagleBone Black	Granular (specific vulnerability types)

data. This eliminates the dependency on labelled datasets while providing actionable insights for targeted security remediation. When evaluated on the BeagleBone Black-the same hardware used by TrustGuard [43]-CARNYX achieves 53.67% recall with detailed vulnerability categorisation despite OS noise.

Last but not least, in contrast to existing frameworks that operate as standalone solutions, CARNYX is designed to complement rather than compete with software-based security techniques such as fuzzing, static analysis, or dynamic analysis, providing hardware-level insights through power consumption analysis. CARNYX's effectiveness across diverse platforms-from STM32F4-based systems to Linux-based BeagleBone Black-demonstrates its broad applicability, advancing embedded system security.

4 CARNYX Framework

To address the growing threat of side-channel attacks, this paper proposes CARNYX, a framework designed to enhance the physical security of embedded systems during their development phase through power consumption analysis. Tailored for low- and medium-end embedded devices, CARNYX operates non-invasively and independently of specific hardware architectures or software, an advantage rooted in its modular design and validated in Section 5. This section outlines CARNYX's conceptual overview 4.1, practical implementation 4.2, and operational mode 4.3, emphasizing its modularity and pre-deployment focus. It comprises three core modules—Data Acquisition, Data Preprocessing, and Anomaly Detection—which process software inputs through the device under test (DUT) to produce categorised vulnerability insights, as illustrated in the abstract architecture (Figure 2). These modules function across two phases—calibration to establish normal behaviour and operational to detect anomalies—ensuring adaptability across diverse low- and medium-end embedded systems.

4.1 Overview

CARNYX facilitates pre-deployment security evaluation by monitoring power consumption patterns, offering a non-invasive approach independent of the DUT's architecture or software specifics, as detailed in Section 4.2. Its general architecture, depicted in Figure 2, organises functionality into three conceptual modules:

1. **Data Acquisition:** Captures physical signals from the DUT using measurement equipment during software execution.
2. **Data Preprocessing:** Enhances signal quality and reduces data dimensionality for efficient analysis.
3. **Anomaly Detection:** Analyses preprocessed signals to identify and categorise vulnerability-specific patterns.

These modules form a dataflow—from software inputs to power responses and ultimately to vulnerability categories—that remains flexible across testing environments. CARNYX operates in two phases: a calibration phase to establish a

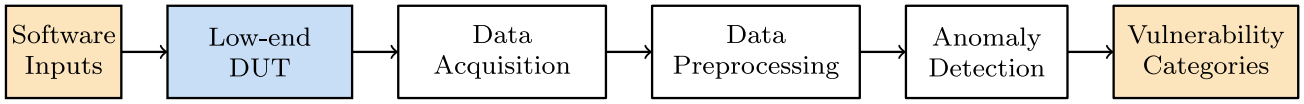

Fig. 2 Abstract architecture of CARNYX.

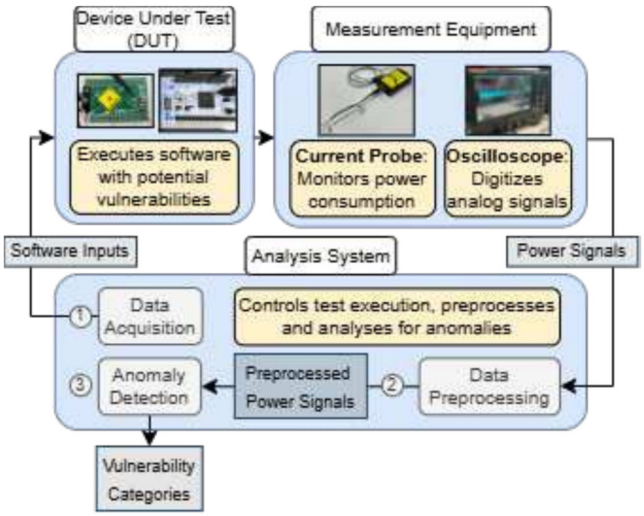

Fig. 3 Practical implementation of CARNYX.

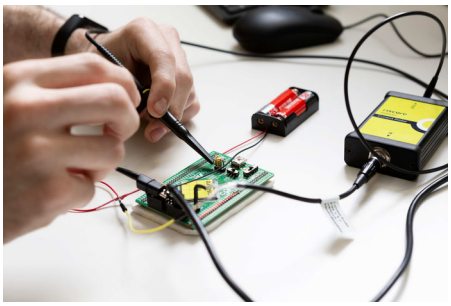

Fig. 4 Setup of the Riscure Piñata with the current probe for power trace acquisition.

baseline of normal behaviour and an operational phase to compare new executions against it, enabling broad applicability to low- and medium-end embedded systems with predictable power signatures, as explored in Section 4.3. Leveraging the non-invasive nature of power analysis, it targets pre-deployment testing, with potential for future runtime adaptations.

4.2 Architecture

This subsection details the practical implementation of CARNYX's conceptual framework, as evaluated in our experiments, specifying the hardware and software components underpinning its functionality. Illustrated in Figure 3, the architecture comprises the DUT, measurement equipment (Riscure current probe and LeCroy WaveRunner 8104 oscilloscope), and an analysis system hosting three modules-Data Acquisition, Data Preprocessing, and Anomaly Detection. These components operate sequentially to enable non-invasive vulnerability detection in low-end embedded systems.

To operationalize this setup, the modules interact with the DUT and measurement equipment in a structured sequence: power consumption data capture, followed by preprocessing and anomaly detection. Subsequent subsections elaborate each module's role, beginning with Data Acquisition, which initiates the workflow by collecting raw power traces from the DUT, as demonstrated in the Riscure Piñata setup (Figure 4).

4.2.1 Data Acquisition Module

The Data Acquisition module acts as the initial point of interaction with the DUT. It is responsible for extracting behavioural signatures by triggering specific software execution within the DUT and capturing the corresponding power consumption responses. This process is versatile and adapts to the control architecture of each DUT. For instance, the approach differs depending on the device's available input methods (e.g., serial or Ethernet port). If the DUT utilizes a serial port, the module transmits commands through it to order the execution of the designated software.

In the beginning, there is no reference to distinguish normal system behaviour from anomalous power responses. CARNYX addresses this challenge by establishing a reference program that represents the DUT's power consumption patterns during normal operation. Moreover, the framework's ability to detect anomalies remains independent of this reference program's functionality.

Once a program is executed in the DUT, its power consumption response is captured by a current probe connected to an oscilloscope (Figure 4). The oscilloscope's sample rate adheres to the Nyquist-Shannon sampling theorem [49], requiring a minimum of twice the highest frequency component for reliable signal reconstruction. While this sets the baseline, empirical studies in signal processing [50, 51] recommend oversampling at 5x to 20x for optimal signal quality versus computational cost. We chose up to 10x (e.g., 1 GS/s for a 168 MHz DUT) to balance: (1) resolution for capturing subtle instruction-level power variations, (2) computational feasibility for trace processing, and (3) storage efficiency,

avoiding excessive demands from higher rates. Preliminary tests confirmed that exceeding 10x offered minimal accuracy gains while significantly increasing processing time and storage needs (Section 5).

Furthermore, power analysis provides useful insights into embedded system behaviour, particularly in low-end systems where hardware simplicity minimizes variability in captured responses, and can also be extended to medium-end systems despite increased system complexity. Unlike complex devices, where intricate hardware interactions cause fluctuating power patterns, low-end systems produce stable execution times and signatures tied to the executed program. Although minor variability may arise from inherent noise or experimental conditions, the preprocessing filters it, ensuring minimal impact on clustering performance-required for CARNYX's anomaly detection. This stability enhances the reliability of power analysis in our framework.

4.2.2 Data Preprocessing Module

Following data acquisition, CARNYX processes a large set of power consumption responses, each with numerous samples, posing a computational challenge for anomaly detection. To address this, the preprocessing reduces data volume while preserving critical information. Additionally, our experiments conducted in controlled environments with consistent noise conditions demonstrate that this methodology effectively mitigates noise. It is also expected to handle varying real-world noise patterns with similar effectiveness, ensuring reliable clustering results.

This preprocessing involves two steps: noise filtering and dimensionality reduction. Noise filtering denoises traces using the interquartile range (IQR) technique [52], identifying outliers —samples deviating significantly from the expected range— and replacing them with linear interpolation. IQR was chosen over other methods like wavelets–requiring frequency tuning–or Kalman filters–needing state-space models. Moving average filters were also considered, but tend to blur sharp transitions in power traces that often indicate vulnerability exploitation. In contrast, IQR provides linear complexity without parameter tuning.

Tests on embedded systems traces [53] show IQR offers fast execution and robust noise reduction across hardware platforms. Its quartile-based approach adapts to signal changes without recalibration, facilitating long-term operation, where noise characteristics might evolve over time. However, while improving traces clarity, their size remains high, requiring further processing.

For dimensionality reduction, Principal Component Analysis (PCA) is applied. Other methods, like t-SNE and UMAP, were tested but are computationally intensive and produce inconsistent results due to their stochastic nature, potentially affecting clustering reliability. In contrast, PCA provides deterministic results with lower computational demands, making it more suitable for resource-constrained embedded systems. While t-SNE occasionally visualises clusters more clearly, these improvements rarely enhance anomaly detection and consistently increase processing time. PCA's preservation of global variance better supports the detection of execution pattern differences. With a computational efficiency of $O(n^2)$ compared to $O(n^3)$ for t-SNE or UMAP, and retention of 99% variance with BIC-selected components [54], PCA enables efficient and effective processing for anomaly detection.

For low- and medium-end embedded systems, which exhibit simpler power consumption patterns, information-based methods provide a data-driven way to identify the most important features using fewer components. One such method, CARNYX, relies on the Bayesian Information Criterion (BIC). BIC strikes a balance between model complexity (i.e., the number of components) and the ability to explain the data's variance. Compared to other criteria, like the Akaike Information Criterion (AIC) [55], BIC places a stronger penalty on complexity, leading to fewer components per trace. This makes BIC especially well-suited for low-dimensional anomaly detection tasks, such as those handled by CARNYX.

As a result, noise is successfully mitigated, enhancing the fidelity of the traces, and PCA reduces their size to a manageable level. The framework can then analyse preprocessed traces for anomaly detection, enabling the identification of potential underlying security issues within the DUT.

4.2.3 Anomaly Detection Module

If a DUT has a vulnerability exploited by the modified software, a fault may emerge. CARNYX relies on a baseline of normal power consumption behaviour, flagging any deviations as anomalies. Once preprocessed, traces are clustered to reveal distinct execution flows within the DUT.

CARNYX uses unsupervised clustering to group similar power consumption responses without predefined labels [56], applicable for anomaly detection from solely power data-unlike supervised methods needing known classes or semi-supervised approaches needing pre-labeled data. Unsupervised clustering identifies anomalies without prior knowledge of vulnerability-related traces-a key requirement for security assessments. Density-based categorisation aligns with power analysis, as similar execution patterns naturally form dense regions in the feature space. We chose HDBSCAN[57] over alternatives like DBSCAN or OPTICS [58] for its superior ability to handle varying cluster densities and noise in high-dimensional data.

After each set of executions in the operational phase, clustering produces between two and n distinct groups. Although we, as researchers, know the bug linked to each program-

providing ground truth for evaluation-the system conducts unsupervised clustering without this knowledge. This design enables CARNYX to independently detect anomalies during real-world deployment. During development testing, our ground truth facilitates validation using confusion matrices and standard metrics. However, such evaluation is not feasible when applying this clustering beyond development or to a different device.

Confusion matrices assess clustering performance by comparing ground truth (vertical axis) and algorithm-assigned (horizontal axis) bug categories. Each cell (*i*,*j*) counts traces from ground truth category *i* assigned to category *j*. Diagonal elements (*i*=*j*) represent correct categorisations, with higher counts indicating better accuracy, while off-diagonal elements reveal three issues: false negatives (error traces categorised as normal), false positives (traces assigned to incorrect error clusters), and Unassigned Anomalies (UAN)-anomalous traces not assigned to specific error categories due to insufficient distinguishing characteristics. We analyse three scenarios to assess the clustering algorithm's effectiveness using this matrix:

- **Anomaly Detection:** Differentiates between calibration and faulty traces. Any trace deviating from calibration clusters is flagged as anomalous, based on the premise that exploited vulnerabilities produce unique power consumption patterns.
- **Arithmetic vs. Memory Error Detection:** Evaluates CARNYX's ability to distinguish between our taxonomy's two primary error categories. The confusion matrix data allows calculation of clustering recall for each error-type category and creation of higher-level "arithmetic" and "memory" clusters.
- **Specific Error Detection:** Assesses identification of individual error types. Diagonal elements (true positives) represent correct error-type assignments, while off-diagonal elements indicate either false negatives (faulty traces assigned to calibration groups) or false positives (traces assigned to incorrect error types). This evaluation measures CARNYX's precision in identifying specific vulnerabilities.

The effectiveness of clustering in each scenario is quantified using five metrics commonly established in state-of-the-art anomaly detection research [59]:

1. **Recall:** Measures the proportion of real positives correctly identified. TP and FN represent the number of true positives and false negatives, respectively.

$$rec = \frac{TP}{TP + FN} \tag{1}$$

2. **Precision:** Measures the proportion of positive predictions that are correct. TP and FP represent true positives and false positives, respectively.

$$prec = \frac{TP}{TP + FP} \tag{2}$$

3. F_1 **score:** The harmonic mean of precision and recall.

$$F_1 = \frac{2 \cdot prec \cdot rec}{prec + rec} \tag{3}$$

4. **Micro-F_1 score:** A standard metric for evaluating global clustering performance across all classes [60].

$$\text{Micro-}F_1 = \frac{2 \sum TP}{\sum TP + \sum FP + \sum FN} \tag{4}$$

5. **Matthews Correlation Coefficient (MCC):** Commonly used in contemporary security research for imbalanced datasets [61] as it considers all confusion matrix elements.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

4.3 Operational Mode

The framework operates in two phases to overcome the lack of an initial reference for anomaly detection. In the calibration phase, CARNYX establishes a baseline of normal power behaviour using repeated executions of a reference program. In the operational phase, it collects power responses from executions of the modified software, clustering them to distinguish faulty from calibration executions. This approach enables CARNYX to detect and categorise vulnerabilities in low-end embedded systems during development.

4.3.1 Calibration Phase

A challenge in anomaly detection is the lack of a reference for accurate categorisation, which CARNYX's calibration phase addresses by establishing a baseline of normal power consumption behaviour. Calibration is unique to each DUT and baseline program, reflecting variability across programs. It executes a known baseline program (see Section 4.2) on the DUT multiple times to account for variations in power responses due to background tasks. This phase also collects idle signals to detect faults causing no power response, forming a reference set with baseline traces for anomaly detection.

In our evaluations, we used 200 calibration traces per DUT (100 baseline + 100 idle), requiring approximately 5 minutes to complete. We determined this number through statistical analysis of trace variance: preliminary tests showed

that variance in power consumption patterns stabilized below 5% after approximately 80 traces, with diminishing returns beyond this point. This finding aligns with research by Tamura *et al.* [62] on sample size requirements for power trace stability. The 100-trace threshold per category thus provides a comfortable margin for statistical reliability while maintaining practical collection time. For systems with greater variability in background processing, this number could be adjusted upward based on variance stabilisation monitoring during initial calibration.

Another important factor that can affect calibration reliability is the variability of the power supply. While our battery-powered DUT (Riscure Piñata) exhibited stable power signatures across different charge levels, other battery-powered devices may show supply-dependent fluctuations. To account for this, CARNYX supports an optional dual-phase calibration protocol-performed at full charge and during linear discharge-to capture potential deviations. This strategy improves trace consistency without altering the core preprocessing or detection flow, and enhances the framework's adaptability to diverse power environments.

4.3.2 Operational Phase

The operational phase leverages the calibration phase's reference set to identify anomalies in the DUT. In real-world practice, each new input is collected, preprocessed, and categorised using the calibrated model in 5–10 seconds. For our experiments, we gathered 360 operational power traces, consisting of 20 executions per vulnerability from Table 4, plus 20 instances each of the calibration program's integer (SUT00I) and floating-point (SUT00F) variants. We preprocessed all traces, trained HDBSCAN algorithm using calibration traces, and then categorised the operational traces into clusters. We selected 20 traces per vulnerability to balance detection sensitivity with experimental efficiency, leveraging the stability of the calibration baseline. Traces not assigned to a calibration cluster are marked as anomalies.

5 Experimental Results

This section explores the applicability of CARNYX for vulnerability detection across various low-end embedded systems through experimentation. The experiment establishes a baseline program and a set of embedded system vulnerabilities. We evaluate CARNYX's performance across three scenarios from Section 4: 1) Anomaly Detection; 2) Arithmetic vs. Memory Error Detection; and 3) Specific Error Detection. These scenarios are tested on three platforms: two STM32F4-based systems (Riscure Piñata and STM NUCLEO-144) and the BeagleBone Black, a medium-end system with a Linux-based operating system. This evaluation demonstrates power consumption's effectiveness in low-end embedded systems, where hardware simplicity enhances signal detection, while also exploring its applicability to more complex medium-end systems with operating system overhead.

5.1 Establishment of a Baseline Program

For our evaluation, we established a baseline program to serve as a reference for normal behaviour of the Device Under Test (DUT)-the embedded system being evaluated-independent of its specific functionality. We created two variants of this program: one using integer arithmetic (SUT00I) and another using floating-point arithmetic (SUT00F), both solving the same equations but through different computational methods. This approach allowed us to test CARNYX's sensitivity in distinguishing between functionally identical programs on the DUT that utilize distinct underlying operations-a critical capability for accurate vulnerability detection in embedded systems.

5.2 Taxonomy of Weaknesses in Embedded Systems

To evaluate our framework's effectiveness at detecting security vulnerabilities in low-end embedded systems, we first establish a taxonomy of common weaknesses, providing a structured approach to test CARNYX across diverse vulnerability types.

Integrating embedded systems with the physical world poses a challenge for vulnerability detection using SCA. Software faults in these systems can manifest in diverse physical responses, with some faults potentially causing similar responses for different inputs. Consequently, a comprehensive evaluation of SCA requires a diverse set of vulnerabilities to maximize coverage. The goal is to identify a broad range of vulnerability categories for a thorough assessment.

We propose a classification for vulnerabilities, divided into two main categories: arithmetic and memory issues. Arithmetic faults arise during arithmetic instructions, whereas memory-related problems emerge during memory management operations. These vulnerabilities, listed in Table 4, correspond to categories from the Common Weakness Enumeration (CWE) [63], a community-developed list of software and hardware weakness types, to ensure thorough coverage in SCA. We selected six arithmetic and ten memory-related weaknesses common in low-end embedded systems.

Starting from a calibration baseline program, we deliberately modified it to introduce each vulnerability in the taxonomy, creating sixteen variants-each exhibiting a specific weakness-and programmed these into the DUTs. Additionally, both DUTs include a custom C-coded program to

Table 4 Weaknesses employed to characterise the behaviour of embedded systems

Type	Code	Name	Definition	Related CWE Code
Arithmetic	01-01	Floating Point Overflow	Occurs when, performing floating point operations, the generated float value is over the range accepted.	CWE-681
	01-02	Floating Point Underflow	Occurs when, performing floating point operations, the generated float value is under the range accepted.	CWE-681
	01-03	Integer Overflow	Occurs when, performing arithmetic operations, the generated integer value is over the range accepted by the integer variable.	CWE-190
	01-04	Integer Underflow	Occurs when, performing arithmetic operations, the generated integer value is under the range accepted by the integer variable.	CWE-191
	01-05	Divide by Zero Integer	Occurs when trying to divide an integer variable by zero.	CWE-369
	01-06	Divide by Zero Decimal	Occurs when trying to divide a decimal variable by zero.	CWE-369
Memory	02-01	Segmentation Fault	Occurs when the program tries to access a part of the memory which it does not have access to.	CWE-119
	02-02	Buffer Overflow	Occurs when a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.	CWE-120
	02-03	Double Free	Occurs when the program tries to free a memory region that has already been freed.	CWE-415
	02-04	Null Pointer Dereference	Occurs when a pointer with a value of null is used as though it pointed to a valid memory area.	CWE-476
	02-05	Out-of-Bounds Write	Occurs when the program writes outside the buffer.	CWE-787
	02-06	Out-of-Bounds Read	Occurs when the program reads out of the buffer.	CWE-125
	02-07	Out-of-Memory	Occurs when there is an attempt to access out of memory, because there is no free memory.	CWE-400
	02-08	Stack Overflow	Occurs when the program reads out over the space of the stack.	CWE-121
	02-09	Stack Underflow	Occurs when the program reads out under the space of the stack.	CWE-124
	02-10	Unaligned Address	Occurs when the address is not a multiple of the transfer size.	CWE-188

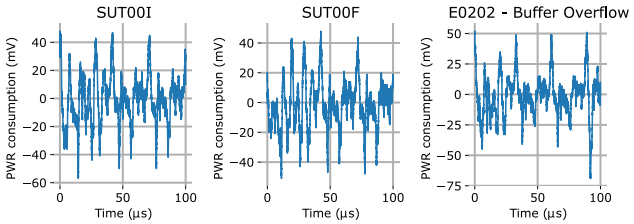

Fig. 5 Power consumption traces from Riscure Piñata over a 100-microsecond window. Calibration responses (SUT00I and SUT00F) exhibit subtle time-domain differences, while a buffer overflow (E0202) produces a distinct power signature.

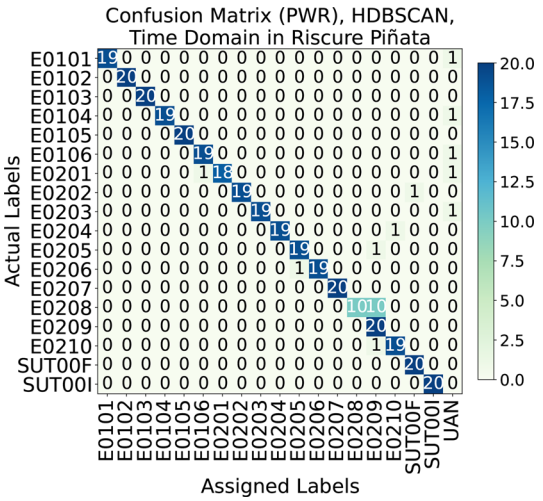

Fig. 6 Riscure Piñata: Metrics for Anomaly Detection (left) and Arithmetic vs. Memory Error Detection (right)

synchronize with measurement equipment, generating digital output signals to mark the start and end of each execution.

5.3 DUT 1: Riscure Piñata

The STM32F4 architecture was selected for its prevalence in industrial and IoT applications and significant market share [64]. The Riscure Piñata and STM NUCLEO-144 platforms serve as complementary testbeds: Piñata provides a controlled environment, while NUCLEO-144 mirrors real-world conditions.

The Riscure Piñata, an experimental device for SCA testing, is part of the STM32F4 family. It features a 32-bit ARM Cortex-M4 monocore with a base clock speed of 168 MHz. Operating without an OS, it receives data via a serial port and offers a highly controlled setup for power analysis, as its hardware omits decoupling components that could consume extra power and mask signals. Upon powering on, Piñata waits for execution requests, runs programs at 168 MHz, and then returns to standby.

Power consumption traces were collected as described in Section 4.3, captured by a Riscure current probe and transmitted to the oscilloscope, whose sampling rate is 1 GS/s, approximately ten times the DUT's clock frequency. The traces represent a time window of 100 microseconds in 25,000 samples (Figure 5).

In our experiments, we sequentially collected all 560 power traces (200 calibration traces and 360 operational traces). Preprocessing required 57.18 seconds (0.10 seconds per trace), while clustering took 31.28 seconds (0.06 seconds per trace). In operational settings, each trace is processed individually using a pre-calibrated model, taking 5–10 seconds for complete processing. This process utilized 150 MB of system memory, indicating a reasonable resource overhead. A proof of concept illustrating power analysis on the Riscure Piñata is included in our GitHub repository[2].

Traces not assigned to a calibration cluster are categorised as anomalies. In this development-phase scenario, we know each trace's ground truth category, allowing validation via a confusion matrix (Figure 7).

The confusion matrix reveals two distinct calibration clusters-integer and floating-point arithmetic-demonstrating CARNYX's ability to differentiate functionally similar programs. Low error rates in calibration groups (0.31%), unassigned anomalies (1.56%), and false positives (1.56%) confirm reliable anomaly detection on this device. Results from the three scenarios are analysed as follows:

1. **Anomaly Detection.** CARNYX demonstrates successful anomaly detection on the Piñata, achieving 100% precision and 99.69% recall in separating faulty from valid signals (Table 5, Figure 6). These metrics align with expectations for the Piñata's simple hardware.
2. **Arithmetic vs. Memory Error Detection.** CARNYX discriminates arithmetic and memory errors with recalls of 97.50% and 99.00%, respectively (Figure 6). These results, alongside the other evaluation metrics, confirm strong clustering performance.
3. **Specific Error Detection.** Specific error detection exceeds 95.00% for most weaknesses, with diagonal confusion matrix elements indicating high ground truth alignment. However, E0208 (Stack Overflow) shows two patterns: one akin to E0209 (Stack Underflow), another unique, diverging from others.

5.4 DUT 2: STM NUCLEO-144

STM NUCLEO-144 is a development board based on STM32F4 architecture. It operates without a full OS but can execute user code with basic resource management capabilities. It features a 32-bit ARM Cortex-M4 single-core processor clocked at 180 MHz, and shares a similar workflow with Piñata: it waits for an execution request. However,

[2] https://github.com/JorgeBarredo14/carnyx

Table 5 Clustering Results for Riscure Piñata

Anomaly Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Error	99.69	100.00	99.84		
No Error	100.00	97.56	98.77	99.72	98.61
Arith. vs. Memory Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Arithmetic Error	97.50	99.15	98.31		
Memory Error	99.00	98.51	98.75	98.61	97.56
Specific Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
E0101	95.00	100.00	97.44		
E0102	100.00	100.00	100.00		
E0103	100.00	100.00	100.00		
E0104	95.00	100.00	97.44		
E0105	100.00	100.00	100.00		
E0106	95.00	95.00	95.00		
E0201	90.00	100.00	94.74		
E0202	95.00	100.00	97.44	94.17	93.94
E0203	95.00	100.00	97.44		
E0204	95.00	100.00	97.44		
E0205	95.00	95.00	95.00		
E0206	95.00	100.00	97.44		
E0207	100.00	100.00	100.00		
E0208	50.00	100.00	66.67		
E0209	100.00	62.50	76.92		
E0210	95.00	95.00	95.00		

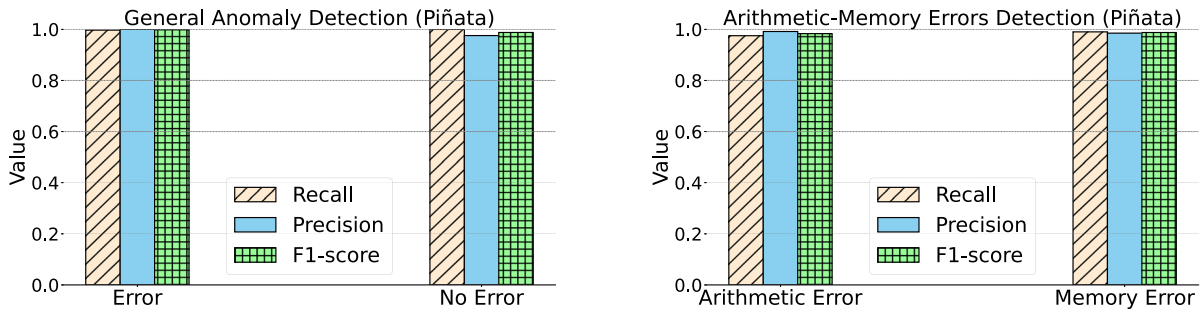

Fig. 7 Riscure Piñata: Confusion Matrix.

unlike Piñata, the NUCLEO-144 offers multiple input ports for data transmission. This allows us to analyse how the selection of the input port impacts power responses. To explore this, we propose two use cases: one transmitting data through the serial port and another through the Ethernet port.

Power consumption traces were collected as outlined in Section 4.3, captured by a current probe and transmitted to the oscilloscope with a 1 GS/s sampling rate.

5.4.1 Use case 1: Input transmission through serial port

In this use case, the collected power consumption traces represent a time window of 50 microseconds in 125,000 samples (Figure 8). The signals exhibit a high degree of noise compared to Piñata due to the STM's complex hardware with additional power-consuming components. As anticipated, this complexity hinders clustering and potentially reduces precision.

In our experiments, all 560 traces (200 calibration [100 baseline + 100 idle] + 360 operational) were collected sequentially. Preprocessing took 83.30 seconds and clustering (training plus anomaly detection) took 6.50 seconds, averaging approximately 0.15 seconds and 0.01 seconds per trace, respectively. A proof of concept illustrating power consumption analysis on this use case, with signals differing from those evaluated here, is included in our GitHub repos-

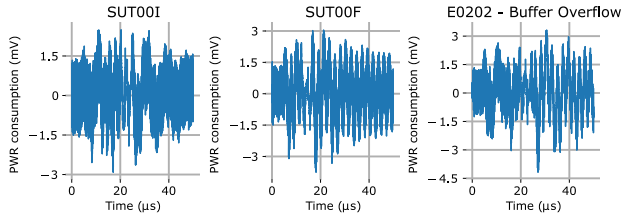

Fig. 8 Power consumption traces from STM NUCLEO-144 via serial port over a 50-microsecond window. Calibration responses (SUT00I and SUT00F) show stable patterns, while a buffer overflow (E0202) induces distinct peaks.

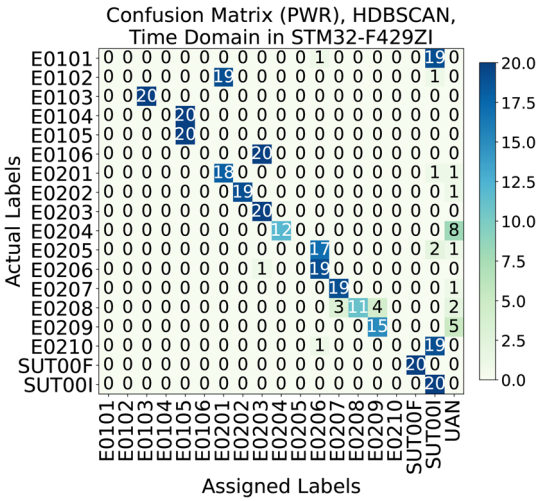

Fig. 9 STM NUCLEO-144 (Serial): Confusion Matrix.

itory[3]. In a real-world operational scenario, each trace is processed using a pre-trained HDBSCAN model in 5-10 seconds. This required 200MB of memory, reflecting increased complexity.

Moreover, we know which ground truth error is related to each power response, as in the previous use case, because the DUT is being analysed in its development phase. Therefore, we can represent the results in a confusion matrix (Figure 9).

The confusion matrix shows a 12.81% recall drop for CARNYX compared to Piñata, with higher false negatives (13.13%), false positives (26.88%), and UANs (5.31%). It still distinguishes integer and floating-point clusters, but E0101 and E0210 often go undetected, suggesting low susceptibility or weak power responses. The three scenarios, detailed in Table 6, further assess its anomaly detection capacity:

1. **Anomaly Detection.** CARNYX achieves 86.88% recall, lower than Piñata's (Figure 10), with a 34.62% MCC drop, reflecting increased complexity's impact.
2. **Arithmetic vs. Memory Error Detection.** Arithmetic faults affect only the current task, while memory errors impact broader processes, suggesting stronger power response changes. Recall is 50% (arithmetic) and 79.50% (memory), with precision at 100.00% and 79.90%, respectively (Figure 10), confirming reliability.
3. **Specific Error Detection.** High false positives (26.88%) complicate clustering (Figure 9). E0101 and E0210 are often undetected, but CARNYX identifies E0103, E0202, E0204, and 55% of E0208 in distinct groups, despite challenges.

Anomaly detection in this scenario confirms the influence of hardware configuration on clustering performance. The observed reduction in metrics likely stems from the activity of additional components not present in simpler devices, such as a more complex memory and processor management unit.

5.4.2 Use case 2: Input transmission through Ethernet

In this use case, the software under test remains the same as in the serial port scenario, but inputs are received through the Ethernet port. The power consumption signals span 100 microseconds across 187,500 samples (Figure 11). These traces exhibit noticeably higher noise levels compared to the serial port case, likely due to the Ethernet port's additional power demands. This port maintains an active interface for incoming frames and connection management messages, activity that introduces a challenging environment for power-based anomaly detection-a scenario common in real-world deployments but rarely quantified in existing literature. The increased noise, stemming from the Ethernet controller's continuous packet management, influences side-channel leakage in low-end systems, as further explored in Section 6.

In our experiments, all 560 traces (200 calibration [100 baseline + 100 idle] + 360 operational) were collected sequentially. Preprocessing took 130.76 seconds and clustering (training plus operational anomaly detection) took 6.21 seconds, averaging approximately 0.23 seconds and 0.01 seconds per trace, respectively. In a real-world operational scenario, each trace is processed using a pre-trained HDBSCAN model in 5-10 seconds. This required 250MB of memory due to larger trace sizes. A proof of concept illustrating power consumption analysis on the STM NUCLEO-144 using the Ethernet port, with signals differing from those evaluated here, is included in our GitHub repository[4]. We analysed the generated clusters and created a confusion matrix (Figure 13) to represent the relationship between ground truth categories and labels assigned by CARNYX.

The confusion matrix reveals the significant challenges of power analysis in the presence of active communication peripherals-an important finding not quantified in

[3] https://github.com/JorgeBarredo14/carnyx

[4] https://github.com/JorgeBarredo14/carnyx

Table 6 Clustering Results for STM NUCLEO-144 (Serial)

Anomaly Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Error	86.88	100.00	92.98		
No Error	100.00	48.78	65.57	88.33	65.10
Arith. vs. Memory Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC(%)
Arithmetic Error	50.00	100.00	66.67		
Memory Error	79.50	79.90	79.70	71.94	56.23
Specific Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
E0101	0.00	0.00	0.00		
E0102	0.00	0.00	0.00		
E0103	100.00	100.00	100.00		
E0104	0.00	0.00	0.00		
E0105	100.00	50.00	66.67		
E0106	0.00	0.00	0.00		
E0201	90.00	48.65	63.16		
E0202	95.00	100.00	97.44	59.20	58.37
E0203	100.00	48.78	65.57		
E0204	60.00	100.00	75.00		
E0205	0.00	0.00	0.00		
E0206	95.00	50.00	65.52		
E0207	95.00	86.36	90.48		
E0208	55.00	100.00	70.97		
E0209	75.00	78.95	76.92		
E0210	0.00	0.00	0.00		

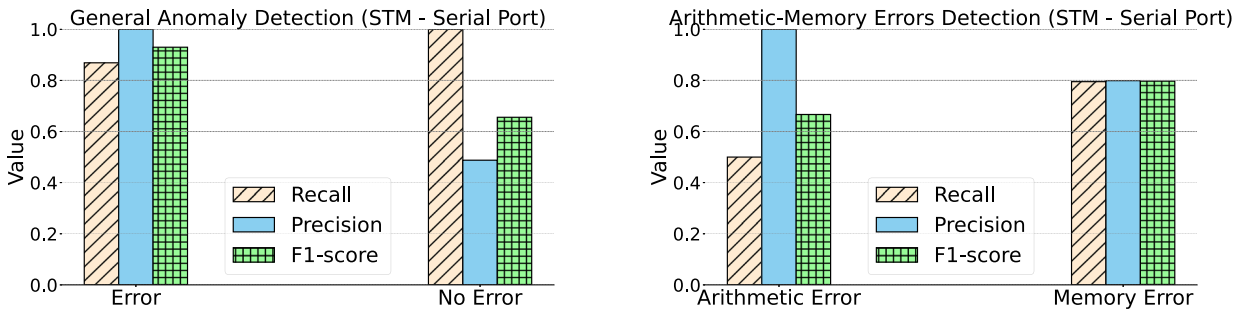

Fig. 10 STM NUCLEO-144 (Serial): Metrics for Anomaly Detection (left) and Arithmetic vs. Memory Error Detection (right).

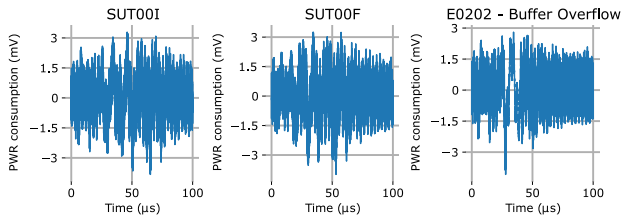

Fig. 11 Calibration and Faulty Signals from STM NUCLEO-144 (Ethernet). Calibration responses (SUT00I and SUT00F) are similar, but a buffer overflow issue triggers a unique waveform.

previous SCA research. Anomaly detection recall drops to 51.25% while maintaining 99.40% precision, with significant increases in false negatives (48.75%) and UANs (9.69%). While this recall might appear modest in isolation, it represents a substantial improvement over the 6.25% expected from random categorisation across 16 vulnerability categories. This demonstrates that even in high-noise environments, CARNYX still captures vulnerability-specific power signatures that would otherwise remain undetected. As this use case differs from the previous one only in port usage, these results specifically isolate the Ethernet controller's impact on power consumption patterns, providing a quantitative measurement of communication peripherals' interference with SCA.

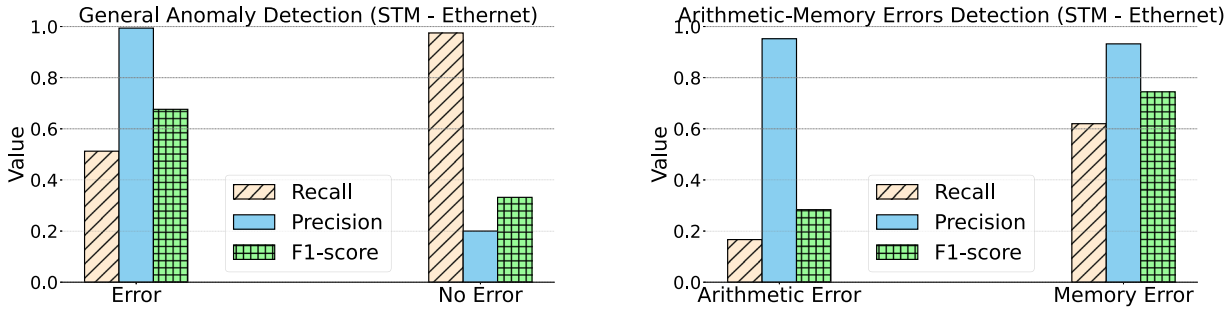

Fig. 12 STM NUCLEO-144 (Ethernet): Metrics for Anomaly Detection (left) and Arithmetic vs. Memory Error Detection (right).

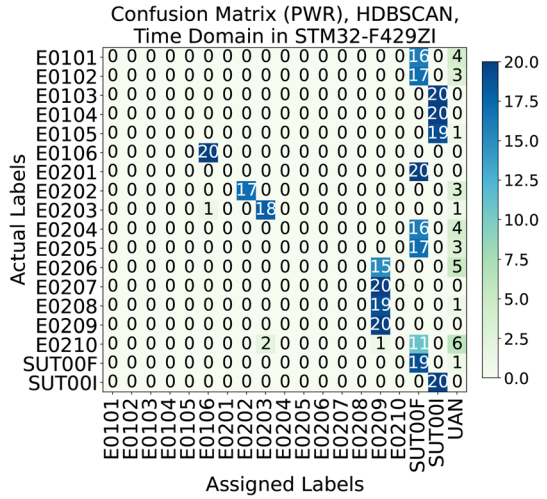

Fig. 13 STM NUCLEO-144 (Ethernet): Confusion Matrix.

Analysis of the three evaluation scenarios (Table 7) offers valuable insights for power analysis in low-end systems with active communication interfaces:

1. **Anomaly Detection.** The usage of the Ethernet port significantly impacts anomaly detection capabilities. The detection rate of 51.25%, as evidenced by the drop in MCC score from 65.10% (serial port) to 30.75% (Ethernet port), reveals the specific challenge posed by network interfaces. Notably, despite this interference, CARNYX still detects vulnerabilities at a rate over 8 times better than random categorisation, demonstrating that signal characteristics remain detectable even in noisy environments-a capability not demonstrated in prior binary detection systems.
2. **Arithmetic vs. Memory Error Detection.** Compared to the serial port case, CARNYX's differentiation capability drops from 50.00% to 16.67% for arithmetic faults and from 79.50% to 62.00% for memory faults (Figure 12). This disproportionate impact reveals an important characteristic: memory errors produce power signatures more resilient to Ethernet noise than arithmetic errors. This finding suggests future SCA systems could prioritise memory vulnerability detection in noisy environments, a nuanced insight unfeasible with binary detection systems.
3. **Specific Error Detection.** Despite the challenging environment, CARNYX successfully distinguished three independent error types with high precision: E0106 (Divide by zero decimal), E0202 (Buffer overflow), and E0203 (Double free). Additionally, E0206 (Out-of-bounds read), E0207 (Out-of-memory), E0208 (Stack overflow), and E0209 (Stack underflow) form a distinct cluster, suggesting similar power signatures that, while not individually distinguishable under Ethernet noise, remain collectively detectable. This level of granularity-identifying specific vulnerability classes even in noisy environments-represents a significant advance over existing binary anomaly detection systems.

These results demonstrate that while communication peripheral noise presents a significant challenge for power-based vulnerability detection, CARNYX still provides actionable insights by identifying specific vulnerability classes even under these adverse conditions-a capability not offered by any existing approach.

5.5 DUT 3: BeagleBone Black

The BeagleBone Black (BBB) represents a medium-end embedded system commonly deployed in industrial applications, equipped with a 1 GHz ARM Cortex-A8 processor and a Linux-based OS. Unlike the STM32F4-based platforms evaluated previously, the BBB constitutes a device with enhanced computational capabilities suitable for tasks such as automation control and edge computing. Its OS and complex hardware introduce additional challenges for power consumption analysis, providing a realistic test case for assessing CARNYX's applicability in industrial environments.

Measuring power consumption in the BBB, unlike the Riscure Piñata and STM NUCLEO-144 with dedicated measurement pins, requires adapting to its complex layout. The

Table 7 Clustering Results for STM NUCLEO-144 (Ethernet)

Anomaly Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Error	51.25	99.40	67.63		
No Error	97.50	20.00	33.20	56.39	30.75
Arith. vs. Memory Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Arithmetic Error	16.67	95.24	28.37		
Memory Error	62.00	93.23	74.47	50.83	39.42
Specific Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
E0101	0.00	0.00	0.00		
E0102	0.00	0.00	0.00		
E0103	0.00	0.00	0.00		
E0104	0.00	0.00	0.00		
E0105	0.00	0.00	0.00		
E0106	100.00	95.24	97.56		
E0201	0.00	0.00	0.00		
E0202	85.00	100.00	91.89	31.22	30.84
E0203	90.00	90.00	90.00		
E0204	0.00	0.00	0.00		
E0205	0.00	0.00	0.00		
E0206	0.00	0.00	0.00		
E0207	0.00	0.00	0.00		
E0208	0.00	0.00	0.00		
E0209	100.00	26.67	42.11		
E0210	0.00	0.00	0.00		

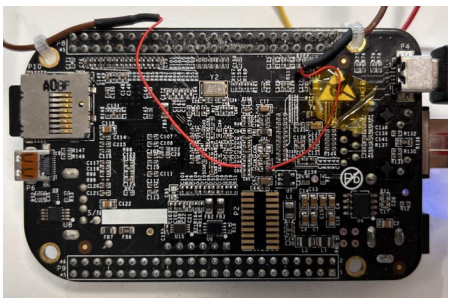

Fig. 14 Modified BeagleBone Black for power measurements.

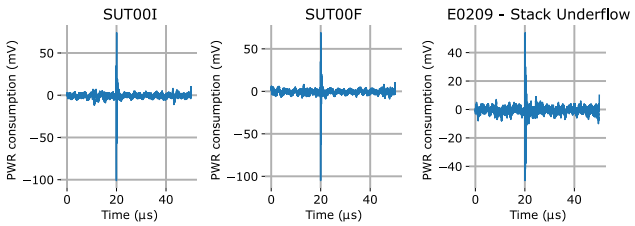

Fig. 15 Power consumption traces from BeagleBone Black over a 50-microsecond window, showing a spike. Calibration traces (SUT00I and SUT00F) maintain consistent shape and amplitude, while a stack underflow (E0209) exhibits variations in spike amplitude and waveform structure.

capacitor C53, near the power regulator, was removed, and wires were soldered to its pads, connecting the Riscure current probe's positive terminal to the input pad and the negative to a GPIO ground pin (Figure 14). This setup captures the power regulator's current, including non-core components, reflecting a practical approach for systems lacking dedicated measurement points.

Power consumption traces, collected per Section 4.3 at 10 GS/s (ten times the BBB's clock frequency), comprise 540 traces (200 calibration [100 baseline + 100 idle] + 340 operational), as shown in Figure 15. Fewer samples (50,000) are used compared to other platforms due to the BBB's higher processor frequency, which executes test programs faster, but the high sampling rate ensures sufficient resolution for power response analysis. Vulnerability E0210 (Unaligned Address) could not be exploited due to the BBB's hardware protections, preventing such memory access violations. Preprocessing took 126.04 seconds (0.23 seconds per trace), clustering required 8.75 seconds (0.016 seconds per trace), and operational trace processing used a pre-calibrated HDBSCAN model in 5–10 seconds, suitable for industrial quality control. The process consumed 400 MB of memory, reflecting operating system complexity.

The confusion matrix (Figure 17) reveals several insights about power-based vulnerability detection in industry-grade

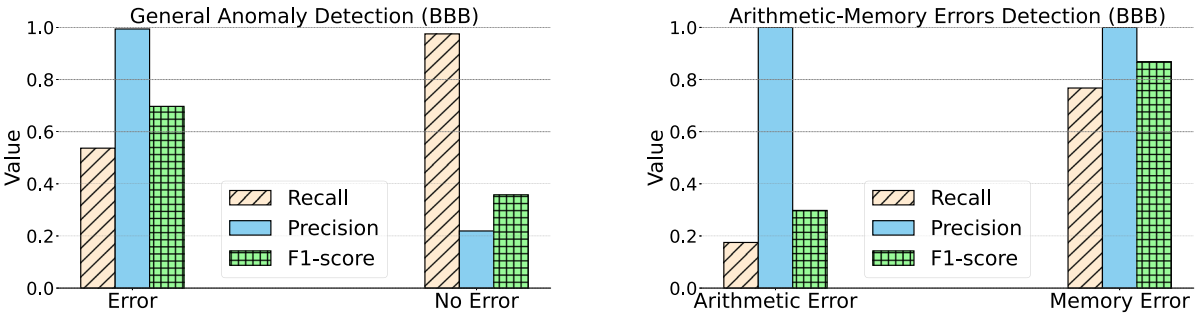

Fig. 16 BeagleBone Black: Metrics for Anomaly Detection (left) and Arithmetic vs. Memory Error Detection (right).

embedded systems. We observe a 46.33% false negative rate, with many error traces categorised as normal operation. This increase over the STM NUCLEO-144 with Ethernet (48.75%) suggests the operating system's background activities introduce comparable noise levels to active communication peripherals. Additionally, we note that 2.64% of traces remain as unassigned anomalies (UAN), indicating power signatures that deviate from normal patterns but lack sufficient distinctive characteristics for specific categorisation.

The analysis of our three evaluation scenarios (Table 8, Figure 16) provides additional context for CARNYX's capabilities on industry-standard OS-equipped embedded systems:

1. **Anomaly Detection.** CARNYX achieves 53.67% recall in detecting anomalous executions on BeagleBone Black, comparable to the STM NUCLEO-144 with Ethernet interface (51.25%). This similarity suggests that both active communication peripherals and operating system overhead-common in industrial deployments-present comparable challenges for power-based anomaly detection. Despite these challenges, the precision remains high at 99.38%, indicating that when CARNYX identifies an anomaly in an industrial setting, it does so with high confidence.
2. **Arithmetic vs. Memory Error Detection.** The framework demonstrates a significant disparity in detection capabilities between arithmetic and memory errors on this platform. Memory-related vulnerabilities are identified with 76.74% recall and 100.00% precision, while arithmetic errors show only 17.50% recall. This pattern, also observed with the STM NUCLEO-144 Ethernet case, further confirms that memory operations produce more distinctive power signatures than arithmetic operations in noisy industrial environments-whether noise stems from communication peripherals or operating system activities typical in industrial control systems.
3. **Specific Error Detection.** Despite the challenging industrial environment, CARNYX successfully identifies five specific vulnerability types with high precision: E0105 (Divide by Zero Integer), E0201 (Segmentation Fault), E0205 (Out-of-bounds Write), E0207 (Out-of-memory), and E0208 (Stack Overflow). This represents a slight improvement over the four vulnerability types detected in the STM NUCLEO-144 Ethernet case, suggesting that certain error classes produce sufficiently distinctive power signatures even in the presence of OS overhead.

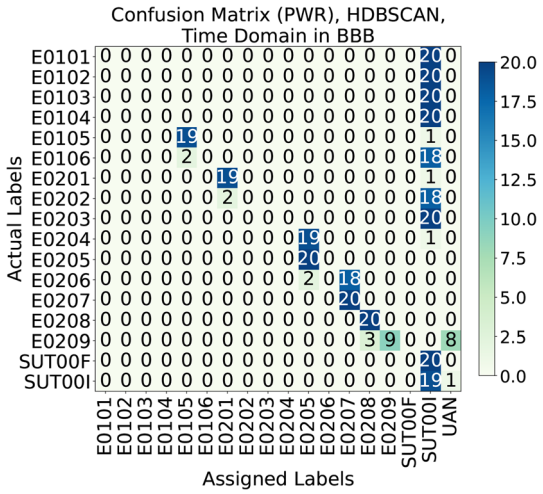

Fig. 17 BeagleBone Black: Confusion Matrix.

These results demonstrate that CARNYX maintains useful detection capabilities even on industry-standard embedded systems with operating system overhead. While the overall recall is lower than on simpler platforms, the framework still identifies specific vulnerability types with high precision, providing actionable security insights beyond binary anomaly detection for real-world industrial applications.

Table 8 Clustering Results for BeagleBone Black

Anomaly Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Error	53.67	99.38	69.70		
No Error	97.50	21.91	35.78	58.82	33.01
Arith. vs. Memory Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Arithmetic Error	17.50	100.00	29.79		
Memory Error	76.74	100.00	86.84	58.01	50.62
Specific Error Detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
E0101	0.00	0.00	0.00		
E0102	0.00	0.00	0.00		
E0103	0.00	0.00	0.00		
E0104	0.00	0.00	0.00		
E0105	95.00	90.48	92.68		
E0106	0.00	0.00	0.00		
E0201	95.00	90.48	92.68	37.06	39.01
E0202	0.00	0.00	0.00		
E0203	0.00	0.00	0.00		
E0204	0.00	0.00	0.00		
E0205	100.00	48.78	65.57		
E0206	0.00	0.00	0.00		
E0207	100.00	52.63	68.97		
E0208	100.00	86.96	93.02		
E0209	45.00	100.00	62.07		
E0210	N/A	N/A	N/A		

6 Discussion

This section assesses CARNYX's role in embedded system security by comparing its performance with state-of-the-art methods, exploring its scope and practical applicability to low-end embedded systems, and identifying key limitations and challenges for future consideration.

6.1 Comparison with State-of-the-Art Methods

Power consumption analysis for anomaly detection in embedded systems has evolved significantly, yet most existing approaches remain limited in two critical dimensions: granularity and environmental adaptability. As shown in Table 9, prior works such as WattsUpDoc [18], Ding *et al.* [44], and TrustGuard [43] achieve reasonable detection accuracy (85-99%) but only provide binary (normal/anomalous) categorisation, lacking detailed insights into vulnerability types. Additionally, these approaches have primarily been evaluated in controlled environments that lack real-world deployment conditions. Thus, CARNYX advances the state of the art in several key dimensions:

- **Granular vulnerability categorisation:** Unlike binary approaches, CARNYX distinguishes between 16 specific vulnerability types, enabling targeted remediation strategies. This granularity persists even in challenging environments, providing actionable security insights beyond simple anomaly detection.
- **Performance in realistic conditions:** Our results demonstrate robust detection across varying levels of environmental complexity-from controlled (99.69% recall on Piñata) to realistic (86.88% on NUCLEO-144 with serial interface) to challenging (51.25% with Ethernet). The latter represents an 8-fold improvement over random categorisation (6.25%) in a noisy environment where no previous system has been evaluated.
- **Hardware accessibility:** While TrustGuard achieves faster processing (<1ms) using specialized FPGA hardware, CARNYX operates on standard equipment with reasonable processing times (5-10s), making it more accessible for practical security testing during development phases where processing speed is less critical than diagnostic depth.
- **Communication peripheral impacts:** Our research quantifies, for the first time, how communication interfaces affect side-channel leakage-with Ethernet reducing detection recall by approximately 35% compared to serial

Table 9 Performance Comparison of CARNYX with Related Works

Method	Detection Recall	Processing Time	Hardware	Granularity
WattsUpDoc [18]	85%	Offline	Specific HW	Binary
Ding *et al.* [44]	92.7%	∼5s	Standard	Binary
Bai *et al.* [48]	96%	∼100ms	Standard	Binary
TrustGuard [43]	99%	<1ms	FPGA	Binary
CARNYX	99.69% (Piñata) 86.88% (NUCLEO-144) 53.67% (BeagleBone Black)	5-10s	Standard	Granular (16 types)

Note: State-of-the-art metrics are from cited works; reimplementation was not attempted due to incompatible hardware (e.g., FPGA vs. STM32F4) and unavailable datasets

connections. This finding builds on prior noise studies [51] and establishes a new baseline for understanding deployment challenges in low-end systems.

These advances demonstrate that power-based SCA can provide detailed vulnerability insights beyond binary detection, even in realistic deployment environments with communication peripherals-a capability not previously demonstrated in the literature.

6.2 Scope and Considerations

Platform generalisability. Our evaluation of CARNYX spans three platforms: the Riscure Piñata and STM NUCLEO-144, both based on the STM32F4 architecture, and the BeagleBone Black, a medium-end system with a Linux-based operating system. The STM32F4 platforms exhibit clear power consumption patterns due to their deterministic execution and hardware simplicity. The BeagleBone Black, with its operating system overhead, yields 53.67% recall, demonstrating applicability to industrial-grade systems. The underlying principles-analysing deviations in power consumption caused by altered instruction sequences during vulnerability exploitation-should extend to other low- and medium-end embedded systems with similar processing and memory constraints.

Cross-architecture considerations. While our evaluations centred on STM32F4 and ARM Cortex-A8 platforms, the underlying methodology naturally extends to other architectural families. RISC-V designs, with their deterministic execution flows, would likely yield clear power signatures during vulnerability exploitation, whereas AVR microcontrollers might offer even more distinct patterns due to their simpler pipeline structures. Despite the additional complexity of dual-core platforms like ESP32 with wireless subsystems, these architectures should remain compatible with CARNYX through appropriate calibration and filtering enhancements. Fundamentally, our approach's focus on behavioural power patterns rather than implementation-specific features suggests broad applicability across diverse embedded system architectures.

Impact of communication peripherals. The significant difference between serial (86.88%) and Ethernet (51.25%) recall rates on the NUCLEO-144 board highlights the impact of communication peripherals on detection accuracy, with Ethernet noise levels comparable to the BeagleBone Black's operating system overhead. This variation aligns with the operational characteristics of these interfaces-serial connections generate minimal background activity, while Ethernet controllers continuously manage network packets and connection state, providing valuable context for real-world deployment scenarios where interface options or operating system complexity vary.

Temperature effects. While temperature can influence power consumption, this factor had minimal impact on our experiments due to the microsecond-scale execution time of our test routines. We observed that low-end devices (Riscure Piñata, STM NUCLEO-144) exhibit less thermal sensitivity than medium-end counterparts (BeagleBone Black) due to simpler architectures and lower power density. Additionally, hardware layout characteristics-such as component density, PCB layers, and thermal vias-can significantly influence temperature distribution and its effects on power signatures. For extended testing, particularly with medium-end devices or dense layouts, we recommend test benches with cooling systems electrically isolated from the device under test to prevent measurement interference. This engineering approach, combined with CARNYX's statistical preprocessing, effectively mitigates temperature-induced variations without requiring methodological changes.

Pre-deployment utility. CARNYX demonstrates particular utility as a pre-deployment security validation tool, where granular vulnerability categorisation facilitates targeted remediation. CARNYX establishes baselines through

reference program execution, then analyses power responses from potentially vulnerable software for specific vulnerability detection. Its applicability to industrial systems like the BeagleBone Black, commonly used in automation and edge computing, enhances its value for real-world embedded security. For operational scenarios, CARNYX's unsupervised clustering addresses ground truth limitations by grouping similar power responses without requiring prior knowledge of vulnerable traces. The framework detects anomalies by identifying power consumption patterns that deviate from established baselines, as demonstrated under realistic conditions with operating system overhead and communication noise, enabling vulnerability detection even when ground truth is unavailable. For systems with higher complexity and multiple concurrent processes, alternative side-channel parameters or complementary approaches may be necessary. *Complementary analysis value.* Finally, CARNYX can serve as a complementary methodology to other vulnerability detection techniques, such as fuzzing, static and dynamic analysis. While these techniques are based on software analysis, the integration of CARNYX enables the identification of vulnerabilities that do not provide differentiable software responses. In this way, CARNYX offers a hardware-based perspective that extends beyond traditional methods.

6.3 Limitations and Security Considerations

CARNYX exhibits several constraints:

- Diminished effectiveness in systems with higher noise levels.
- Current validation limited to STM32F4 and ARM Cortex-A8 architectures, necessitating recalibration for other platforms.
- Processing latency of 5-10 seconds per trace that may be suboptimal for certain applications.

From a security perspective, potential evasion techniques include execution flow manipulation (e.g., code polymorphism [65]) that could generate altered power consumption patterns. Though malicious code injection and evasion techniques presents another theoretical concern [66], the constrained nature of low-end embedded systems typically limits the feasibility of such sophisticated attacks.

7 Conclusion

SCA-based hardware analysis offers valuable advantages over software methods by capturing physical responses to security vulnerabilities in embedded systems. However, existing power-based anomaly detection approaches have been limited to binary categorisation without root cause identification, significantly impeding effective countermeasures. This paper presented CARNYX, a framework that advances beyond this limitation by categorising anomalies into specific vulnerability types, enabling targeted mitigation strategies.

Our experimental results demonstrate that CARNYX performs exceptionally well in controlled environments (99.69% recall on Riscure Piñata) and maintains strong performance in more realistic scenarios (86.88% recall on STM NUCLEO-144 via serial). Even in the challenging case of Ethernet communication-where active network interfaces introduce significant noise-CARNYX still achieves 51.25% recall across 16 vulnerability categories, significantly outperforming the 6.25% expected from random categorisation. Results on the BeagleBone Black, a real-world medium-end system with operating system overhead, achieve 53.67% recall. This represents a substantial advance over existing binary detection methods that provide no vulnerability-specific insights and have not been evaluated under similarly realistic conditions.

These results reveal two key findings. First, power consumption analysis can provide granular vulnerability insights in low- and medium-end embedded systems, particularly effective due to their hardware simplicity. Second, we have quantified how communication peripherals impact side-channel leakage-a previously unexamined factor in security literature that significantly affects detection performance. Our successful testing on BeagleBone Black further demonstrates applicability to medium-end systems used in industrial automation and edge computing. This second finding establishes important baseline measurements for future work in realistic deployment environments.

CARNYX's automated architecture consisting of data acquisition, preprocessing, and clustering modules enables efficient vulnerability detection during pre-deployment testing, addressing a critical gap in embedded security. Its source code and implementation examples for all test scenarios are available on GitHub[5], facilitating further research in this area.

Future work could pursue several promising directions to enhance CARNYX. First, validation might extend to multiple low- and medium-end architectures like additional ARM and RISC-V platforms, assessing cross-architecture applicability. Second, wavelet-based preprocessing techniques could potentially filter communication peripheral noise, as our findings suggest this represents a significant but potentially separable interference source. Third, incorporating multiple physical channels (power combined with electromagnetic emanations) could improve detection accuracy in noisy environments. Fourth, integrating CARNYX into CI/CD pipelines would streamline security testing in development workflows. Fifth, adapting for runtime monitoring, implementing FPGA-based acceleration, and optimising energy

[5] https://github.com/JorgeBarredo14/carnyx

consumption would extend CARNYX's utility to operational environments and resource-constrained test scenarios. These directions build directly on our discovery of communication peripherals' impact on side-channel leakage, turning this challenge into an opportunity for targeted methodological improvements.

Acknowledgements CRITIC Project Grant PLEC2024-011222 funded by AEI/10.13039/501100011033, FEDER and UE. Mikel Iturbe is partially supported by the Basque Government (grant number IT1676-22).

Author Contributions All authors contributed to the study conception and design. Data collection and analysis were performed by J.B., M.E., J.L.F., and M.I. The first draft was written by J.B., and all authors reviewed and approved the final manuscript.

Data Availability The source code and three proofs of concepts are available at a GitHub repository[1]. Additional data supporting this study are available from the corresponding author upon reasonable request.

Declarations

Ethical Approval Not applicable, as this study involves no human participants or animals.

References

1. Danladi, M., Baykara, M.: Low Power Wide Area Network Technologies: Open Problems, Challenges, and Potential Applications (2022). https://doi.org/10.18280/rces.090205
2. Meneghello, F., Calore, M., Zucchetto, D., Polese, M., Zanella, A.: IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices (2019). https://doi.org/10.1109/JIOT.2019.2935189
3. Markets, R.: Global Embedded System Market (2020 to 2025) - Rapid Adoption of Embedded Systems in Smart Homes Presents Lucrative Opportunities. https://www.globenewswire.com/news-release/2020/03/26/2006966/28124/en/Global-Embedded-System-Market-2020-to-2025-Rapid-Adoption-of-Embedded-Systems-in-Smart-Homes-Presents-Lucrative-Opportunities.html (2019)
4. Ravi, S., Raghunathan, A., Kocher, P., Hattangady, S.: Security in Embedded Systems: Design Challenges (2004). https://doi.org/10.1145/1015047.1015049
5. ENISA. ENISA Threat Landscape 2022 (2022). https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022
6. Open Web Application Security Project. OWASP Internet of Things (IoT) Top 10 2018. https://owasp.org/www-project-internet-of-things/
7. Johnson, W.A., Ghafoor, S., Prowell, S.: A Taxonomy and Review of Remote Attestation Schemes in Embedded Systems (2021). https://doi.org/10.1109/ACCESS.2021.3119220
8. Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., Markakis, E.K.: A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues (2020). https://doi.org/10.1109/COMST.2019.2962586
9. Loi, F., Sivanathan, A., Habibi Gharakheili, H., Radford, A., Sivaraman, V.: Systematically Evaluating Security and Privacy for Consumer IoT Devices (2017). https://doi.org/10.1145/3139937.3139938
10. Ferrag, M.A., Shu, L.: The Performance Evaluation of Blockchain-Based Security and Privacy Systems for the Internet of Things: A Tutorial (2021). https://doi.org/10.1109/JIOT.2021.3078072
11. Kleidermacher, D.: Integrating Static Analysis into a Secure Software Development Process (2008). https://doi.org/10.1109/THS.2008.4534479
12. Zaddach, J., Bruno, L., Francillon, A., Balzarotti, D.: Avatar: A Framework to Support Dynamic Security Analysis of Embedded Systems' Firmwares (2014). https://doi.org/10.14722/ndss.2014.23229
13. Eisele, M., Maugeri, M., Shriwas, R., Huth, C., Bella, G.: Embedded fuzzing: a review of challenges, tools, and solutions (2022). https://doi.org/10.1186/s42400-022-00123-y
14. Picek, S., Heuser, A., Jovic, A., Ludwig, S.A., Guilley, S., Jakobovic, D., Mentens, N.: Side-channel analysis and machine learning: A practical perspective (2017). https://doi.org/10.1109/IJCNN.2017.7966373
15. Rioja, U., Paguada, S., Batina, L., Armendariz, I.: The Uncertainty of Side-Channel Analysis: A Way to Leverage from Heuristics (2021). https://doi.org/10.1145/3446997
16. Hospodar, G., Gierlichs, B., Mulder, E., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: A first study (2011). https://doi.org/10.1007/s13389-011-0023-x
17. Nappa, A., Papadopoulos, P., Varvello, M., Gomez, D.A., Tapiador, J., Lanzi, A.: PoW-How: An Enduring Timing Side-Channel to Evade Online Malware Sandboxes (2021). https://doi.org/10.1007/978-3-030-88418-5_5
18. Clark, S.S., Ransford, B., Rahmati, A., Guineau, S., Sorber, J., Xu, W., Fu, K.: WattsUpDoc: Power Side Channels to Nonintrusively Discover Untargeted Malware on Embedded Medical Devices (2013). https://www.usenix.org/conference/healthtech13/workshop-program/presentation/clark
19. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual Information Analysis (2008). https://doi.org/10.1007/978-3-540-85053-3_27
20. Sehatbakhsh, N., Nazari, A., Alam, M., Werner, F.T., Zhu, Y., Zajić, A.G., Prvulović, M.: REMOTE: Robust External Malware Detection Framework by Using Electromagnetic Signals (2020). https://api.semanticscholar.org/CorpusID:208116287
21. Lazim Qaddoori, S., Ali, Q.I.: An embedded and intelligent anomaly power consumption detection system based on smart metering (2023). https://doi.org/10.1049/wss2.12054
22. Ara, G., Cucinotta, T., Mascitti, A.: Simulating execution time and power consumption of real-time tasks on embedded platforms (2022). https://doi.org/10.1145/3477314.3507030
23. Krosman, K., Sosnowski, J., Gawkowski, P.: Object oriented time series exploration: Applied to power consumption analysis of embedded systems (2021). https://doi.org/10.1016/j.eswa.2021.115531https://www.sciencedirect.com/science/article/pii/S0957417421009398
24. Camposano, R., Wilberg, J.: Embedded system design (1996). https://doi.org/10.1007/BF00134682.https://publica.fraunhofer.de/handle/publica/189659

25. Bolton, W.: Programmable Logic Controllers (2009). https://doi.org/10.1016/B978-1-85617-751-1.00001-X
26. Al-Ali, A., Al-Rousan, M.: Java-based home automation system (2004). https://doi.org/10.1109/TCE.2004.1309414
27. Eceiza, M., Flores, J.L., Iturbe, M.: Fuzzing the Internet of Things: A Review on the Techniques and Challenges for Efficient Vulnerability Discovery in Embedded Systems (2021). https://doi.org/10.1109/JIOT.2021.3056179
28. Muench, M., Stijohann, J., Kargl, F., Francillon, A., Balzarotti, D.: What You Corrupt Is Not What You Crash: Challenges in Fuzzing Embedded Devices (2018). https://doi.org/10.14722/ndss.2018.23166
29. Winter, J.: Trusted computing building blocks for embedded linux-based ARM trustzone platforms (2008). https://doi.org/10.1145/1456455.1456460
30. Muñoz, A., Ríos, R., Román, R., López, J.: A survey on the (in)security of trusted execution environments (2023). https://doi.org/10.1016/j.cose.2023.103180. https://www.sciencedirect.com/science/article/pii/S0167404823000901
31. van der Veen, V., Dutt-Sharma, N., Cavallaro, L., Bos, H.: Memory Errors: The Past, the Present, and the Future (2012). https://doi.org/10.1007/978-3-642-33338-5_5
32. Costin, A., Zaddach, J., Francillon, A., Balzarotti, D.: A large-scale analysis of the security of embedded firmwares (2014). https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/costin
33. Batina, L., Jauernig, P., Mentens, N., Sadeghi, A.R., Stapf, E.: In Hardware We Trust: Gains and Pains of Hardware-assisted. Security (2019). https://doi.org/10.1145/3316781.3323480
34. Thakor, V.A., Razzaque, M.A., Khandaker, M.: Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review. Comparison and Research Opportunities (2021). https://doi.org/10.1109/ACCESS.2021.3052867
35. Council of European Union. Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures to ensure a high common level of cybersecurity in the Union and amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972 and repealing the Directive (EU) 2016/1148 (NIS 2 Directive) (2022). https://eur-lex.europa.eu/eli/dir/2022/2555/oj
36. European Commission. Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements, amending Regulation (EU) 2019/1020 and Directive (EU) 2020/1828 (2022). https://www.cyberresilienceact.eu/the-cyber-resilience-act/
37. IEC 62443-4-1:2018 - Security for industrial automation and control systems - Part 4-1: Secure product development lifecycle requirements. Standard, International Electrotechnical Commission (2018). https://webstore.iec.ch/en/publication/33615
38. IEC 62443-4-2:2019 - Security for industrial automation and control systems - Part 4-2: Technical security requirements for IACS components. Standard, International Electrotechnical Commission (2019). https://webstore.iec.ch/publication/34421
39. Papp, D., Ma, Z., Buttyan, L.: Embedded systems security: Threats, vulnerabilities, and attack taxonomy (2015). https://doi.org/10.1109/PST.2015.7232966
40. Das, D., Sen, S.: Electromagnetic and Power Side-Channel Analysis: Advanced Attacks and Low-Overhead Generic Countermeasures through White-Box Approach (2020) https://www.mdpi.com/2410-387X/4/4/30
41. Wang, X., Zhou, Q., Harer, J., Brown, G., Qiu, S., Dou, Z., Wang, J., Hinton, A., Gonzalez, C.A., Chin, P.: Deep learning-based classification and anomaly detection of side-channel signals (2018). https://doi.org/10.1117/12.2311329
42. Moore, S., Yampolskiy, M., Gatlin, J., McDonald, J.T., Andel, T.R.: Buffer overflow attack's power consumption signatures (2016). https://doi.org/10.1145/3015135.3015141
43. Zhang, T., Tehranipoor, M., Farahmandi, F.: TrustGuard: Standalone FPGA-Based Security Monitoring Through Power Side-Channel (2024). https://doi.org/10.1109/TVLSI.2023.3335876
44. Ding, F., Li, H., Luo, F., Hu, H., Cheng, L., Xiao, H., Ge, R.: DeepPower: Non-intrusive and Deep Learning-based Detection of IoT Malware Using Power Side Channels (2020). https://doi.org/10.1145/3320269.3384727
45. Pham, D.P., Marion, D., Mastio, M., Heuser, A.: Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification (2021). https://doi.org/10.1145/3485832.3485894
46. Abbasi, Z., Kargahi, M., Mohaqeqi, M.: Anomaly detection in embedded systems using simultaneous power and temperature monitoring (2014). https://doi.org/10.1109/ISCISC.2014.6994033
47. Wang, P., Govindarasu, M., Ashok, A., Sridhar, S., McKinnon, D.: Data-Driven Anomaly Detection for Power System Generation. Control (2017). https://doi.org/10.1109/ICDMW.2017.152
48. Bai, Y., Park, J., Tehranipoor, M., Forte, D.: Real-time instruction-level verification of remote IoT/CPS devices via side channels (2022). https://doi.org/10.1007/s43926-022-00021-2
49. Lindon, J.C., Tranter, G.E., Koppenaal, D.: Encyclopedia of spectroscopy and spectrometry (2016). https://www.sciencedirect.com/referencework/9780128032244/encyclopedia-of-spectroscopy-and-spectrometry
50. Betta, G., Liguori, C., Pietrosanto, A.: Structured approach to estimate the measurement uncertainty in digital signal elaboration algorithms (1999). https://doi.org/10.1049/ip-smt:19990001
51. Mangard, S., Oswald, M., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards (2007). https://doi.org/10.1007/978-0-387-38162-6
52. Vinutha, H.P., Poornima, B., Sagar, B.M.: Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset (2018). https://doi.org/10.1007/978-981-10-7563-6_53
53. Hegde, R., Shanbhag, N.: Energy-efficient signal processing via algorithmic noise-tolerance (1999). https://doi.org/10.1145/313817.313834
54. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis (1987). https://doi.org/10.1016/0169-7439(87)80084-9
55. Burnham, K.P., Anderson, D.R.: Multimodel inference: understanding AIC and BIC in model selection (2004). https://doi.org/10.1177/0049124104268644
56. Rodriguez, M.Z., Comin, C.H., Casanova, D., Bruno, O.M., Amancio, D.R., Costa, L.d.F., Rodrigues, F.A.: Clustering algorithms: A comparative approach (2019). https://doi.org/10.1371/journal.pone.0210236
57. McInnes, L., Healy, J., Astels, S., et al.: hdbscan: Hierarchical density based clustering. (2017). https://doi.org/10.21105/joss.00205
58. Kriegel, H.P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering (2011). https://doi.org/10.1002/widm.30
59. Flach, P., Kull, M.: Precision-Recall-Gain Curves: PR Analysis Done Right (2015). https://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right.pdf
60. Harbecke, D., Chen, Y., Hennig, L., Alt, C.: Why only Micro-F1? Class Weighting of Measures for Relation Classification (2022). https://doi.org/10.18653/v1/2022.nlppower-1.4
61. Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation (2020). https://doi.org/10.1186/s12864-019-6413-7
62. Tamura, M., Tsujita, S.: A study on the number of principal components and sensitivity of fault detection using PCA (2007). https://doi.org/10.1016/J.COMPCHEMENG.2006.09.004
63. MITRE Corporation. Common Weakness Enumeration (CWE) (2025). https://cwe.mitre.org

64. Raje, K.: STM32 Series Single Chip Microcomputer Market Report 2025 (2025). https://www.cognitivemarketresearch.com/stm32-series-single-chip-microcomputer-market-report
65. Couroussé, D., Barry, T., Robisson, B., Jaillon, P., Potin, O., Lanet, J.L.: Runtime code polymorphism as a protection against side channel attacks (2016). https://doi.org/10.1007/978-3-319-45931-8_9
66. Han, Y., Chan, M., Aref, Z., Tippenhauer, N.O., Zonouz, S.: Hiding in Plain Sight? On the Efficacy of Power Side Channel-Based Control Flow Monitoring (2022). https://www.usenix.org/conference/usenixsecurity22/presentation/han