Computers & Security 159 (2025) 104692

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

GJALLARHORN: A framework for vulnerability detection via electromagnetic side-channel analysis in embedded systems

Jorge Barredo [a,b,*], Maialen Eceiza [a], Jose Luis Flores [c], Mikel Iturbe [b]

[a] IKERLAN Technology Research Centre, Pº José María Arizmendiarrieta, 2, Arrasate/Mondragón, 20500, Basque Country, Spain
[b] Mondragon Unibertsitatea, Loramendi Kalea, 4, Arrasate/Mondragón, 20500, Basque Country, Spain
[c] Universidad del País Vasco (UPV/EHU), Europa Plaza, 1, Donostia/San Sebastián, 20018, Basque Country, Spain

ARTICLE INFO

Keywords:
Embedded systems
Side-channel analysis
Security monitoring
Vulnerability detection
IoT security

ABSTRACT

The proliferation of embedded systems within the Internet of Things (IoT) has heightened the difficulty of detecting vulnerabilities due to their inherent resource constraints. This paper introduces GJALLARHORN, a framework extending electromagnetic side-channel analysis (EM SCA) for early-stage vulnerability detection in embedded systems. Unlike conventional methods requiring code access or imposing computational overhead, GJALLARHORN non-invasively analyses EM emissions to identify anomalous patterns indicating potential security vulnerabilities. By observing hardware-level manifestations of software execution, GJALLARHORN complements software-level analysis, revealing vulnerabilities that might otherwise remain undetected. The framework adapts to device complexity, enabling categorisation of up to 16 distinct vulnerability types, including buffer overflows, memory leaks, and arithmetic errors. Evaluations on both low-end (STM NUCLEO-144) and high-end (Raspberry Pi 3B) architectures demonstrate GJALLARHORN's effectiveness, achieving a recall of 95.94% and F_1 score of 96.39% on the low-end system, and 73.33% recall with 84.61% F_1 score on the high-end system. Our results reveal that memory-related vulnerabilities produce more distinguishable EM signatures than arithmetic errors, offering valuable insights for externally detecting vulnerabilities. By enabling detection during development, GJALLARHORN helps mitigate risks before deployment, potentially reducing the economic impact of security incidents in IoT infrastructure.

1. Introduction

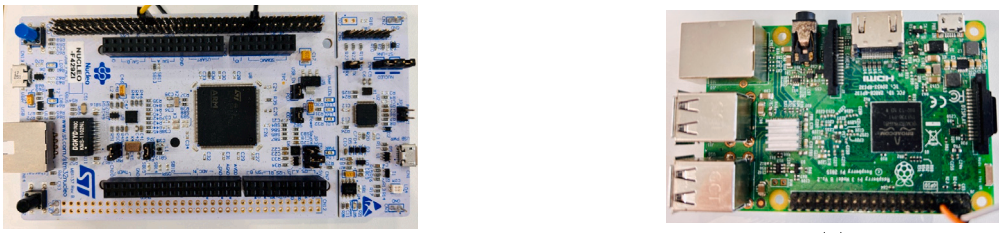
The Internet of Things (IoT) market is forecast to grow from $492.7 billion in 2023 to $3,454.2 billion by 2033 (Market.us, 2024), whilst embedded systems are expected to increase from 16.7 to 29.7 billion devices by 2027 (IoT Analytics, 2023). These systems underpin critical applications in industry, telecommunications and smart homes; however, their rapid uptake is accompanied by mounting security risks. Resource constraints, such as limited processing power and memory, make it difficult to deploy robust defences. In addition, once deployed, most devices remain unchanged. Consequently, patching them is seldom practicable, and their vulnerabilities persist, as highlighted in the OWASP IoT 2018 Top 10 (Open Web Application Security Project, 2018). Recent analysis reveals that 42% of IoT devices transmit sensitive data unencrypted (NETGEAR and Bitdefender, 2024), whilst the economic implications are considerable, with incidents averaging $330,000 (World Economic Forum, 2024). Consequently, regulatory frameworks such as the EU's NIS 2 Directive (Council of European Union, 2022) and Cyber Resilience Act (European Commission, 2022) mandate stringent security for connected devices in critical infrastructure.

Current security evaluation techniques exhibit limitations. Static analysis scrutinises source code before execution but fail to identify vulnerabilities manifesting during runtime (Chess and West, 2007). Runtime analysis approaches (Abadi and Fournet, 2001) offer behavioural monitoring but impose computational overhead that overwhelms the limited resources of most embedded devices. Similarly, fuzzing employs input generation to uncover software flaws, yet its resource requirements restrict scalability in embedded contexts (Miller et al., 1990). These constraints highlight a critical gap in traditional methods addressing the unique challenges of embedded systems.

In contrast, Side-Channel Analysis (SCA) presents an alternative method, bypassing these limitations through physical measurements. Initially developed for cryptographic system evaluation (Kocher, 1996), SCA exploits unintended leakage through channels such as power consumption and electromagnetic (EM) emissions to detect vulnerabilities. Its minimal computational footprint makes it particularly suitable

* Corresponding author at: Pº José María Arizmendiarrieta, 2, Arrasate/Mondragón, 20500, Basque Country, Spain.
E-mail address: jbarredo@ikerlan.es (J. Barredo).

https://doi.org/10.1016/j.cose.2025.104692
Received 2 April 2025; Received in revised form 15 July 2025; Accepted 29 September 2025
Available online 4 October 2025



(a) (b)

Fig. 1. Evaluated devices: (a) STM NUCLEO-144 and (b) RPi 3B.

for embedded systems, whilst revealing flaws invisible to code-based analysis (Quisquater and Samyde, 2001).

This paper introduces GJALLARHORN, a framework that extends SCA principles to systematically analyse EM emissions, enabling the detection of up to 16 distinct vulnerability types in embedded systems. Earlier SCA methods usually yield either a binary (normal/anomalous) result or a limited multi-class categorisation, and they rely on fixed methodologies. In contrast, GJALLARHORN tailors its preprocessing and anomaly detection routines to the complexity of each target, whether a low-end microcontroller or a high-end multicore platform. This flexibility ensures a comprehensive vulnerability assessment across the diverse hardware landscape of modern IoT.

Experimental validation across two embedded architectures demonstrates GJALLARHORN's effectiveness. On the low-end STM NUCLEO-144, shown alongside the high-end Raspberry Pi 3B (RPi 3B) in Fig. 1, it achieves 95.94% recall, identifying vulnerabilities that elude traditional methods. On the RPi 3B, it detects 73.33% of EM anomalies, demonstrating adaptability to varying computational capabilities. Additionally, our findings reveal that memory-related errors generate more pronounced EM signatures than arithmetic faults, providing valuable insights for vulnerability profiling.

This research makes two main contributions. First, it proposes a comprehensive framework for vulnerability detection using an EM-based SCA monitor. The approach is non-invasive and architecture-agnostic, providing a hardware-level view that complements software analysis. We validate the framework across both low-end and high-end embedded systems, demonstrating its adaptability to devices with very different computational resources.

Second, we present a detailed quantitative analysis of EM emission patterns under different vulnerability conditions. The results expose clear differences between the signatures produced by memory-related faults and those generated by arithmetic errors, giving practitioners practical cues for fine-grained diagnosis.

To support reproducibility and encourage further research in this area, we have released the source code and proof-of-concept implementations for the two validated use cases.[1]

The remainder of this paper is organised as follows: Section 2 establishes foundational concepts, Section 3 examines existing research in embedded system security and side-channel analysis, Section 4 details the framework's technical implementation, Section 5 presents experimental results, Section 6 explores implications and limitations, and Section 7 synthesises key findings and their technological significance.

2. Background

This section lays the groundwork for the security evaluation framework presented in this study. It first explores embedded systems, outlining their characteristics and a hardware-based taxonomy, followed by an analysis of security challenges stemming from their constraints and increasing connectivity. Next, it examines SCA, tracing its development and its application to embedded device security. Together, these subsections provide the technical and theoretical foundation for the vulnerability assessment approach described later in this paper.

2.1. Embedded systems

An embedded system combines tightly integrated hardware and software tailored to perform a specific function (Jiménez et al., 2013). Once installed, these systems generally allow limited modifications due to their predefined purpose (Camposano and Wilberg, 1996). Operating reactively, they process inputs under hardware constraints. Examples include security systems and industrial control systems, such as programmable logic controllers (PLCs).

2.1.1. Taxonomy of embedded systems

The classification of embedded systems depends on various application-specific factors, yet hardware capabilities directly affect the implementation of security features. Drawing on the taxonomy by Muench et al. (2018), Table 1 groups embedded systems into three categories according to their hardware resources.

2.1.2. Security in embedded systems

Embedded systems function under tight hardware constraints, including limited memory and processing capacity, which create difficulties for integrating effective security measures. For example, cryptographic functions demand significant computational resources (Batina et al., 2019), leading to a balance between security and operational performance. In the past, development often focused on essential functionality rather than security, resulting in vulnerabilities such as buffer overflows in IoT firmware (Thakor et al., 2021) and side-channel leakage in cryptographic implementations (Das and Sen, 2020). Additionally, unlike general-purpose systems, embedded devices typically lack runtime protections like stack canaries, increasing their exposure to risks.

As embedded systems grow in complexity, interconnectivity, and use (Thakor et al., 2021), these security challenges become more pronounced, requiring approaches that align with resource limitations. Consequently, regulatory bodies have introduced frameworks such as the Directive (EU) 2022/2555 (NIS2 Directive) (Council of European Union, 2022), the Cyber Resilience Act (CRA) (European Commission, 2022), and ISO/IEC 62443-4-1/2 (International Electrotechnical Commission, 2018, 2019) to establish comprehensive security standards for devices in critical infrastructure. These regulations require thorough testing, including methods like fuzzing as specified in ISO/IEC 62443-4-1, to detect vulnerabilities during development. However, such testing relies on a detailed understanding of system behaviour, which is hard to achieve given the constraints. Non-invasive techniques, such as pre-deployment vulnerability analysis using EM emissions, provide an efficient means to address these requirements without placing excessive demands on the system. Therefore, tailoring security methods to the specific needs of embedded systems is vital for reducing risks and ensuring compliance with regulatory standards.

2.2. Side-channel analysis

Side-channel analysis (SCA) exploits the physical by-products of computation, such as timing, power consumption or EM radiation,

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

Table 1
Taxonomy of embedded systems based on hardware resources.
Source: Adapted from Muench et al. (2018).

Category	Processor	RAM	OS	Protection	Ref.
High-end	Multicore 32-/64-bit	≥1 GB	✓	MMU[a], MPU[b], DEP[c]	Main (2010)
Medium-end	16/32-bit 1–2 cores	1 MB–1 GB	✓	MMU[a] only	Parameswaran and Wolf (2008)
Low-end	8/16-bit single-core	<1 MB	×	None	Muench et al. (2018), Eceiza et al. (2021)

[a] MMU: Memory Management Unit.
[b] MPU: Memory Protection Unit.
[c] DEP: Data Execution Protection.

to infer internal states or spot anomalous behaviour. Kocher's seminal timing and power attacks on cryptographic hardware (Kocher, 1996) demonstrated that logical secrets invariably leave analogue fingerprints. Subsequent work extended the technique to fault detection and malware spotting on resource-constrained devices because SCA adds *no* runtime overhead to the target and requires *no* source code. We restrict the remainder of the manuscript to EM leakage, as EM probes can be used without contact, a practical advantage over power consumption monitoring.

3. Related work

This section reviews existing research on security evaluation for embedded systems and IoT devices. It explores the limitations of traditional techniques and traces the development of EM-based SCA. Previous studies offer insights into vulnerability detection, though many struggle to adapt to diverse architectures or offer detailed anomaly categorisation. In this context, GJALLARHORN uses EM emissions to deliver an adaptable and precise evaluation method, addressing these shortcomings.

3.1. Security evaluation in embedded systems and IoT

Research into security evaluation for embedded systems and IoT has grown in importance as their deployment expands. Early studies concentrated on identifying common vulnerabilities, with audits confirming their presence in operational devices (Stoyanova et al., 2020). Later efforts shifted towards application-specific frameworks tailored to particular contexts. For example, Alrawi et al. (2019) examined home automation ecosystems to identify weaknesses, whilst FuzzDocs (You et al., 2022) developed automated API documentation extraction for specific systems. Research on RFID-based IoT applications also produced techniques for security and reverse engineering protection (Fernández-Caramés et al., 2017). However, these approaches often struggle to scale across the varied hardware found in IoT environments.

In parallel, frameworks that embed security within the design process have emerged. SecIoT (Huang et al., 2016) offers a general model, whilst other solutions focus on specific domains, such as continuous health monitoring (Hussain et al., 2021), smart infrastructure (Pacheco and Hariri, 2016), and access control (Bouij-Pasquier et al., 2015). Islam et al. (2016) proposed a risk assessment framework for automotive embedded systems, and Gras et al. (2020) investigated side-channel leakage in x86_64 microarchitectures. Traditional methods, including static analysis (Chess and West, 2007), runtime monitoring (Abadi and Fournet, 2001), and fuzzing (Miller et al., 1990), remain widely used. Static analysis inspects code structures prior to execution, runtime monitoring tracks operational behaviour, and fuzzing tests systems with varied inputs. Yet, these techniques encounter difficulties: static analysis misses runtime issues, runtime monitoring places heavy demands on limited resources, and fuzzing requires considerable computational capacity, reducing its effectiveness in diverse settings. These limitations indicate a need for alternative strategies.

3.2. EM-based side-channel analysis

Early EM-SCA work focused on cryptographic key extraction (Quisquater and Samyde, 2001). Over the last decade, the emphasis has shifted towards integrity monitoring, yet three limitations persist, as exposed in Table 2. First, most frameworks report only binary alarms and merely state whether an execution is anomalous (Nazari et al., 2017; Han et al., 2017; Wang et al., 2018). Attempts at richer labelling stop at four or five coarse classes and never map results to the CWE catalogue (Chawla et al., 2021; Sayakkara et al., 2019). Second, published methodologies rarely scale across hardware; most assume a single-core microcontroller at a fixed clock. Some works mention multi-core devices (Sehatbakhsh et al., 2020; Khan et al., 2019), yet their preprocessing is fixed, and portability is not evaluated. Third, runtime figures lack engineering context: processing times ranging from seconds to minutes are rarely related to continuous-integration loops, compliance testing or field diagnostics.

GJALLARHORN tackles these shortcomings by translating each cluster into one of sixteen CWE-aligned vulnerability types, offering actionable output; by parameterising both its time- and frequency-domain preprocessing algorithms so that moving from an MCU to a GHz-class device requires retuning rather than redesign; and by framing its current budget (37s to 65s per 340 traces) within realistic deployment scenarios while outlining straightforward optimisation paths.

3.3. Comparison with traditional vulnerability detection approaches

SCA-based detection provides advantages over traditional approaches. Static analysis (Chess and West, 2007), dynamic analysis (Abadi and Fournet, 2001), and fuzzing (Miller et al., 1990) are commonly employed for vulnerability detection, but each faces challenges in embedded systems (Eceiza et al., 2021). Table 3 outlines these methods alongside GJALLARHORN's SCA approach, illustrating their respective strengths.

GJALLARHORN enhances these methods by detecting vulnerabilities through hardware-level physical effects, specifically EM emissions, which are not captured by code-level analysis. Traditional methods, such as static and dynamic analysis, examine software behaviour within the execution environment, while fuzzing tests it through input manipulation. However, these approaches often rely on human software-level inspections that require extensive time and highly skilled personnel for manual code review or the development of customised automated testing frameworks. In contrast, GJALLARHORN observes the physical manifestations of execution via EM emissions, enabling it to uncover vulnerabilities that might remain hidden from such resource-intensive processes. This hardware-focused approach is particularly valuable in

Table 2
Comparison of related work on EM SCA frameworks and GJALLARHORN.

No.	Citation	Device Under Test (DUT)	Accuracy(%)	Categorisation	Hardware Adaptability
1	Wang et al. (2018)	Arduino, Raspberry Pi, Siemens PLC	90.00	Binary (normal/anomaly)	Fixed model
2	Nazari et al. (2017)	A13-OLinuXino	97.04	Binary (normal/anomaly)	Device-specific
3	Han et al. (2017)	Allen Bradley PLC	98.90	Binary (normal/anomaly)	PLC-specific
4	Pham et al. (2021)	Raspberry Pi	99.82	Binary (normal/anomaly)	Single device
5	Khan et al. (2019)	Altera Cyclone II, TS7250, A13-OLinuXino	100.00	Binary (normal/anomaly)	Fixed methodology
6	Chawla et al. (2021)	Cortex-M4, MSP430, FPGA	98.60	Multi-class (4 types)	Fixed wavelet analysis
7	Sehatbakhsh et al. (2020)	ODROID-XU4, Arduino Mega, MSP430	99.50	Binary (normal/anomaly)	No adaptation
8	Sayakkara et al. (2019)	Raspberry Pi, Arduino	>90	Multi-class (software activities)	Fixed neural network
9	GJALLARHORN	STM NUCLEO-144, Raspberry Pi 3B	95.94	Granular (up to 16 types)	Model adaptation

Table 3
Comparison between GJALLARHORN and traditional vulnerability detection methods.

Approach	Detection scope	Resource overhead	Code Access required	Runtime coverage	Hardware sensitivity
Static Analysis	Syntax/semantic errors	Low	Yes	Limited	None
Dynamic Analysis	Runtime behaviour	High	Partial	Good	Low
Fuzzing	Input-dependent errors	Medium-High	No	Variable	Low
Formal Verification	Logical inconsistencies	Very High	Yes	Comprehensive	None
GJALLARHORN	EM emanation	Low on target	No	Excellent	High

resource-constrained embedded systems, where the computational demands of dynamic analysis or the expertise needed for thorough static analysis are often impractical. By leveraging EM emissions, GJALLARHORN reduces the dependence on time-consuming and personnel-heavy methods, offering a more efficient and effective solution for vulnerability detection.

3.4. Comparison with existing techniques on the same hardware

Meaningful benchmarking calls for results obtained on the same boards. Table 4 therefore gathers the published studies, together with two widely used software tools, that report quantitative figures on either a Raspberry Pi 3/4 or a Cortex-M development kit, which are the two platforms examined in this work. By fixing the hardware, the table lets us separate the benefit that comes purely from observing EM leakage from the further gains delivered by our adaptive signal-processing pipeline.

All EM-based schemes leave the DUT untouched and so add no runtime cost. Even so, work on the RPi reports only a binary normal/abnormal flag, whereas the richest taxonomy on Cortex-M parts separates four classes. GJALLARHORN increases that coverage to sixteen CWE-aligned categories on the STM32 and keeps the same taxonomy on the RPi 3B.

The software baselines provide further context. Memcheck offers exhaustive dynamic-memory checking, yet it slows a RPi 3B programme by more than an order of magnitude and cannot run on bare-metal Cortex-M boards. AFL++ explores a broader bug surface, though each target needs a bespoke harness and long fuzzing campaigns. In contrast, GJALLARHORN delivers a multi-class verdict in under a minute without instrumenting the board, which suits iterative development workflows.

Overall, the EM-aligned figures place GJALLARHORN on a favourable point of the speed–coverage curve: it provides much finer diagnostics than earlier EM studies while avoiding the substantial overhead associated with heavyweight software tools.

4. GJALLARHORN framework

This paper introduces GJALLARHORN, a framework designed to evaluate the security of embedded systems during their developmental stages through EM SCA. Designed to support both constrained and complex embedded architectures, the framework provides a non-intrusive mechanism for detecting potential system weaknesses across diverse devices. This section delineates GJALLARHORN's conceptual foundations (Section 4.1), architectural implementation (Section 4.2), and operational mode (Section 4.3), focusing on its three interrelated modules–Data Acquisition, Data Preprocessing and Anomaly Detection–that transform raw EM emissions into vulnerability categories.

4.1. Conceptual overview

Emerging as a solution to pre-deployment security challenges, GJALLARHORN enables vulnerability assessment by examining EM signals generated during device operation. The framework provides a hardware-level view of software execution, capturing physical manifestations of vulnerabilities that remain invisible to purely software-based techniques.

Fig. 2 outlines its three sequential modules. During data acquisition, the system records raw EM emissions from the Device Under Test (DUT) as the workload runs. The resulting traces move to the data preprocessing stage, where filtering and down-sampling reduce complexity and expose salient features. Finally, the anomaly detection module analyses the processed signals and classifies patterns that indicate potential security weaknesses.

These interconnected modules create a flexible pipeline that transforms software inputs into vulnerability categories. By first establishing a baseline of normal system behaviour, and then comparatively

Table 4
Vulnerability-detection methods evaluated on hardware comparable with ours. "Granularity" is the number of classes (or fault types) distinguished at run-time.

Method	Device	Recall/ Accuracy	Runtime Overhead	Granularity
Sayakkara et al. (2019)	RPi 3B/4	>90%	None	2 (binary)
Wang et al. (2018)	Arduino/RPi 3B	90.00%	None	2 (binary)
Nazari et al. (2017)	A13-OLinuXino	97.04%	None	2 (binary)
Chawla et al. (2021)	Cortex-M4	98.60%	None	4
Memcheck	RPi 3B	N/R[a]	20–30× Slower[b]	Memory Errors
AFL++	RPi 3B	N/R[c]	≈0.8M Exec/s[d]	Crash/Hang/Ok
GJALLARHORN	NUCLEO-144	95.9%	None	16 (CWE)
	RPi 3B	73.3%	None	16 (CWE)

[a] Aggregate $F_1 \approx 72\,\%$ on synthetic workloads (Nong et al., 2021).
[b] Typical slowdown reported in the Valgrind manual (Valgrind Developers, 2025).
[c] Grey-box fuzzers are judged by the number of unique crashes (Guédou, 2017).
[d] Median throughput on an RPi 3B @ 1.2 GHz (Tickelton, 2020).

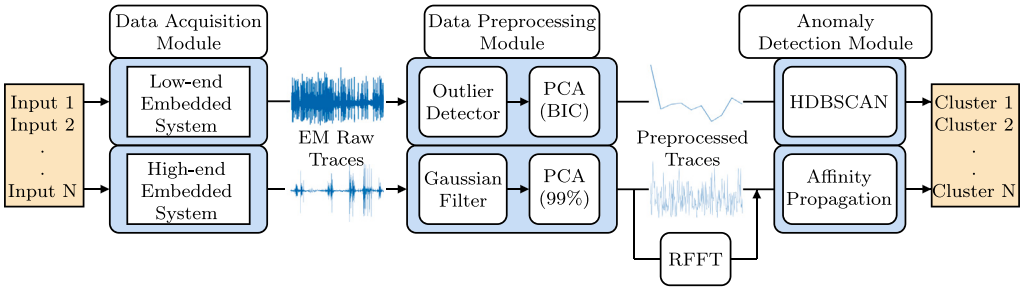


Fig. 2. Architectural design of GJALLARHORN.

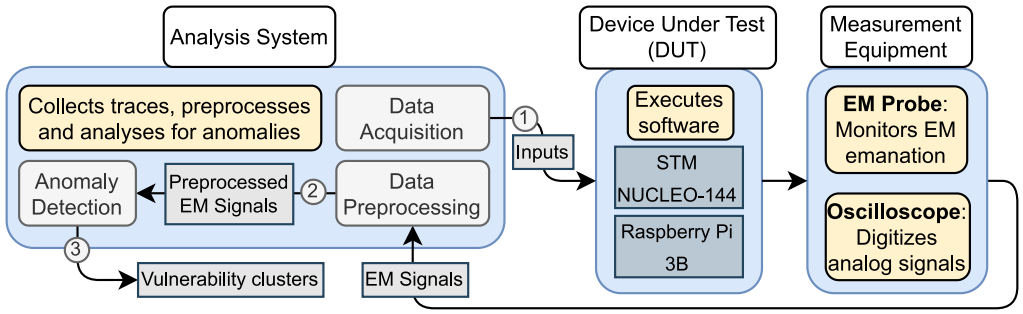


Fig. 3. Architectural implementation of GJALLARHORN.

analysing subsequent executions, GJALLARHORN provides a dynamic methodology for detecting potential security risks across diverse embedded systems.

4.2. Architectural implementation

The implementation of GJALLARHORN translates its conceptual framework into a methodical approach for EM SCA vulnerability detection.

As represented in Fig. 3, the implementation comprises the DUT, EM measurement equipment, and an analytical system housing the three core modules described in Section 4.1. These components operate sequentially, enabling non-invasive vulnerability identification across embedded system architectures.

The operational workflow begins with the Data Acquisition module, which captures raw EM traces emanating from the DUT during execution. These signals then pass to the Data Preprocessing module, which addresses noise issues and reduces dimensionality to prepare the data for analysis. Finally, the Anomaly Detection module examines the preprocessed data to identify and categorise potential vulnerabilities based on deviations from normal behaviour patterns.

This modular architecture enables GJALLARHORN to adapt to various embedded system platforms whilst maintaining a non-invasive approach to security assessment. Furthermore, the sequential processing methodology ensures thorough analysis without requiring code access or imposing computational overhead on the target system.

The following subsections detail each module's functionality and its contribution to vulnerability detection across diverse embedded systems.

4.2.1. Data acquisition module

The data acquisition module oversees two primary tasks: executing software on the DUT and collecting its corresponding EM emissions. This process is tailored to the device's control architecture, accommodating variations such as the presence or absence of an operating system. In the absence of an initial baseline, the framework first establishes a reference program representing the device's normal physical behaviour. Thus, the ability to detect EM anomalies does not depend on the specific functionality of this program, ensuring broad applicability across diverse embedded systems. To create a comprehensive dataset, modified versions of the reference program are developed, each embedding a specific vulnerability for analysis.

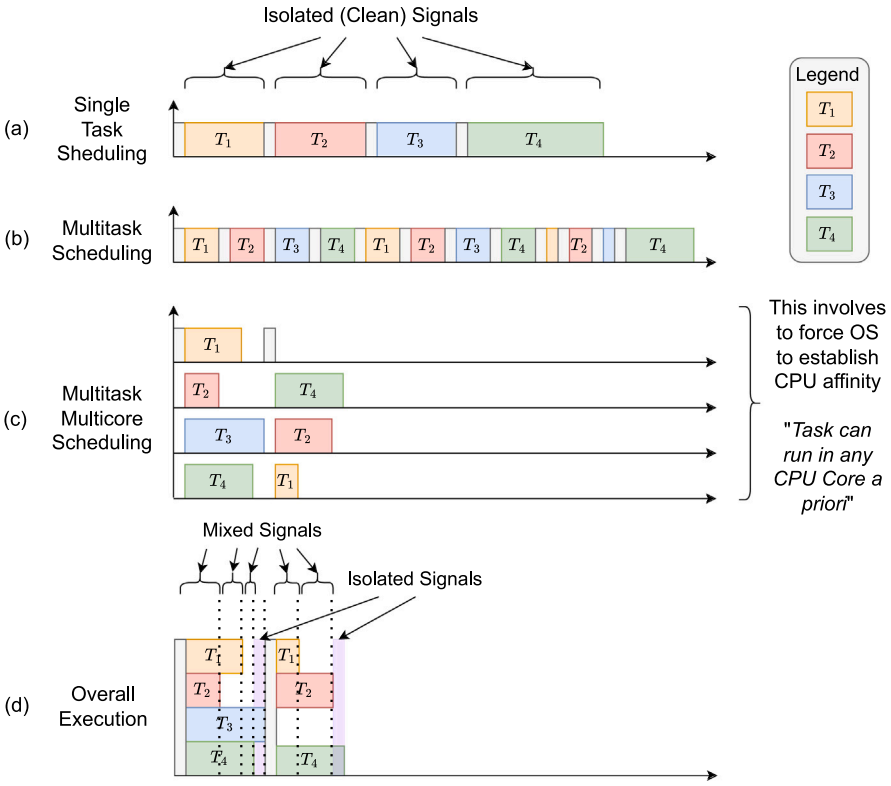


Fig. 4. Task scheduling in embedded systems: (a) Single Task Scheduling with sequential execution; (b) Multitask Scheduling with interleaved execution; (c) Multitask Multicore Scheduling across CPU cores; and (d) Overall Execution showing resulting signal patterns.

Data collection takes place in an electromagnetically isolated laboratory environment to minimise external interference. During execution, an EM probe is positioned in close proximity to the processor, capturing emissions that are recorded using an oscilloscope. The sampling rate is set at approximately ten times the device's maximum clock frequency, adhering to the Nyquist-Shannon theorem (Lindon et al., 2016) to ensure accurate signal reconstruction. While this sets the baseline, empirical studies in signal processing (Betta et al., 1999; Mangard et al., 2010) recommend oversampling at 5x to 20x for optimal signal quality versus computational cost.

The captured EM responses exhibit considerable variability across different embedded systems, which arise from differences in execution times, bandwidth limitations in the temporal domain (particularly in high-end systems), and the presence of EM signals unrelated to the software under test. Such differences stem primarily from the underlying hardware and software architecture, as well as inherent noise. For instance, Fig. 4 illustrates how low-end devices execute processes only when explicitly triggered, lacking resources for background tasks, whereas high-end systems manage concurrent operations across multiple cores, resulting in more complex EM emissions. This study evaluates these differences using a low-end STM NUCLEO-144 and a high-end RPi 3B, as defined in Section 2.1.

4.2.2. Data preprocessing module

The data acquisition phase generates a large volume of raw EM traces, which would be computationally demanding to analyse as they are. To address this, the preprocessing module refines these traces into simpler forms, keeping the key details needed for later analysis. Since EM emissions differ noticeably between low-end and high-end embedded systems, this module uses customised strategies, including noise filtering and dimensionality reduction, selected according to device complexity and the most suitable analysis domain (time or frequency). These methods, as our experiments have shown, deliver the best results for each system type.

Noise filtering. This step boosts the signal-to-noise ratio in each trace (Macnae et al., 1984), with the filtering method tailored to the system's complexity. For low-end systems, where emissions tend to be simpler but spikier, an outlier detector using the interquartile range (IQR) technique (H. P et al., 2018) spots samples outside the normal range and replaces them with linear interpolation to keep the signal intact. High-end systems, on the other hand, produce more complex emissions with richer frequency content. A Gaussian filter (Duda et al., 2001) is therefore preferable, as it smooths the noise while preserving valuable spectral details that an outlier detector might remove. In our tests, these approaches proved most effective.

Principal component analysis (PCA). After filtering, PCA cuts down the traces' dimensionality (Wold et al., 1987), with the number of components kept depending on the system. For low-end systems, the Bayesian Information Criterion (BIC) (Watanabe, 2012) picks the right number of components, striking a balance between explained variance and model complexity. For high-end systems, which carry more intricate signals, the Percentage of Explained Variance (PEV) method (Tamura and Tsujita, 2007) keeps components that account for 99% of the signal's variance, ensuring almost no information is lost. Our experiments confirmed these choices as optimal, with BIC reducing STM NUCLEO-144 data by 99.76% while maintaining detection accuracy above 95%, and PEV preserving the essential frequency features in RPi 3B signals with 94.79% dimensionality reduction.

The analysis domain also hinges on the embedded system's architecture. Low-end systems, with single-core processors and fixed clock speeds, emit EM traces that show clear peaks at the base frequency, so time-domain analysis is a solid fit. High-end systems, by contrast, run multiple cores and vary clock speed, producing more dynamic signals. For these devices, frequency-domain analysis with the Real Fast Fourier Transform (RFFT) (Sorensen et al., 1987) is the preferred choice. The RFFT focuses on non-negative frequencies because real-world EM signals only carry useful information there.

4.2.3. Anomaly detection module

After preprocessing, each EM response is categorised to identify distinct execution patterns. The framework groups similar responses into clusters representing specific execution flows, ranging from normal operation to potential vulnerabilities. By analysing deviations from normal behaviour clusters, GJALLARHORN detects anomalies indicating possible security weaknesses. This process, when conducted during development, enables early vulnerability identification and mitigation before deployment.

Given the absence of predefined classes, GJALLARHORN employs clustering, as supervised or semi-supervised methods are impractical without labelled data. The preprocessing strategies differ between low-end and high-end systems (time-domain for the former, frequency-domain for the latter), necessitating distinct clustering algorithms (Fahad et al., 2014):

- For low-end embedded systems, preprocessed EM responses from identical inputs tend to concentrate in localised regions of the data space. Here, density-based clustering, specifically HDBSCAN (McInnes et al., 2017), proves effective. HDBSCAN offers the advantage of identifying clusters with varying densities, including a cluster for noise samples.
- For high-end embedded systems, the presence of an OS with background tasks and workload distribution across multiple cores introduces greater variability in EM responses. Exemplar-based clustering, such as Affinity Propagation (Wang et al., 2013; Frey and Dueck, 2007), is better suited here. This method identifies representative samples (exemplars) that characterise each cluster, performing well with the frequency-domain representations of preprocessed traces.

Following each set of executions, clustering generates a variable number of clusters, ranging from two to *n*. Since this process occurs during the DUT's development, we can examine all resulting clusters and evaluate the effectiveness of the clustering algorithm. The performance assessment methodology, including metrics and analysis scenarios, is presented in Section 5.1.1.

4.3. Operational mode

The implementation of the GJALLARHORN modules follows a structured procedure to ensure effective anomaly detection. To overcome the challenge of establishing an initial reference for accurate fault categorisation, the framework operates in two distinct phases. The first phase calibrates the system by defining a baseline of normal behaviour, while the subsequent phase collects and analyses EM responses to assess the framework's ability to distinguish between normal and faulty executions.

4.3.1. Calibration phase

A calibrated reference is fundamental to reliable anomaly detection. By establishing a reference of the DUT's typical EM emissions, this phase enables the identification of deviations linked to potential vulnerabilities.

During calibration, multiple EM responses are collected from the DUT while it executes a program, as outlined in Section 4.2.1. This approach builds a robust reference set by accounting for variations in EM emissions, particularly in high-end embedded systems where background tasks may influence the signal. Additionally, idle signals are gathered to address cases where a vulnerability might not produce a distinct EM response. For each DUT, 100 normal behaviour signals and 100 idle signals are recorded, forming a comprehensive reference set for use in the operational phase.

4.3.2. Operational phase

In this phase, the framework compiles a blind dataset of EM responses by executing modified versions of the original program, each designed to probe specific vulnerabilities. To create a mixed dataset, additional responses from the original program executions are also included. All captured EM traces then undergo the preprocessing and clustering procedures described in Section 4.2.2 and Section 4.2.3. Traces not assigned to calibration clusters are categorised as anomalies, indicating potential faults within the system.

5. Evaluation

This section assesses the applicability of the GJALLARHORN methodology across embedded systems of varying complexity. Conducted during development, the evaluation begins by defining a reference program for each DUT to represent its normal behaviour (Section 5.2). Subsequently, a set of predefined vulnerabilities is introduced into these programs, creating modified versions for each fault type (Section 5.3).

5.1. Evaluation framework and adaptive techniques

To accommodate the differences between low-end and high-end embedded systems, GJALLARHORN adapts its preprocessing and analysis techniques based on the target architecture. Table 5 outlines the key differences in methodology for each system type. As shown, the framework opts for time-domain analysis with HDBSCAN clustering for the simpler STM NUCLEO-144, where emissions are steadier and easier to group, while it shifts to frequency-domain analysis via RFFT and Affinity Propagation clustering for the more complex RPi 3B, which produces more varied and dynamic signals. These choices reflect the need to match the analysis to the hardware's behaviour, ensuring GJALLARHORN can effectively spot vulnerabilities across diverse platforms.

To comprehensively assess GJALLARHORN's effectiveness across different system complexities, we established specific evaluation metrics and analysis scenarios, detailed in the following section. These metrics provide quantitative measures of the framework's capability to detect and classify anomalies in both low-end and high-end embedded systems.

5.1.1. Performance metrics and analysis scenarios

To evaluate the clustering results systematically, we generate a confusion matrix. This matrix compares the ground truth categories (vertical axis) with the algorithm-assigned categories (horizontal axis). Diagonal elements represent correctly categorised EM responses, while off-diagonal elements indicate miscategorisations: false negatives occur when faulty traces are assigned to calibration clusters, and false positives arise when normal traces are placed in incorrect error-type clusters. Additionally, some traces may be categorised into an Unassigned Anomalies (UAN) group, encompassing anomalies not associated with any specific error-type cluster. Based on this confusion matrix, we evaluate the framework's performance through three progressively challenging scenarios:

- **Anomaly Detection**: Distinguishes between calibration and faulty signals. Any trace diverging from the calibration clusters is categorised as anomalous, assuming that exploited vulnerabilities generate distinct EM emission patterns. This approach enables the identification of deviations from normal operation, irrespective of the specific fault type.
- **Arithmetic vs. Memory Error Detection**: Examines the framework's capability to differentiate between the two primary error categories outlined in the taxonomy–arithmetic and memory vulnerabilities. Using data from the confusion matrix, clustering recall is calculated for each error-type category, allowing the formation of higher-level "arithmetic" and "memory" clusters.

Table 5
Adaptive techniques for different hardware architectures in GJALLARHORN.

Component	Low-end embedded system (STM NUCLEO-144)	High-end embedded system (Raspberry Pi 3B)
Signal Domain	Time domain	Frequency domain via RFFT
Noise Filtering	IQR-based outlier detection	Gaussian filter
Dimensionality Reduction	PCA with BIC Criterion	PCA with PEV threshold (99%)
Clustering Algorithm	HDBSCAN	Affinity Propagation
Processing Time	37.15 s	64.74 s
Dimensionality Reduction	99.76% (from 10,000 to 24 samples)	94.79% (from 10,000 to 521 samples)

- **Specific Error Detection**: This focuses on identifying individual error types with precision. Diagonal elements in the confusion matrix (true positives) reflect traces correctly assigned to their specific error category, whereas off-diagonal elements indicate miscategorisations—either false negatives, where faulty traces are grouped with calibration clusters, or false positives, where traces are assigned to incorrect error types. This scenario measures GJALLARHORN's accuracy in pinpointing distinct vulnerabilities.

For each scenario, the effectiveness of clustering is evaluated using five metrics: recall, precision, F1 score, micro-F1 score, and Matthews Correlation Coefficient (MCC), commonly employed for assessing clustering (Flach and Kull, 2015).

Recall. Computes the proportion of true positives (TP) among the true positive and false negative (FN) predictions.

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

Precision. Evaluates the proportion of true positives (TP) among the true positive and false positive (FP) predictions.

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

F_1 *score.* Measures the harmonic mean of precision and recall.

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

Micro-F_1 score. Evaluates the clustering performance across the predicted classes (Harbecke et al., 2022).

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

Matthews correlation coefficient (MCC). By considering all confusion matrix elements, it is widely employed to evaluate imbalanced datasets (Chicco and Jurman, 2020).

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

These metrics allow us to evaluate GJALLARHORN's ability to distinguish normal executions from faulty ones and to categorise different types of vulnerabilities. The following subsections present the results of this evaluation on both low-end (STM NUCLEO-144) and high-end (RPi 3B) systems.

5.2. Implementation of a reference program

The evaluation starts by establishing a reference program to serve as a benchmark for normal system behaviour. For this purpose, two programs were developed to solve equations: one using integer arithmetic (denoted as SUT00I) and another using floating-point arithmetic (SUT00F). These variants allow an assessment of GJALLARHORN's capability to differentiate between executions that differ solely in their arithmetic operations, providing insight into its sensitivity to subtle computational variations.

5.3. Taxonomy of vulnerabilities in embedded systems

Embedded systems, integrating physical and digital domains, exhibit distinct physical responses to software faults, often detectable via SCA. To assess SCA for vulnerability detection, we propose a taxonomy classifying faults into two primary categories: arithmetic, arising from numerical operations, and memory-related, linked to memory management instructions. Table 6 lists 16 specific vulnerabilities, six arithmetic and ten memory-related, selected for their prevalence and alignment with the Common Weakness Enumeration (CWE) database (MITRE Corporation, 2025), each identified by its CWE code. These faults are embedded into reference programs to generate faulty versions for evaluation across diverse devices.

The taxonomy's effectiveness depends on system complexity. Low-end systems with minimal error-handling often yield pronounced SCA signals, while high-end systems with robust mitigation mechanisms may obscure certain faults. This distinction guides the analysis of EM signatures across the selected devices, as detailed in subsequent sections.

5.4. Implementation details

To ensure reproducibility, we have released the source code and proof-of-concept implementations for both use cases.[1] The signal acquisition setup consists of specific configurations for each DUT, with precise probe positioning and environmental controls to maintain consistent results.

For signal preprocessing, we implemented the pipeline described in Algorithm 1, which adapts to the complexity of the target device.

Algorithm 1 EM Signal Preprocessing Pipeline

Input: Raw EM trace *signal*, device type *device_type*
Output: Preprocessed signal

```
function PreprocessSignal(signal, device_type)
    if device_type = "low-end" then
        clean_signal ← OutlierDetector(signal, IQR_threshold = 1.5)
        components ← PCA(clean_signal)
        n_components ← SelectByBIC(components)
        reduced_signal ← components[0 : n_components]
        output_signal ← reduced_signal
    else if device_type = "high-end" then
        clean_signal ← GaussianFilter(signal, σ = 2.0)
        components ← PCA(clean_signal)
        n_components ← SelectByPEV(components, threshold = 0.99)
        reduced_signal ← components[0 : n_components]
        fft_signal ← RFFT(reduced_signal)
        output_signal ← fft_signal
    end if
    return output_signal
end function
```

The H-Field probe was positioned 2 mm above the processor package for all experiments after conducting sensitivity analysis to identify

Table 6
Weaknesses used to characterise the behaviour of embedded systems.

Type	Code	Name	Definition	Related CWE Code
Arithmetic	E0101	Floating Point Overflow	Value exceeds maximum representable floating-point magnitude.	CWE-681
	E0102	Floating Point Underflow	Non-zero result smaller than minimum representable floating-point value.	CWE-681
	E0103	Integer Overflow	Result exceeds maximum capacity of an integer variable.	CWE-190
	E0104	Integer Underflow	Result falls below minimum capacity of an integer variable.	CWE-191
	E0105	Divide by Zero Integer	Division operation with a zero integer divisor.	CWE-369
	E0106	Divide by Zero Decimal	Division operation with a zero decimal divisor.	CWE-369
Memory	E0201	Segmentation Fault	Unauthorized memory access violating OS protection boundaries.	CWE-119
	E0202	Buffer Overflow	Writing beyond allocated memory bounds, corrupting adjacent data.	CWE-120
	E0203	Double Free	Attempting to deallocate already freed memory.	CWE-415
	E0204	Null Pointer Dereference	Accessing memory through an uninitialized pointer.	CWE-476
	E0205	Out-of-Bounds Write	Writing data beyond allocated array boundaries.	CWE-787
	E0206	Out-of-Bounds Read	Reading data from outside allocated array boundaries.	CWE-125
	E0207	Out-of-Memory	Memory allocation failure due to resource exhaustion.	CWE-400
	E0208	Stack Overflow	Call depth or variable allocation exceeding stack memory limits.	CWE-121
	E0209	Stack Underflow	Operation on an empty or insufficient stack.	CWE-124
	E0210	Unaligned Address	Memory access at addresses not conforming to hardware alignment requirements.	CWE-188

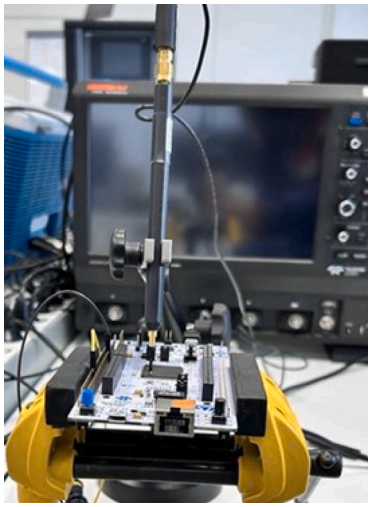


Fig. 5. Data acquisition setup for STM NUCLEO-144.

the position with optimal signal strength. For the STM NUCLEO-144, the Langer RF-B 0.3-3 H-Field Probe was placed directly above the STM32F429ZI processor, while for the RPi 3B, the Langer RF-R 0.3-3 H-Field Probe was positioned over the BCM2837 SoC. The complete implementation follows the architecture described in the previous sections, with specific parameters tuned for each device based on empirical testing.

5.5. Low-end embedded system: STM NUCLEO-144

The STM NUCLEO-144 is a development board from STMicroelectronics, whose architecture was selected for its prevalence in industrial and IoT applications and significant market share (Raje, 2025). It features a 32-bit ARM Cortex-M4 single-core processor operating at 180 MHz without a full operating system. It executes programs in isolation, free from background task interference, making it a controlled scenario for time-domain clustering.

5.5.1. Parametrisation

Upon startup, the device enters a waiting state, ready to process execution requests. When triggered, the processor runs the requested program at its base clock frequency until completion, then returns to idle. For this evaluation, 20 instances of each fault type from Table 6 are executed, alongside 20 instances of SUT00I and SUT00F. Combined with calibration traces (Section 4.3), these signals undergo preprocessing and clustering.

EM responses are captured using a Langer RF-B 0.3-3 H-Field Probe positioned above the processor, recorded by a LeCroy WaveRunner 8104 oscilloscope at 1 GS/s—approximately ten times the clock frequency (Fig. 5). Each trace spans a 50-microsecond window across 10,000 samples (Fig. 6).

5.5.2. Results

Since the evaluation occurs during development, ground truth data links each EM response to its fault type, enabling a detailed cluster analysis via a confusion matrix (Fig. 7). Two distinct calibration clusters emerge—one for SUT00I and one for SUT00F. The remaining clusters represent anomalies. The preprocessing and clustering process averaged 37.15 s.

Anomaly Detection. The framework achieves a recall of 95.94% and an F_1 score of 97.93% for faulty traces, with 100% recall and an F_1 score of 86.02% for normal traces (Table 7). The micro-F_1 score of 96.39% and MCC of 85.09% indicate high overall accuracy, with minimal UANs. This suggests that time-domain analysis, leveraging HDBSCAN, effectively captures the distinct EM signatures of faults in single-core systems.

Arithmetic vs. Memory Error Detection. Arithmetic errors show a recall of 78.33% (F_1: 87.85%), while memory errors achieve 91.50% (F_1: 90.59%), with a micro-F_1 score of 88.06% and MCC of 79.98% (Fig. 8). Hence, 18% of arithmetic traces are outliers, likely due to their subtler EM impact compared to memory faults, which disrupt execution more significantly. This disparity aligns with the hypothesis that memory-related anomalies produce stronger physical responses, enhancing their detectability.

Specific Error Detection. Of the 16 error types, 14 are identified, with 12 exceeding 90% recall (e.g., E0103: 100%, E0201: 95%). However, E0106 (divide by zero decimal) and E0202 (buffer overflow) exhibit 0% recall, misclustered with E0210 (unaligned address) and E0205 (out-of-bounds write), respectively. This indicates that certain faults with overlapping EM patterns—possibly due to similar execution paths or minimal timing differences—challenge fine-grained differentiation in the time domain.

These findings confirm GJALLARHORN's robustness for low-end embedded systems, where consistent EM emissions enable high anomaly detection rates and reliable error type categorisation. The misclustering of E0106 and E0202 suggests that time-domain analysis may require supplementary features (e.g., higher-resolution sampling) to resolve subtle fault distinctions, particularly for arithmetic errors with low physical impact.

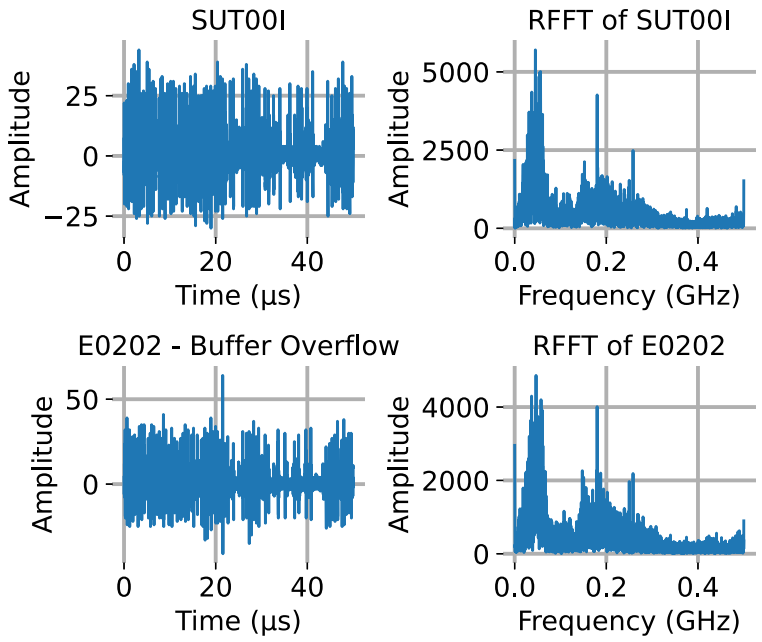


Fig. 6. STM NUCLEO-144: Time-domain and frequency-domain representations of normal (above) and faulty (below) executions. The faulty signal exhibits a distinct peak at approximately 22 µs, distinguishing it from the normal signal and highlighting the utility of time-domain analysis for anomaly detection in low-end systems.

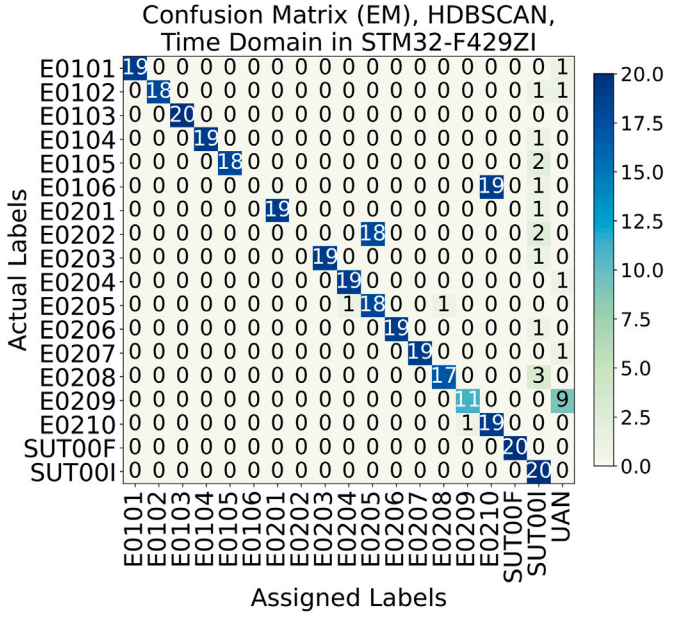


Fig. 7. STM NUCLEO-144: Confusion Matrix.

5.6. High-end embedded system: Raspberry Pi 3B

The RPi 3B features a 1.2 GHz quad-core ARM Cortex-A53 processor running Raspberry Pi OS. Its quad-core setup supports simultaneous task execution, which introduces noise into EM responses and makes it difficult to eliminate unrelated EM interference. The differing clock frequencies across cores highlight the need for frequency-domain analysis.

5.6.1. Parametrisation

After booting, the operating system manages resources and accepts execution requests. Programs are split into tasks distributed across the four cores, as shown in Fig. 4. For this evaluation, 20 instances of faulty software for each error from Table 6 are executed, as well as 20 instances of each original program's integer and floating-point versions. Concatenated with the calibration traces, their EM responses undergo preprocessing and clustering. Each trace not assigned to a calibration group is considered an anomaly.

The EM emissions are captured by a Langer RF-R 0.3-3 H-Field Probe, which is connected to the LeCroy WaveRunner 8104 oscilloscope (Fig. 9). The oscilloscope's sampling rate is 10 GS/s, approximately ten times the board's clock frequency (1.2 GHz), ensuring faithful signal reconstruction. The waveforms represent a time window of 1 microsecond in 10,000 samples (Fig. 10). We could not trigger faults related to error E0210 (unaligned address) on the RPi 3B, as its ARM Cortex-A53 processor handles unaligned memory accesses, unlike the STM32 microcontroller in the NUCLEO-144. Consequently, its exploitation was limited to the STM NUCLEO-144, where unaligned accesses generate hardware exceptions that can be manipulated.

5.6.2. Results

High-end systems perform background tasks and possess intricate hardware, resulting in increased noise in their EM emissions. For clustering in these devices, the clusters from both calibration programs are grouped as a single calibration cluster (SUT00), while all other clusters are identified as anomalies, as shown in the confusion matrix (Fig. 11). The preprocessing and clustering of calibration and operational signals required an average of 64.74 s, indicating the greater complexity of high-end systems compared to low-end ones. The matrix allows analysis of the three scenarios:

Anomaly Detection. The framework records a recall of 73.33% (F_1: 84.61%) for faulty traces and 100% (F_1: 50.00%) for normal ones, yielding a micro-F_1 score of 76.47% and MCC of 49.44% (Table 8). While precision remains 100% for anomalies, the lower recall reflects interference from multicore task scheduling, reducing EM consistency compared to low-end systems.

Arithmetic vs. Memory Error Detection. Arithmetic errors exhibit a recall of 5% (F_1: 8.63%), while memory errors reach 54.44% (F_1: 68.05%), with a micro-F_1 score of 42.35% and MCC of 30.50%

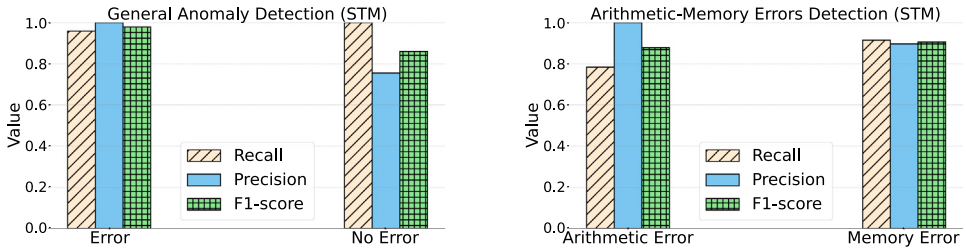


Fig. 8. STM NUCLEO-144: Metrics for Anomaly Detection and Arithmetic vs. Memory Error Detection.

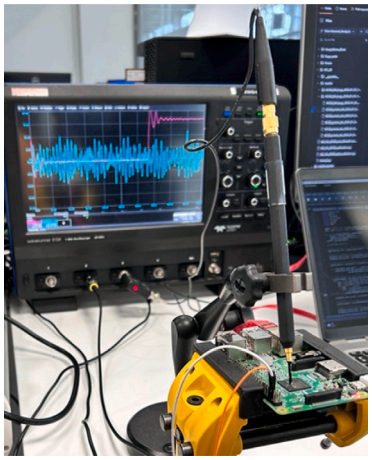


Fig. 9. Data acquisition setup for RPi 3B.

Table 7
Clustering results for STM NUCLEO-144.

Anomaly detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Error	95.94	100.00	97.93	96.39	85.09
No Error	100.00	75.47	86.02		
Arith. vs. Memory Error Detection	**Recall (%)**	**Precision (%)**	**F_1 (%)**	**Micro-F_1 (%)**	**MCC (%)**
Arithmetic Error	78.33	100.00	87.85	88.06	79.98
Memory Error	91.50	89.70	90.59		
Specific Error Detection	**Recall (%)**	**Precision (%)**	**F_1 (%)**	**Micro-F_1 (%)**	**MCC (%)**
E0101	95.00	100.00	97.44		
E0102	90.00	100.00	94.74		
E0103	100.00	100.00	100.00		
E0104	95.00	100.00	97.44		
E0105	90.00	100.00	94.74		
E0106	0.00	0.00	0.00		
E0201	95.00	100.00	97.44		
E0202	0.00	0.00	0.00	81.67	81.23
E0203	95.00	100.00	97.44		
E0204	95.00	95.00	95.00		
E0205	90.00	50.00	64.23		
E0206	95.00	100.00	97.44		
E0207	95.00	100.00	97.44		
E0208	85.00	94.00	89.47		
E0209	55.00	91.00	68.75		
E0210	95.00	50.00	65.52		

(Fig. 12). Memory faults cluster into two distinct groups, suggesting detectable physical responses despite noise, whereas arithmetic errors are largely obscured, likely due to overlap with background EM activity.

Specific Error Detection. Recall for specific errors is generally low, with only E0207 (45%) and E0204 (25%) exceeding 0%, and most at 0% (e.g., E0101: 5%, E0202: 0%). The micro-F_1 score of 17.05% and MCC of 13.82% indicate significant challenges in isolating individual fault signatures, as noise and core variability dilute distinctiveness.

The reduced performance on high-end embedded systems underscores the limitations of frequency-domain analysis (via Affinity Propagation, Section 4.2.3) in noisy environments. While anomaly detection remains viable, the framework struggles to differentiate specific errors, suggesting that multicore systems require enhanced noise filtering or multi-dimensional feature extraction to improve fault resolution.

5.7. Overview of GJALLARHORN's performance

Tables 7 and 8 highlight distinct performance levels across hardware architectures. The STM NUCLEO-144 achieves a micro-F_1 score of 96.39% for anomaly detection and 88.06% for error type detection, with MCCs of 85.09% and 81.23%, reflecting robust performance in low-end systems. In contrast, the RPi 3B records lower micro-F_1 scores of 76.47% and 42.35%, with MCCs of 49.44% and 13.82%. This performance gap stems from the complexities of multicore architectures in high-end systems.

Furthermore, the RPi 3B's quad-core ARM Cortex-A53 processor, operating at 1.2 GHz, interleaves tasks across cores with affinity constraints, leading to signal mixing between the software under test, OS kernel operations, and background processes (Chawla et al., 2021). This intermixing results in overlapping EM emissions, reducing the signal-to-noise ratio and complicating the isolation of fault-specific signatures.

Arithmetic errors exhibit a recall of 5% compared to 54.44% for memory errors in the RPi 3B. Memory errors, such as buffer overflows (E0202), cause significant control flow disruptions, altering instruction pipelines and memory access patterns, which manifest as pronounced EM peaks detectable even amidst noise. Arithmetic errors, however, like floating-point underflows (E0102), produce subtler EM variations.

On the evaluated RPi 3B, the ARM Cortex-A53 processor handles arithmetic operations without triggering hardware exceptions unless explicitly configured. For instance, integer division by zero (E0105) generates a SIGFPE signal, ending the program with a "Floating point exception" message, whilst floating-point operations, following IEEE 754 standards, yield infinity or NaN values without interrupts (IEEE, 2019). These operations result in minimal changes to the processor's power profile, producing weak EM signatures that are often masked by the 1.2 GHz base clock frequency. In contrast, the STM NUCLEO-144's ARM Cortex-M4 processor, operating at 180 MHz, halts execution on such errors, causing clear EM peaks in the time domain, achieving a higher recall of 78.33% for arithmetic errors.

Compared to traditional methodologies, GJALLARHORN provides a non-invasive approach. Static analysis detects approximately 70%–80% of memory issues (Chess and West, 2007) but misses 30% of runtime faults. Dynamic analysis achieves 85%–90% detection rates (Abadi and Fournet, 2001) but introduces 20%–200% performance overhead, impractical for constrained devices. Fuzzing covers 80%–95% of vulnerabilities (Miller et al., 1990) but demands extended execution times. GJALLARHORN, with a 91.50% recall for memory errors in low-end

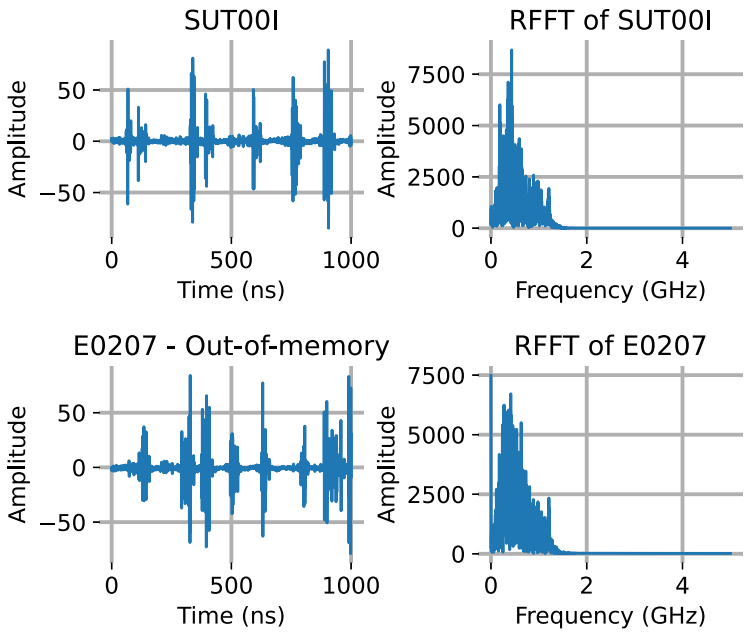


Fig. 10. RPi 3B: Time- and frequency-domain plots of normal (top) and faulty (bottom) executions. Time-domain signals appear similar, highlighting the value of frequency-domain analysis for complex systems.

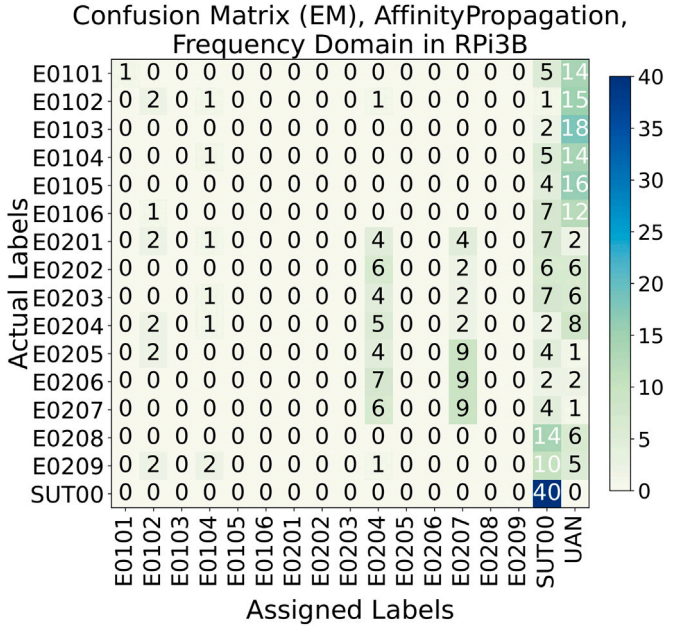


Fig. 11. RPi 3B: Confusion Matrix.

Table 8
Clustering results for Raspberry Pi 3B.

Anomaly detection	Recall (%)	Precision (%)	F_1 (%)	Micro-F_1 (%)	MCC (%)
Error	73.33	100.00	84.61	76.47	49.44
No Error	100.00	33.33	50.00		
Arith. vs. Memory Error Detection	**Recall (%)**	**Precision (%)**	**F_1 (%)**	**Micro-F_1 (%)**	**MCC (%)**
Arithmetic Error	5.00	31.58	8.63	42.35	30.50
Memory Error	54.44	90.74	68.05		
Specific Error Detection	**Recall (%)**	**Precision (%)**	**F_1 (%)**	**Micro-F_1 (%)**	**MCC (%)**
E0101	5.00	100.00	9.52		
E0102	10.00	18.00	12.90		
E0103	0.00	0.00	0.00		
E0104	5.00	14.00	7.41		
E0105	0.00	0.00	0.00		
E0106	0.00	0.00	0.00		
E0201	0.00	0.00	0.00		
E0202	0.00	0.00	0.00	17.05	13.82
E0203	0.00	0.00	0.00		
E0204	25.00	13.00	17.24		
E0205	0.00	0.00	0.00		
E0206	0.00	0.00	0.00		
E0207	45.00	24.00	31.58		
E0208	0.00	0.00	0.00		
E0209	0.00	0.00	0.00		
E0210	N/A	N/A	N/A		

systems, complements these methods by leveraging EM emissions without imposing computational overhead on the device.

6. Discussion

This section examines GJALLARHORN's evaluation across embedded systems, focusing on its applicability, potential attack vectors and limitations.

6.1. Applicability and real-world relevance

GJALLARHORN adapts to diverse hardware complexities, as demonstrated on STM NUCLEO-144 and Raspberry Pi 3B (Johnson et al., 2021). Low-end systems with single-core designs and fixed clock frequencies produce consistent EM emissions. Hence, the STM NUCLEO-144's time-domain analysis, utilising HDBSCAN clustering, effectively identifies anomalies in these stable patterns. Conversely, high-end systems like the RPi 3B, featuring a quad-core ARM Cortex-A53 processor with dynamic frequency scaling (600 MHz to 1.2 GHz), generate variable EM emissions. Furthermore, the Linux-based RPi OS introduces periodic tasks that overlap with the software under test, thus complicating

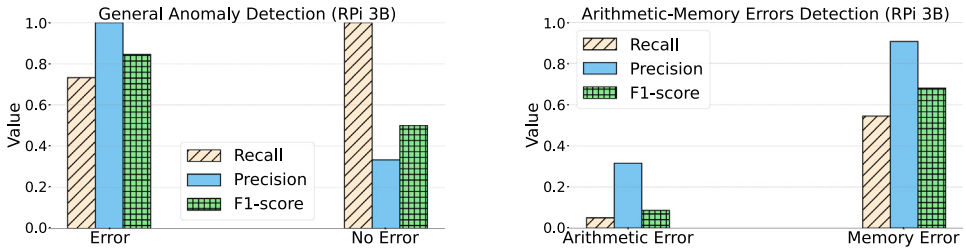


Fig. 12. RPi 3B: Metrics for Anomaly Detection and Arithmetic vs. Memory Error Detection.

fault detection (Chawla et al., 2021). This necessitates advanced pre-processing, such as RFFT-based frequency-domain analysis, to isolate key frequency components.

Hardware and OS interactions notably impact high-end systems. The RPi 3B's Cortex-A53 lacks comprehensive hardware-level exception handling for most arithmetic operations without specific compiler flags, resulting in silent failures. For instance, floating-point operations producing infinity or NaN values minimally alter the processor's execution state, generating EM variations below the Langer Probe's detection threshold. Additionally, memory errors, whilst more detectable, are affected by background noise in multicore systems. In contrast, the STM NUCLEO-144's bare-metal configuration ensures memory errors directly impact the processor's instruction pipeline, creating distinct EM signatures without OS interference.

GJALLARHORN's hardware-level view, focusing on EM emissions, provides insights that complement software-based methods, enabling the detection of vulnerabilities through their physical manifestations. The processing time of 37–65 s, whilst limiting continuous monitoring deployment, is mitigated by the framework's dimensionality reduction (>94%), achieved through PCA and RFFT, which reduces computational complexity. Further optimisation via parallelisation could enhance scalability, positioning GJALLARHORN as a valuable tool for non-contact vulnerability assessment, distinct from power-based SCA (Kocher et al., 1999).

6.2. Attack vectors and countermeasures

This framework assumes attackers have limited knowledge of target EM signatures but may employ evasion strategies. Code polymorphism could alter execution patterns, disrupting EM signatures (Couroussé et al., 2016), whilst hardware shielding might reduce EM leakage (Das and Sen, 2020). Artificial noise injection, injecting signals at frequencies overlapping with the processor's clock harmonics, could obscure fault signatures (Trippel et al., 2017), particularly in high-end systems where background noise is prevalent.

Advanced adversaries might deploy metamorphic malware, timing randomisation to desynchronise EM emissions, or signature mimicry to emulate normal behaviour. However, several practical countermeasures can significantly reduce the effectiveness of EM-based vulnerability detection without requiring hardware redesign.

6.2.1. Physical countermeasures

Probe distancing. In the magnetic near field the coupling strength decays with approximately $1/r^3$ (Ott, 2009). Empirical work shows that increasing the probe gap from 4–5 mm to 10 cm leads to an SNR reduction of roughly 40 dB and drops key-recovery success rates from above 90% to below 30% on ARM-A53 SoCs (Longo et al., 2015). Our own measurements confirm this trend: repeating the full anomaly/error-type pipeline with the probe placed at 10 cm, 50 cm and 1 m yielded micro-F_1 scores of 28%, 13% and <5%, respectively, in line with the attenuation predicted by the $1/r^3$ model. Far-field studies on Cortex-M4 boards similarly report that extending the distance from 6 m to 15 m demands about ten times more traces to sustain accuracy (Wang et al., 2020).

Electromagnetic shielding. Continuous conductive enclosures may contribute 42–70 dB of attenuation in the 1 MHz–1 GHz band (Dong et al., 2023), although a 1 mm slot can cut that figure by about 28 dB (Armstrong, 2009). Tests with 0.5 mm copper sheet report 25–48 dB of absorption, while sub-millimetre mesh cages typically shave ≈23 dB off the SNR for microcontroller targets (Wu et al., 2024). Combining a 10 cm probe offset (≈60 dB practical) with lightweight shielding (+40–50 dB) therefore promises more than 90 dB overall; in forthcoming work we will enclose the STM32 board in a copper shield (≈45 dB) and quantify the combined impact on all sixteen vulnerability classes. For space-constrained products, we will also evaluate epoxy potting, which eliminates seams at the cost of one-way assembly and a modest thermal penalty.

Beyond these physical barriers, multi-channel analysis that fuses EM and power-rail data (Kocher et al., 1999), together with enhanced wavelet-based noise filtering (Hospodar et al., 2011) and dynamic calibration, can further harden systems against sophisticated, adaptive adversaries.

6.3. Limitations and broader considerations

Detection reliability varies with system complexity, with single-core boards achieving higher precision because their EM signatures are clearer. Consequently, GJALLARHORN is presently most effective in single-core devices, even those running lightweight operating systems, where concurrent processes exert less influence on emissions. In multi-core environments, signal inter-mixing from several cores and OS tasks significantly degrades accuracy.

The proposed framework already addresses sixteen CWE-aligned vulnerabilities, yet it does not encompass emerging threats such as speculative-execution attacks (Kocher et al., 2019), whose cache-centric patterns differ markedly. Furthermore, the current 37–65s processing latency per 340 traces limits true continuous monitoring and therefore calls for optimisation. Physical counter-measure evaluation offers an immediate path: distancing the probe to 10 cm trims the SNR by ≈60 dB and depresses micro-F_1 to 28%; enclosing the board in a lightweight Faraday cage is expected to add 42–70 dB, provided seams and apertures remain < 1 mm (Dong et al., 2023; Armstrong, 2009). The resulting > 90 dB attenuation should push classifier AUC below 0.60 with trace counts under 10,000; forthcoming experiments on the STM32 and Raspberry Pi will verify these projections across all sixteen vulnerability classes.

Experimental analysis also suggests that advanced signal aggregation, such as multi-resolution wavelet decomposition or broader time–frequency analysis (Hospodar et al., 2011), could mitigate the accuracy loss seen in multicore systems. Decomposing EM traces into separate bands may isolate anomalies in noisy contexts and extend applicability to medium-end platforms such as the ESP32 or ARM Cortex-M7.

A balanced approach that merges GJALLARHORN's hardware-level insight with software-based countermeasures will thus provide comprehensive protection for embedded systems, combining complementary strengths while respecting practical deployment constraints.

7. Conclusion

This study explores the potential of EM analysis for early vulnerability detection in embedded systems. Whilst SCA has been applied to IoT, its scope has typically been limited to specific devices and functionalities. To address this, we introduce GJALLARHORN, a framework using EM leakage to categorise signals and detect anomalies linked to potential vulnerabilities. By offering a hardware-level view of software execution, it complements software-based security methods, identifying vulnerabilities through their EM signatures rather than relying solely on code or runtime analysis. Operating independently of the software under test, GJALLARHORN adapts readily to diverse embedded systems.

Evaluations on the low-end STM NUCLEO-144 and high-end Raspberry Pi 3B demonstrate robust performance, achieving micro-F_1 scores of 96.39% and 76.47% for anomaly detection, respectively. The framework identifies diverse bugs across 16 CWE-aligned vulnerabilities, achieving 100% precision for anomalies on both platforms and 89.70%–90.74% for error type differentiation. These results indicate EM-based SCA's applicability for security evaluation during development, enabling the mitigation of vulnerabilities before deployment. To facilitate adoption and extension of our approach, we have made the complete framework implementation and proof-of-concept code publicly available. Beyond development, its effectiveness in noisy environments suggests utility for periodic security monitoring.

Future research could enhance GJALLARHORN in several directions. Firstly, extending evaluation to medium-end platforms (such as ESP32 or ARM Cortex-M7 systems) would establish a performance spectrum across the embedded systems hierarchy. Secondly, multi-channel signal fusion combining electromagnetic, power, and timing data could improve detection robustness in noisy environments, particularly for high-end systems where background tasks obscure subtle signatures. Evaluating GJALLARHORN against emerging attack vectors such as speculative execution vulnerabilities would expand its applicability to contemporary threats. Improvements to the preprocessing pipeline, specifically frequency-domain feature extraction optimised for arithmetic operations, could address the observed performance gap in high-end systems. Additionally, industrial case studies in regulated sectors could validate the framework's alignment with standards like the EU's Cyber Resilience Act, thus establishing a pathway for practical adoption in critical infrastructure protection.

CRediT authorship contribution statement

Jorge Barredo: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation. **Maialen Eceiza:** Writing – review & editing, Visualization, Validation, Methodology, Conceptualization. **Jose Luis Flores:** Writing – review & editing, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Mikel Iturbe:** Writing – review & editing, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mikel Iturbe reports financial support was provided by Basque Government. Jorge Barredo reports financial support was provided by Spain Ministry of Science, Innovation and Universities. Maialen Eceiza reports financial support was provided by Spain Ministry of Science, Innovation and Universities. Mikel Iturbe reports financial support was provided by Spain Ministry of Science, Innovation and Universities. Mikel Iturbe reports a relationship with Basque Government that includes: funding grants. Jorge Barredo reports a relationship with Spain Ministry of Science, Innovation and Universities that includes: funding grants. Maialen Eceiza reports a relationship with Spain Ministry of Science, Innovation and Universities that includes: funding grants. Mikel Iturbe reports a relationship with Spain Ministry of Science, Innovation and Universities that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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

Data availability

We share a link to a Github repository containing the source code and multiple PoCs throughout the manuscript.

References

Abadi, M., Fournet, C., 2001. Mobile values, new names, and secure communication. In: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. POPL '01, Association for Computing Machinery, New York, NY, USA, pp. 104–115. http://dx.doi.org/10.1145/360204.360213.

Alrawi, O., Lever, C., Antonakakis, M., Monrose, F., 2019. SoK: Security evaluation of home-based IoT deployments. In: 2019 IEEE Symposium on Security and Privacy. SP, pp. 1362–1380. http://dx.doi.org/10.1109/SP.2019.00013.

Armstrong, K., 2009. EMC Techniques in Electronic Design, Part 4: Shielding (Screening). Technical Report, Cherry Clough Consultants, URL: https://www.emcstandards.co.uk/files/part_4_text_and_graphics_21_may_09.pdf.

Batina, L., Jauernig, P., Mentens, N., Sadeghi, A.-R., Stapf, E., 2019. In hardware we trust: Gains and pains of hardware-assisted security. In: Proceedings of the 56th Annual Design Automation Conference 2019. DAC '19, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/3316781.3323480.

Betta, G., Liguori, C., Pietrosanto, A., 1999. Structured approach to estimate the measurement uncertainty in digital signal elaboration algorithms. IEE Proc. - Sci. Meas. Technol. 146, 21–26. http://dx.doi.org/10.1049/ip-smt:19990001.

Bouij-Pasquier, I., Kalam, A., Ouahman, A., Montfort, M., 2015. A Security Framework for Internet of Things. pp. 19–31. http://dx.doi.org/10.1007/978-3-319-26823-1_2.

Camposano, R., Wilberg, J., 1996. Embedded system design. Des. Autom. Embedded Syst. 1, 5–50. http://dx.doi.org/10.1007/BF00134682.

Chawla, N., Kumar, H., Mukhopadhyay, S., 2021. Machine learning in wavelet domain for electromagnetic emission based malware analysis. IEEE Trans. Inf. Forensics Secur. PP, http://dx.doi.org/10.1109/TIFS.2021.3080510, 1–1.

Chess, B., West, J., 2007. Secure Programming with Static Analysis. Addison-Wesley Professional.

Chicco, D., Jurman, G., 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21, http://dx.doi.org/10.1186/s12864-019-6413-7.

Council of European Union, 2022. 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). Technical Report, Council of European Union, URL: https://eur-lex.europa.eu/eli/dir/2022/2555/oj.

Couroussé, D., Barry, T., Robisson, B., Jaillon, P., Potin, O., Lanet, J.-L., 2016. Runtime code polymorphism as a protection against side channel attacks. In: Foresti, S., Lopez, J. (Eds.), Information Security Theory and Practice. Springer International Publishing, Cham, pp. 136–152. http://dx.doi.org/10.1007/978-3-319-45931-8_9.

Das, D., Sen, S., 2020. Electromagnetic and power side-channel analysis: advanced attacks and low-overhead generic countermeasures through white-box approach. Cryptography 4 (4), http://dx.doi.org/10.3390/cryptography4040030.

Dong, Q., Adams, Z., Watkins, R., Chang, C.-M., Lee, B., Levin, C., 2023. Investigation of Faraday cage materials with low eddy current and high RF shielding effectiveness for PET/MRI applications. Phys. Med. Biol. 68, http://dx.doi.org/10.1088/1361-6560/acdec4.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification. Wiley, pp. 15–17.

Eceiza, M., Flores, J.L., Iturbe, M., 2021. Fuzzing the Internet of Things: A review on the techniques and challenges for efficient vulnerability discovery in embedded systems. IEEE Internet Things J. 8 (13), 10390–10411. http://dx.doi.org/10.1109/JIOT.2021.3056179.

European Commission, 2022. 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. Technical Report, European Commission, URL: https://www.cyberresilienceact.eu/the-cyber-resilience-act/.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S., Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE Trans. Emerg. Top. Comput. 2 (3), 267–279. http://dx.doi.org/10.1109/TETC.2014.2330519.

Fernández-Caramés, T.M., Fraga-Lamas, P., Suárez-Albela, M., Castedo, L., 2017. Reverse engineering and security evaluation of commercial tags for RFID-based IoT applications. Sensors 17 (1), 28. http://dx.doi.org/10.3390/s17010028.

Flach, P.A., Kull, M., 2015. Precision-recall-gain curves: PR analysis done right. In: Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1. NIPS '15, MIT Press, Cambridge, MA, USA, pp. 838–846, URL: https://papers.nips.cc/paper/2015/file/33e8075e9970de0cfea955afd4644bb2-Paper.pdf.

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315 (5814), 972–976. http://dx.doi.org/10.1126/science.1136800.

Gras, B., Giuffrida, C., Kurth, M., Bos, H., Razavi, K., 2020. ABSynthe: Automatic blackbox side-channel synthesis on commodity microarchitectures. In: Proceedings 2020 Network and Distributed System Security Symposium, Vol. 1. NDSS 2020, Internet Society, Reston, VA, pp. 633–650. http://dx.doi.org/10.14722/ndss.2020.23018.

Guédou, R., 2017. Using MIASM to fuzz binaries with AFL. BeeRump Workshop (community event), Bordeaux, URL: https://guedou.github.io/talks/2017_BeeRump/slides.pdf.

H. P, V., Poornima, B., Sagar, B., 2018. Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. pp. 511–518. http://dx.doi.org/10.1007/978-981-10-7563-6_53.

Han, Y., Etigowni, S., Liu, H., Zonouz, S., Petropulu, A., 2017. Watch me, but don't touch me! contactless control flow monitoring via electromagnetic emanations. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS '17, Association for Computing Machinery, New York, NY, USA, pp. 1095–1108. http://dx.doi.org/10.1145/3133956.3134081.

Harbecke, D., Chen, Y., Hennig, L., Alt, C., 2022. Why only micro-F1? Class weighting of measures for relation classification. In: Proceedings of NLP Power! the First Workshop on Efficient Benchmarking in NLP. Association for Computational Linguistics, Dublin, Ireland, pp. 32–41. http://dx.doi.org/10.18653/v1/2022.nlppower-1.4.

Hospodar, G., Gierlichs, B., Mulder, E., Verbauwhede, I., Vandewalle, J., 2011. Machine learning in side-channel analysis: A first study. J. Cryptogr. Eng. 1, 293–302. http://dx.doi.org/10.1007/s13389-011-0023-x.

Huang, X., Craig, P., Lin, H., Yan, Z., 2016. SecIoT: a security framework for the Internet of Things. Secur. Commun. Networks 9 (16), 3083–3094. http://dx.doi.org/10.1002/sec.1259.

Hussain, A., Ali, T., Althobiani, F., Draz, U., Irfan, M., Yasin, S., Shafiq, S., Safdar, Z., Glowacz, A., Nowakowski, G., Khan, M., Alqhtani, S.M., 2021. Security framework for IoT based real-time health applications. Electronics 10, 1–15. http://dx.doi.org/10.3390/electronics10060719.

IEEE, 2019. IEEE standard for floating-point arithmetic. http://dx.doi.org/10.1109/IEEESTD.2019.8766229, IEEE Std 754-2019 (Revision IEEE 754-2008).

International Electrotechnical Commission, 2018. IEC 62443-4-1:2018 - Security for Industrial Automation and Control Systems - Part 4-1: Secure Product Development Lifecycle Requirements. Technical Report, International Electrotechnical Commission, URL: https://webstore.iec.ch/publication/33615.

International Electrotechnical Commission, 2019. IEC 62443-4-2:2019 - Security for Industrial Automation and Control Systems - Part 4-2: Technical Security Requirements for IACS Components. Technical Report, International Electrotechnical Commission, URL: https://webstore.iec.ch/publication/34421.

IoT Analytics, 2023. State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally. URL: https://iot-analytics.com/number-connected-iot-devices/.

Islam, M.M., Lautenbach, A., Sandberg, C., Olovsson, T., 2016. A risk assessment framework for automotive embedded systems. In: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security. Association for Computing Machinery, New York, NY, USA, pp. 3–14. http://dx.doi.org/10.1145/2899015.2899018.

Jiménez, M., Palomera, R., Couvertier, I., 2013. Introduction to Embedded Systems: Using Microcontrollers and the MSP430. Springer, http://dx.doi.org/10.1007/978-1-4614-3143-5.

Johnson, W.A., Ghafoor, S., Prowell, S., 2021. A taxonomy and review of remote attestation schemes in embedded systems. IEEE Access 9, 142390–142410. http://dx.doi.org/10.1109/ACCESS.2021.3119220.

Khan, H.A., Sehatbakhsh, N., Nguyen, L., Prvulovic, M., Zajic, A., 2019. Malware detection in embedded systems using neural network model for electromagnetic side-channel signals. J. Hardw. Syst. Secur. http://dx.doi.org/10.1007/s41635-019-00074-w.

Kocher, P.C., 1996. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference. In: Lecture Notes in Computer Science, vol. 1109, Springer, pp. 104–113. http://dx.doi.org/10.1007/3-540-68697-5_9.

Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., Yarom, Y., 2019. Spectre attacks: Exploiting speculative execution. In: 2019 IEEE Symposium on Security and Privacy. SP, pp. 1–19. http://dx.doi.org/10.1109/SP.2019.00002.

Kocher, P.C., Jaffe, J., Jun, B., 1999. Differential power analysis. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. In: Lecture Notes in Computer Science, vol. 1666, Springer, pp. 388–397. http://dx.doi.org/10.1007/3-540-48405-1_25.

Lindon, J.C., Tranter, G.E., Koppenaal, D., 2016. Encyclopedia of Spectroscopy and Spectrometry. Academic Press.

Longo, J., Mulder, E., Page, D., All, M., 2015. SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip, vol. 9293, pp. 620–640. http://dx.doi.org/10.1007/978-3-662-48324-4_31.

Macnae, J., Lamontagne, Y., West, G., 1984. Noise processing techniques for time-domain EM systems. GEOPHYSICS 49, 934–948. http://dx.doi.org/10.1190/1.1441739.

Main, C., 2010. Real-time and general-purpose operating systems unite via virtualization. URL: https://embeddedcomputing.com/technology/software-and-os/real-time-and-general-purpose-operating-systems-unite-via-virtualization.

Mangard, S., Oswald, E., Popp, T., 2010. Power Analysis Attacks: Revealing the Secrets of Smart Cards, first ed. Springer Publishing Company, Incorporated.

Market.us, 2024. Internet of Things (IoT) market. URL: https://market.us/report/internet-of-things-iot-market.

McInnes, L., Healy, J., Astels, S., 2017. hdbscan: Hierarchical density based clustering. J. Open Source Softw. 2, http://dx.doi.org/10.21105/joss.00205.

Miller, B.P., Fredriksen, L., So, B., 1990. An empirical study of the reliability of UNIX utilities. Commun. ACM 33 (12), 32–44. http://dx.doi.org/10.1145/96267.96279.

MITRE Corporation, 2025. Common weakness enumeration (CWE). URL: https://cwe.mitre.org.

Muench, M., Stijohann, J., Kargl, F., Francillon, A., Balzarotti, D., 2018. What you corrupt is not what you crash: Challenges in fuzzing embedded devices. http://dx.doi.org/10.14722/ndss.2018.23176.

Nazari, A., Sehatbakhsh, N., Alam, M., Zajic, A., Prvulovic, M., 2017. EDDIE: EM-based detection of deviations in program execution. In: Proceedings of the 44th Annual International Symposium on Computer Architecture. ISCA '17, Association for Computing Machinery, New York, NY, USA, pp. 333–346. http://dx.doi.org/10.1145/3079856.3080223.

NETGEAR, Bitdefender, 2024. The 2024 IoT Security Landscape Report. Technical Report, NETGEAR, Inc., URL: https://blogapp.bitdefender.com/hotforsecurity/content/files/2024/06/2024-IoT-Security-Landscape-Report_consumer.pdf.

Nong, Y., Cai, H., Ye, P., Li, L., Chen, F., 2021. Evaluating and comparing memory error vulnerability detectors. Inf. Softw. Technol. 137, 106614. http://dx.doi.org/10.1016/j.infsof.2021.106614.

Open Web Application Security Project, 2018. OWASP Internet of Things (IoT) top 10 2018. URL: https://owasp.org/www-project-internet-of-things/.

Ott, H.W., 2009. Electromagnetic Compatibility Engineering. John Wiley & Sons.

Pacheco, J., Hariri, S., 2016. IoT security framework for smart cyber infrastructures. In: 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems. FAS*W, pp. 242–247. http://dx.doi.org/10.1109/FAS-W.2016.58.

Parameswaran, S., Wolf, T., 2008. Embedded systems security–an overview. Des. Autom. Embedded Syst. 12 (3), 173–183. http://dx.doi.org/10.1007/s10617-008-9027-x.

Pham, D.-P., Marion, D., Mastio, M., Heuser, A., 2021. Obfuscation revealed: Leveraging electromagnetic signals for obfuscated malware classification. In: Proceedings of the 37th Annual Computer Security Applications Conference. Association for Computing Machinery, New York, NY, USA, pp. 706–719. http://dx.doi.org/10.1145/3485832.3485894.

Quisquater, J.-J., Samyde, D., 2001. ElectroMagnetic analysis (EMA): Measures and counter-measures for smart cards. 2140, pp. 200–210. http://dx.doi.org/10.1007/3-540-45418-7_17,

Raje, K., 2025. STM32 Series Single Chip Microcomputer Market Report 2025. URL: https://www.cognitivemarketresearch.com/stm32-series-single-chip-microcomputer-market-report.

Sayakkara, A., Le-Khac, N.-A., Scanlon, M., 2019. Leveraging electromagnetic side-channel analysis for the investigation of IoT devices. Digit. Investig. 29, S94–S103. http://dx.doi.org/10.1016/j.diin.2019.04.012.

Sehatbakhsh, N., Nazari, A., Alam, M., Werner, F.T., Zhu, Y., Zajić, A.G., Prvulović, M., 2020. REMOTE: Robust external malware detection framework by using electromagnetic signals. IEEE Trans. Comput. 69, 312–326, URL: https://api.semanticscholar.org/CorpusID:208116287.

Sorensen, H., Jones, D., Heideman, M., Burrus, C., 1987. Real-valued fast Fourier transform algorithms. IEEE Trans. Acoust. Speech Signal Process. 35 (6), 849–863. http://dx.doi.org/10.1109/TASSP.1987.1165220.

Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., Markakis, E.K., 2020. A survey on the Internet of Things (IoT) forensics: Challenges, approaches, and open issues. IEEE Commun. Surv. & Tutorials 22 (2), 1191–1221. http://dx.doi.org/10.1109/COMST.2019.2962586.

Tamura, M., Tsujita, S., 2007. A study on the number of principal components and sensitivity of fault detection using PCA. Comput. Chem. Eng. 31, 1035–1046. http://dx.doi.org/10.1016/j.compchemeng.2006.09.004.

Thakor, V.A., Razzaque, M.A., Khandaker, M.R.A., 2021. Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. IEEE Access 9, 28177–28193. http://dx.doi.org/10.1109/ACCESS.2021.3052867.

Tickelton, M., 2020. Using AFL to fuzz ARM binaries on a Raspberry Pi and with AFL QEMU Mode. URL: https://tickelton.gitlab.io/articles/afl-arm-rpi/.

Trippel, T., Weisse, O., Xu, W., Honeyman, P., Fu, K., 2017. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In: 2017 IEEE European Symposium on Security and Privacy. EuroS&P, pp. 3–18. http://dx.doi.org/10.1109/EuroSP.2017.42.

Valgrind Developers, 2025. Valgrind User Manual. Online, URL: https://valgrind.org/docs/manual/valgrind_manual.pdf.

Wang, C.-D., Lai, J.-H., Suen, C.Y., Zhu, J.-Y., 2013. Multi-exemplar affinity propagation. IEEE Trans. Pattern Anal. Mach. Intell. 35 (9), 2223–2237. http://dx.doi.org/10.1109/TPAMI.2013.28.

Wang, R., Wang, H., Dubrova, E., 2020. Far field EM side-channel attack on AES using deep learning. In: Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security. ASHES '20, Association for Computing Machinery, New York, NY, USA, pp. 35–44. http://dx.doi.org/10.1145/3411504.3421214.

Wang, X., Zhou, Q., Harer, J., Brown, G., Qiu, S., Dou, Z., Aguayo Gonzalez, C., Hinton, A., Wang, J., Chin, S., 2018. Deep learning-based classification and anomaly detection of side-channel signals. p. 6. http://dx.doi.org/10.1117/12.2311329.

Watanabe, S., 2012. A widely applicable Bayesian information criterion. http://dx.doi.org/10.48550/arXiv.1208.6338.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2, 37–52. http://dx.doi.org/10.1016/0169-7439(87)80084-9.

World Economic Forum, 2024. How the Internet of Things (IoT) became a dark web target – and what to do about it. URL: https://www.weforum.org/agenda/2024/05/internet-of-things-dark-web-strategy-supply-value-chain/.

Wu, J., Gao, Y., Lu, Z., Yang, X., Xin, B., 2024. Electromagnetic shielding performance of copper wire/carbon fiber fabric reinforced composite materials. J. Reinf. Plast. Compos. http://dx.doi.org/10.1177/07316844241230732.

You, M., Kim, Y., Kim, J., Seo, M., Son, S., Shin, S., Lee, S., 2022. FuzzDocs: An Automated security evaluation framework for IoT. IEEE Access 10, 102406–102420. http://dx.doi.org/10.1109/ACCESS.2022.3208146.