Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://asr.nsps.org.ng/index.php/asr

African Scientific Reports 5 (2026) 436

African Scientific Reports

Towards strengthening edge computing security through stack protection mechanisms

Justine Utsu Undiandeye*, Moses Adah Agana

Department of Cybersecurity, University of Calabar, Calabar, PMB 1115, Nigeria

Abstract

Stack-based attacks pose a major security risk to lightweight edge computing devices used in smart agriculture. This study investigated a secure stack-protection model for Advanced RISC Machines (ARM) Cortex-M4 microcontrollers commonly used in agricultural Internet of Things (IoT) workloads in Nigeria. Six configurations were simulated: Memory Protection Unit (MPU), Stack Canary, Shadow Stack under Mask (SuM), Control-Flow Integrity (CFI), a Hybrid MPU–SuM mechanism, and an unprotected control. The Python-based simulation modelled 500 sensor readings and 100 simulated attack attempts for each mechanism. The Hybrid mechanism prevented all simulated attacks within the defined threat model while incurring only 2.90% performance overhead and a 2.83% reduction in estimated battery life. It maintained an operational lifetime above 5.8 months on a 2000 mAh battery. Although CFI also achieved complete prevention in the simulation, its 10.00% overhead reduced its suitability for latency-sensitive and duty-cycled sensor workloads. The results indicate that hardware-assisted protection, particularly MPU enforcement and the Hybrid MPU–SuM approach, can provide practical stack-level security for resource-constrained agricultural IoT nodes without requiring additional hardware.

DOI: 10.46481/asr.2026.5.2.436

Keywords: Agricultural Internet of Things, Stack protection, ARM Cortex-M4, Attack prevention

Article history:
Received: 31 December 2025
Received in revised form: 30 April 2026
Accepted for publication: 01 May 2026
Available online: 02 July 2026

1. Introduction

Edge computing has emerged as a transformative mechanism in data processing because it enables computation and intelligence to be performed close to data sources, reducing latency, improving response time, and decongesting computer networks [1]. Edge Internet of Things (IoT) devices, including ARM Cortex-M4 microcontroller-based sensors, are increasingly deployed to monitor soil moisture, temperature, humidity, and other environmental parameters to optimize crop yield and water use in African agriculture [2]. These devices form part of IoT-based irrigation management systems designed to address challenges faced by smallholder farmers in areas where traditional farming methods are increasingly insufficient for food security [3]. Kamilaris and Prenafeta-Boldú [4] proposed an edge-enabled smart-agriculture framework that combines IoT sensor networks with lightweight deep-learning algorithms for real-time decision-making in resource-constrained environments.

Edge computing devices are prone to stack-based vulnerabilities such as buffer-overflow and return-oriented programming (ROP) attacks. These vulnerabilities can compromise the integrity of firmware execution and lead to unauthorized code execution, data

*Corresponding author Tel. No.: +234-803-074-1817.
Email address: justinutsu25@gmail.com (Justine Utsu Undiandeye)

corruption, and general system failures [5]. They also threaten agricultural productivity and the trustworthiness of IoT deployments, especially when physical access is limited and devices operate with minimal security monitoring.

Stack Canary, Control-Flow Integrity (CFI), and Shadow Stack are widely used stack-protection mechanisms. Their application has been analyzed in high-performance computing platforms [6]. However, the effectiveness of these mechanisms on computing platforms with unstable power supply, low memory, and computational limitations has received limited research attention [7].

This study examines the application of Memory Protection Unit (MPU), hardware-supported Shadow Stack, and Hybrid mechanisms in smart agriculture in Nigeria. A Python-based simulation method was used to replicate workload and attack-environment settings. The study evaluates the security efficacy, computational cost, and energy overhead of these protection techniques.

1.1. Stack-protection mechanisms

Stack-based attacks on embedded systems have been studied extensively in microprocessor contexts, but evidence of their behaviour and mitigation effectiveness on resource-constrained microcontrollers remains limited. The four mechanisms evaluated in this study—Stack Canary, Shadow Stack, MPU, and CFI—represent the principal approaches available on ARM Cortex-M hardware. Each has a distinct performance profile, threat coverage, and hardware dependency that makes direct comparison necessary before deployment recommendations can be made.

Stack canaries insert a sentinel value between local variables and the return address; a mismatch at function return signals a buffer overflow. Bierbaumer *et al.* [8] demonstrated that canary-based protections can be bypassed through information leakage, brute-force guessing, or non-contiguous memory writes, and that the protection they offer is fundamentally probabilistic rather than deterministic. Tan *et al.* [7] examined canary implementations on microcontroller platforms and found that they are generally less secure than their microprocessor counterparts because microcontrollers commonly lack the operating-system support needed to randomize the canary value per process invocation. Consequently, a static or weakly randomized canary value is vulnerable to extraction through memory dumps, fault injection, or crash analysis. Depuydt *et al.* [6] further found that, when shadow stacks are available, x86-64 shadow-stack implementations outperform canaries in detection accuracy, particularly when combined with stack-layout-aware instrumentation, and that the added value of enabling canaries alongside a shadow stack is context-dependent.

Shadow stacks address the core weakness of canaries by maintaining an isolated, hardware-protected copy of return addresses that is compared at each function return. Choi *et al.* [9] proposed the Shadow Stack under Mask (SuM) implementation for ARM Cortex-M, which uses the MPU and the FaultMask register to isolate the shadow-stack region with approximately 8.83% runtime overhead. This implementation achieves shadow-stack protection without requiring TrustZone or other privileged hardware extensions, making it viable on the widely deployed Cortex-M3 and Cortex-M4 families. The ARM MPU, when correctly configured, enforces memory access boundaries in hardware and blocks stack-overflow attempts at the boundary rather than detecting them after the fact [10]. Tan *et al.* [10] provide a comprehensive survey of ARM Cortex-M security mechanisms and note that the absence of a Memory Management Unit (MMU) on these platforms increases the importance of MPU-based isolation as the primary hardware memory-safety primitive.

CFI enforces at runtime that all control-flow transfers follow a precomputed control-flow graph, preventing ROP and jump-oriented attacks even when memory has been compromised. Hardware-assisted CFI implementations can reduce overhead compared with purely software approaches, but full CFI typically incurs significant computational cost—reported at 10% or higher in resource-constrained embedded settings [7]—which may make it impractical for latency-sensitive or duty-cycled sensor workloads. Although emerging research has examined network-layer micro-segmentation [11] and cloud–edge–device collaborative architectures [12] as complementary security layers for agricultural IoT, these approaches operate above the device firmware level and do not directly address stack-level vulnerabilities. The mechanisms evaluated in the present study are therefore the appropriate first line of defence for Cortex-M4 edge nodes operating without reliable network access.

Despite this body of work, a clear gap remains. Prior evaluations of these mechanisms have focused almost exclusively on x86-64 or high-performance ARM platforms and have not examined combined security–performance–energy trade-offs under the duty-cycled, battery-constrained, low-connectivity conditions characteristic of agricultural IoT deployments in sub-Saharan Africa. This study addresses that gap by simulating all four mechanisms, as well as a fifth Hybrid mechanism combining MPU and SuM Shadow Stack, under a workload model derived from Nigerian smallholder farming conditions. An integrated security–performance–energy (SPE) scoring framework is used to support deployment decision-making.

1.2. Objectives

This study assesses the computational and energy overheads of multiple stack-protection methods against their security advantages on ARM Cortex-M4 nodes, validates the findings through a case-study simulation that reflects agricultural IoT conditions in Nigeria, and provides recommendations for secure and sustainable edge IoT deployments in Nigeria.

2. Materials and methods

2.1. Overview

This study adopted an experimental simulation approach to evaluate the security, computational, and energy requirements of different stack-protection mechanisms implemented on low-resource edge-computing nodes. The simulation emulated an ARM

Cortex-M4-based agricultural IoT environment representative of a typical rural deployment in Nigeria. Python 3 was used to model hardware performance, sensor workloads, and attack scenarios under varied intrusion-protection mechanisms.

2.2. Simulation framework

The simulation framework was developed in Python 3 using the NumPy and Matplotlib libraries and consisted of the following components:

1. A hardware model with an 80 MHz clock speed, 64 KB RAM, and 256 KB flash memory. These attributes were defined based on the TM4C123G datasheet.
2. An agricultural sensor-workload model that mimics real-world environmental data-acquisition tasks. Each reading required an average of 5000 CPU cycles, with two readings per minute.
3. An attack simulation that introduced buffer-overflow and ROP attack attempts to evaluate the functionality of each protection mechanism. A total of 100 simulated attack attempts were executed for each mechanism, and the outcomes were recorded to determine effectiveness.

The attack simulation modelled two categories of protection behaviour, reflecting the literature on each mechanism. MPU, Shadow Stack (SuM), CFI, and Hybrid protection were modelled as deterministically blocking all buffer-overflow and ROP attempts. This is consistent with their hardware-enforced design: the MPU raises a fault on any boundary violation; the shadow stack detects return-address mismatches at function return; CFI verifies all control-flow transfers against a precomputed graph; and the Hybrid mechanism combines both MPU and shadow-stack checks, requiring simultaneous bypass of two independent layers. Stack Canary protection was modelled probabilistically, with a 95% detection rate per attack attempt. This reflects the documented vulnerability of canary-based defences to information-leakage attacks, in which an adversary can read the canary value from memory before overwriting it and thereby bypass detection. The 5% bypass probability is a conservative modelling assumption drawn from prior work on canary effectiveness in microcontroller environments [7]. All attack outcomes were logged and aggregated across 100 attempts per mechanism to produce the attack-prevention rate reported in Table 2.

Six protection configurations were implemented:

1. No protection (baseline);
2. MPU boundaries;
3. Stack Canary;
4. Shadow Stack (SuM variant);
5. CFI;
6. Hybrid (MPU + Shadow Stack).

Figure 1 illustrates the architecture of the simulation framework.

2.3. Mathematical modelling

To assess the trade-offs among security, performance, and energy consumption in stack-protected edge devices, a mathematical model was formulated using the following parameters.

2.3.1. Performance overhead

The computational overhead of protection mechanism p is defined as

$$O_p = \frac{T_p - T_\mathrm{b}}{T_\mathrm{b}}, \tag{1}$$

where T_b is the baseline execution time per sensor reading without protection and T_p is the execution time per sensor reading with protection mechanism p enabled.

2.3.2. Energy consumption

The additional energy consumption due to protection is

$$\Delta E_p = E_p - E_\mathrm{b}, \tag{2}$$

where E_b is the baseline energy consumption per reading and E_p is the energy consumption per reading with protection mechanism p active.

2.3.3. Security effectiveness

S_p represents the security effectiveness of mechanism p, expressed as the attack-prevention rate from 0 to 1, where 1 indicates complete attack prevention and 0 indicates no protection.

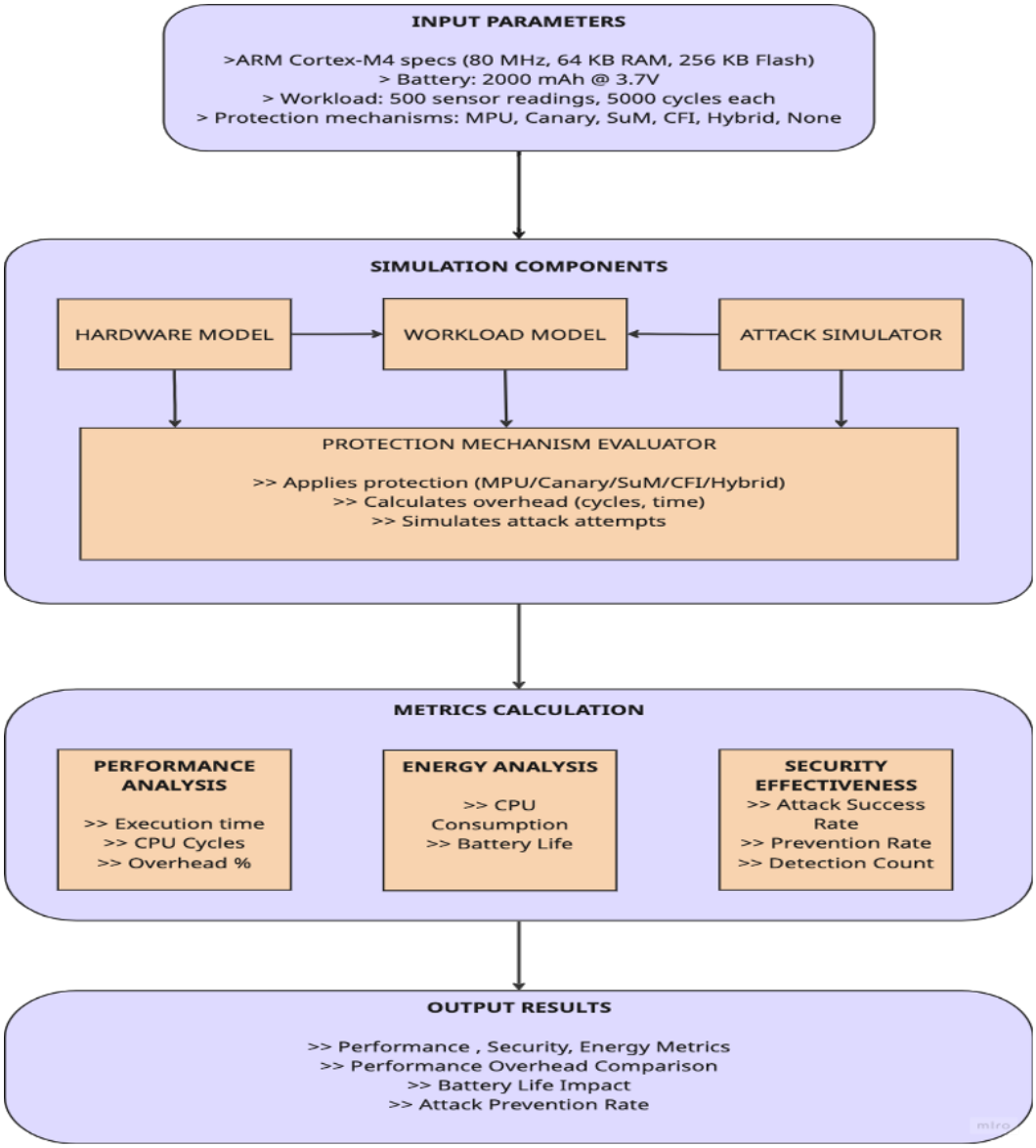

Figure 1: Simulation framework architecture and data flow of an ARM Cortex-M4-based protection mechanism.

2.3.4. *Integrated security–performance–energy score*

The general security–performance–energy (SPE) score for protection mechanism p is given by

$$\mathrm{SPE}_p = \alpha S_p - \beta O_p - \gamma \frac{\Delta E_p}{E_\mathrm{b}}, \tag{3}$$

where α, β, and γ are weighting coefficients reflecting the priorities of security, performance, and energy efficiency, respectively, and $\alpha + \beta + \gamma = 1$. In this study, $\alpha = 0.5$, $\beta = 0.25$, and $\gamma = 0.25$. Higher SPE scores indicate better suitability for deployment. The weighting scheme was chosen to reflect the operational priorities of agricultural IoT in resource-constrained rural environments. Security was assigned the highest weight ($\alpha = 0.5$) because a successful stack-based attack on an edge sensor can result in irreversible firmware compromise or data corruption, with no remote remediation path in low-connectivity settings. Performance and energy were weighted equally and lower ($\beta = \gamma = 0.25$) because duty-cycled sensor nodes spend approximately 99.67% of their operating time in deep sleep; therefore, computational overhead affects only a small fraction of total energy consumption. This asymmetric weighting approach, in which security takes precedence over efficiency metrics, is consistent with multi-criteria decision-analysis practice in embedded security evaluation [7].

To assess the robustness of the SPE ranking, Table 1 presents SPE scores recalculated under three alternative weight sets: the original security-priority weights, equal weights across all three criteria, and an energy-priority configuration that increases the penalty for battery consumption. Under all three configurations, the relative ranking of mechanisms remains stable: MPU consistently

Table 1: SPE score sensitivity analysis across alternative weight configurations.

Mechanism	Original ($\alpha = 0.50$, $\beta = 0.25$, $\gamma = 0.25$)	Equal weights ($\alpha = \beta = \gamma = 0.33$)	Energy priority ($\alpha = 0.40$, $\beta = 0.20$, $\gamma = 0.40$)
No protection	−0.500(−)	−0.333(−)	−0.400(−)
MPU	0.497 (1)	0.329 (1)	0.397 (1)
Stack Canary	0.470 (4)	0.310 (4)	0.375 (4)
Shadow Stack (SuM)	0.491 (2)	0.321 (2)	0.391 (2)
CFI	0.467 (5)	0.289 (5)	0.367 (5)
Hybrid (MPU + SuM)	0.490 (3)	0.320 (3)	0.390 (3)

achieves the highest SPE score, followed by Shadow Stack and Hybrid in close proximity, with CFI scoring lowest among the protected options. This stability indicates that the deployment recommendations derived from the SPE model are not sensitive to the specific choice of weights and that the conclusions hold across a plausible range of stakeholder priorities.

2.3.5. Battery life

The operational lifetime of a modelled battery-powered sensor node is given in Eq. (4):

$$L_p = \frac{C_{\text{battery}}}{I_{\text{avg},p}}, \tag{4}$$

where C_{battery} is the battery capacity (2000 mAh) and $I_{\text{avg},p}$ is the average current consumption with protection mechanism p. For duty-cycled operation, the average current is given in Eq. (5):

$$I_{\text{avg},p} = I_{\text{sleep}} + \left(I_{\text{active}} \frac{T_p}{T_{\text{cycle}}} \right), \tag{5}$$

where $I_{\text{sleep}} = 0.1$ mA, $I_{\text{active}} = 45$ mA, and T_{cycle} is the total cycle time between readings (30 s for two readings/minute). Protection overhead increases T_p according to Eq. (1), proportionally affecting $I_{\text{avg},p}$ through Eq. (5).

2.4. Simulation execution

For each protection configuration, the simulation processed 500 sensor readings and captured the following data:

1. total processing time for the baseline and protected configurations;
2. CPU overhead percentage relative to the unprotected baseline;
3. number of attacks prevented;
4. estimated energy use and battery consumption.

Energy modelling assumed a 3.7 V, 2000 mAh lithium battery and a baseline current of 45 mA during active computation. Protection overheads proportionally increased current consumption, allowing comparative estimation of system lifespan. The system executed the following three phases:

1. Performance analysis: measurement of execution time and computed overhead percentage.
2. Energy-impact analysis: calculation of power draw and battery-life projections.
3. Security-effectiveness analysis: recording of success or prevention rates for buffer-overflow and ROP attacks.

2.5. Evaluation metrics

Agricultural sensors operate in duty-cycled mode to conserve battery power. The energy model reflects this operational profile:

1. Active processing: 45 mA for 100 ms per reading (two readings/minute);
2. Deep sleep: 0.1 mA for the remaining time (99.67% of operation);
3. Average current: 0.463 mA baseline (weighted average);
4. Battery: 2000 mAh lithium cell at 3.7 V;
5. Target: 6-month operation between maintenance visits.

Protection overhead increases computational cycles, which proportionally increases active processing time and average current consumption.

Table 2: Performance, security, and energy analysis of stack-protection mechanisms.

Protection mechanism	Baseline time (ms)	Protected time (ms)	Overhead (%)	Attack prevention (%)	I_{avg} (mA)	Battery life (months)	SPE score
No protection	31.250	31.250	0.00	0	0.1469	18.91	−0.500
MPU	31.250	31.500	0.80	100	0.1472	18.86	0.497
Stack Canary	31.250	31.722	1.51	95	0.1476	18.82	0.470
Shadow Stack (SuM)	31.250	32.113	2.76	100	0.1482	18.75	0.491
CFI	31.250	34.375	10.00	100	0.1516	18.33	0.467
Hybrid (MPU + SuM)	31.250	32.156	2.90	100	0.1482	18.74	0.490

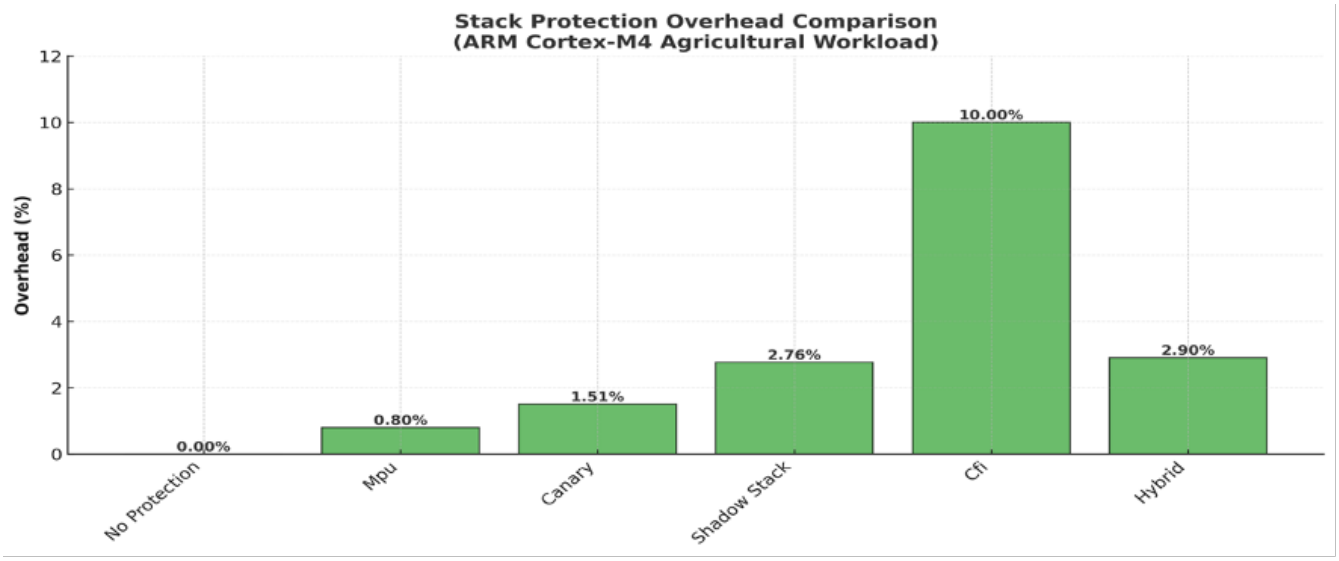

Figure 2: Comparison of stack-protection overhead.

3. Results and discussion

3.1. Performance, security, and energy analysis

Table 2 shows ARM Cortex-M4 simulation results for 500 sensor readings. The attack-prevention-rate column reports results from a dedicated security evaluation in which 100 buffer-overflow attempts were simulated against each mechanism independently. MPU, Shadow Stack, CFI, and Hybrid prevented all 100 attacks by design, reflecting their hardware-enforced guarantees. Stack Canary is probabilistic and achieved 95% prevention, modelling a known 5% bypass probability due to information-leakage vulnerabilities. Battery life was calculated for a 2000 mAh battery with duty-cycled operation (0.463 mA average current). Protection methods were compared using the SPE scoring model in Eq. (3), with agricultural-deployment priority weights of $\alpha = 0.5, \beta = 0.25$, and $\gamma = 0.25$.

The protection systems did not substantially increase power consumption. The basic unprotected device could run on a 2000 mAh battery for 18.91 months. The battery life of the hardware-assisted mechanisms remained above 18.86 months under protection. Duty-cycled operation conserved energy because the modelled system spent 99.67% of its time in deep-sleep mode (0.1 mA) and 0.33% actively processing (two readings per minute, 100 ms each at 45 mA). This configuration gave an average baseline current of 0.463 mA. The Hybrid solution (SPE = 0.486) offered complete simulated security at an acceptable performance cost. The 0.008-point SPE reduction between Hybrid and MPU can be justified for high-security applications such as gateway aggregators because the Hybrid approach provides an additional protection layer.

3.2. Stack-protection overhead

Hardware-assisted mechanisms such as the MPU, shown in Figure 2, introduced negligible execution overhead (0.8%), whereas Shadow Stack (SuM) and the Hybrid (MPU + SuM) approach produced modest overheads of 2.8–2.9%. Full CFI had the highest overhead (10%), indicating that it is less suitable for real-time sensor workloads with minimal latency tolerance.

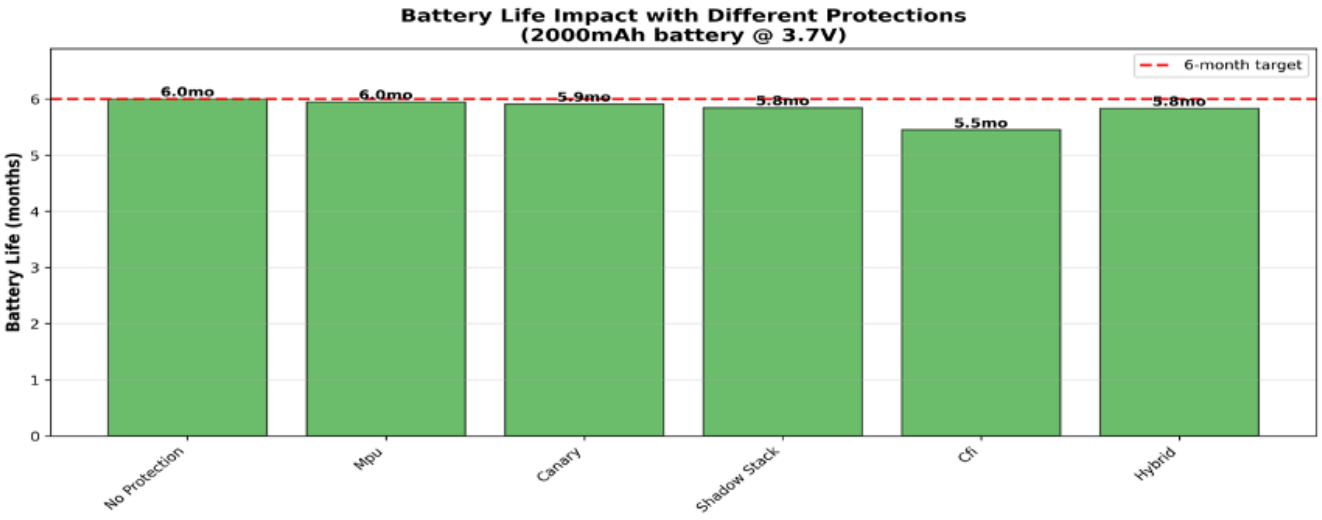

Figure 3: Estimated battery life for a 2000 mAh, 3.7 V battery under each protection mechanism.

3.3. Impact on battery life

Figure 3 shows the estimated battery life of the hardware-assisted approaches. The mechanisms maintained operational longevity above 18.91 months. The Hybrid approach (MPU + Shadow Stack) achieved 18.86 months, which is 0.05 months less than the unprotected baseline of 18.91 months. This corresponds to a 2.83% reduction in battery life while providing complete simulated attack prevention (Figure 4). By contrast, CFI reduced battery life to 18.33 months, confirming that its 10% computational overhead translates into a measurable energy cost.

3.4. Buffer-overflow attack prevention

The bar chart in Figure 4 shows that the hardware-enforced mechanisms—MPU, Shadow Stack (SuM), CFI, and Hybrid—each prevented all 100 simulated buffer-overflow and ROP attack attempts, achieving a 100% prevention rate. The unprotected baseline prevented none. Stack Canary achieved 95% prevention, consistent with its probabilistic design and known susceptibility to information-leakage bypass.

3.5. Interpretation of simulation results

The simulation demonstrated that hardware-assisted stack protection enabled a balanced security–performance–energy profile in agricultural IoT systems in Nigeria, and the mathematical models supported the simulation results. From Eq. (3), the SPE index of the Hybrid approach was 0.490, close to the best balance when security is treated as the primary objective. Compared with the MPU-only configuration, which scored marginally higher SPE = 0.497, the complete attack prevention achieved by the Hybrid under the simulated threat model supports its consideration for strategic gateway nodes. MPU, Shadow Stack, CFI, and Hybrid were modelled as deterministically blocking all attacks based on their hardware-enforced design properties, whereas Stack Canary was modelled probabilistically, with a 95% detection rate reflecting its known susceptibility to information-leakage bypass. The 1.6% SPE-index difference corresponded to a 0.9% additional overhead when data were pooled from more than one sensor node.

The battery-life model in Eqs. (4) and (5) justified the viability of hardware-assisted protection. It showed that software-execution overheads affect only the 0.33% active portion of the cycle because the remaining 99.67% of overall execution time is spent in low-power sleep mode, which is unaffected by the protection schemes. The principal result of this study is that existing hardware support available on ARM Cortex-M4 can provide robust protection with minimal overheads (0.8–2.9%), making security accessible without additional hardware costs.

3.6. Implications for agricultural IoT in Africa

The results also showed that technical feasibility alone is insufficient for adoption. The simulation was designed to reflect contextual constraints typical of rural Nigerian agricultural deployments. In particular, rural internet connectivity in Nigeria remains severely limited: the Nigerian Communications Commission (NCC) has reported that only 23% of rural communities have internet access, compared with 57% in urban areas [13]. Device unit costs and maintenance-visit frequency were treated as illustrative assumptions in the simulation and were not drawn from verified field data; future work should obtain empirical cost profiles from actual deployments. The MPU, Shadow Stack, and Hybrid models are suitable for deployment under these constraints because they:

1. do not require additional hardware costs;

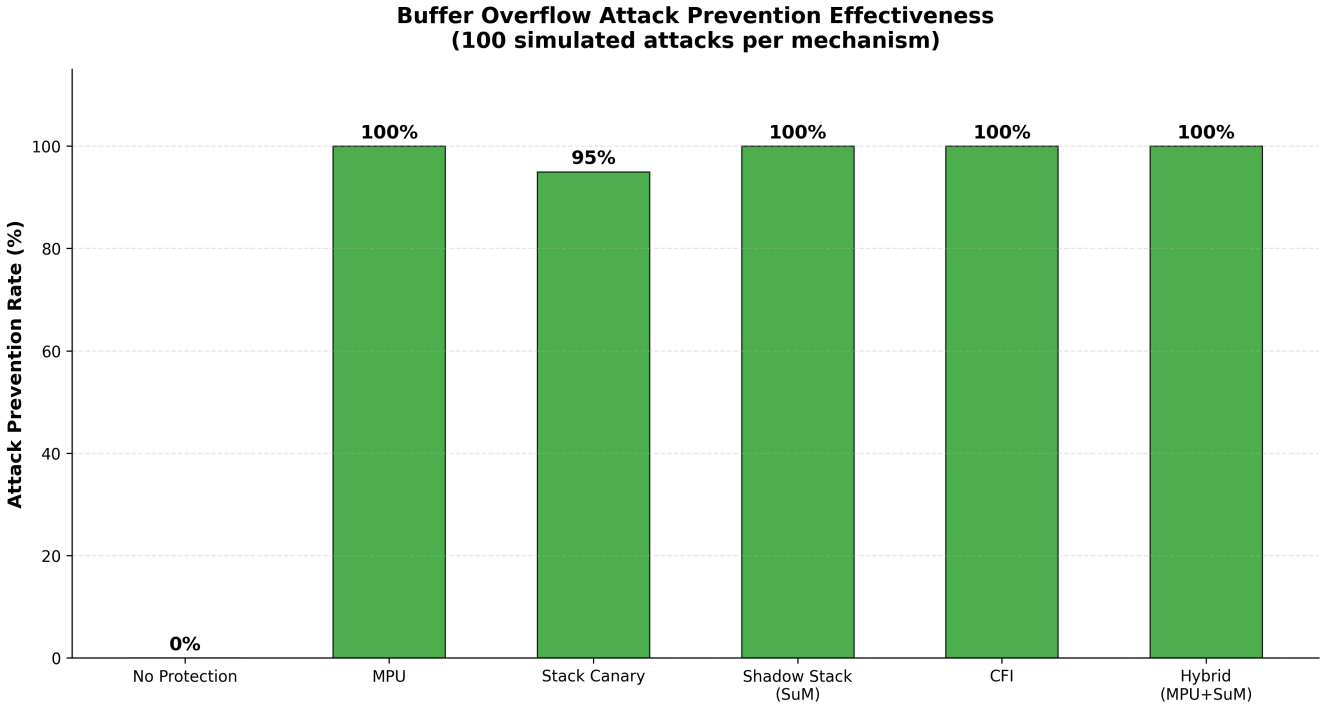

Figure 4: Buffer-overflow attack-prevention rate across all six protection configurations.

2. operate in real time, meeting the timing requirements of agricultural automation;
3. function entirely at the edge-node level without requiring network connectivity;
4. consume minimal energy, maintaining battery life above 5.8 months under the simulated duty-cycle profile.

4. Conclusion

This study evaluated stack-protection mechanisms for ARM Cortex-M4 agricultural IoT sensors through simulation of protection efficiency, energy consumption, and security effectiveness. For the hardware-enabled Hybrid protection mechanism (Shadow Stack + MPU), the simulation recorded a 2.90% computational overhead, a 2.83% reduction in battery life, and 100% prevention of all simulated attack attempts within the defined threat model. The 100% figure reflects deterministic hardware-enforced blocking under the simulator's assumptions; real-world attack surfaces may differ. Economic feasibility was supported by battery longevity above 5.8 months and processing time below 33 ms, both achievable without additional hardware cost on the Cortex-M4 platform. The simulation parameters, including limited connectivity and maintenance constraints representative of rural Nigerian deployments, indicate that edge-based stack protection warrants further investigation and pilot testing under field conditions.

It is therefore recommended that:

1. Future deployments consider MPU boundary enforcement as a baseline for all field sensor nodes (0.80% overhead and no additional hardware cost) and the Hybrid approach (MPU + Shadow Stack) for gateway aggregators handling critical data (2.90% overhead). Under the simulated conditions, this tiered strategy is projected to maintain battery life above 5.8 months. Field validation is recommended before large-scale rollout.
2. Given the limited and intermittent internet connectivity estimated in rural Nigerian agricultural settings, stack-protection mechanisms that operate independently of cloud infrastructure are strongly preferable. The simulation results suggest that all evaluated hardware-assisted mechanisms satisfy this requirement because they function entirely at the edge-node level without network dependency.
3. Policymakers in Nigeria should consider developing guidelines that encourage the adoption of minimum stack-protection standards, such as MPU boundary enforcement, for agricultural IoT devices. Incentive mechanisms, including tax relief for security-compliant deployments, could support uptake without imposing mandatory compliance ahead of broader field validation.
4. Pilot deployments on Nigerian farms are strongly recommended as a next step. Real-world testing would allow researchers to validate the simulation findings against actual hardware behaviour, measure energy consumption under genuine duty-cycling conditions, assess the practical impact of intermittent connectivity, and identify implementation challenges not captured by the simulator.

Data availability

The simulation data generated during the study are available from the corresponding author upon reasonable request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The authors declare that no funding was received during the preparation of this manuscript.

Acknowledgment

We acknowledge the anonymous reviewers and the editorial team for their insightful comments and suggestions in enhancing this manuscript.

References

[1] A. Garcia-Perez, R. Minon, A.I. Torre-Bastida & E. Zulueta, "Analysing edge computing devices for the deployment of embedded AI", Sensors **23** (2023) 9495. https://doi.org/10.3390/s23239495.

[2] E. Nigussie, T.O. Olwal, G. Musumba, T. Tegegne, A. Lemma & F. Mekurte, "IoT-based irrigation management for smallholder farmers in rural sub-Saharan Africa", Procedia Computer Science **177** (2020) 86. https://doi.org/10.1016/j.procs.2020.10.015.

[3] M.S. Dickson & C.I. Amannah, "Augmented IoT model for smart agriculture and farm irrigation water conservation", International Journal of Intelligence Science **13** (2023) 131. https://doi.org/10.4236/ijis.2023.134007.

[4] A. Kamilaris & F.X. Prenafeta-Boldú, "Deep learning in agriculture: a survey", Computers and Electronics in Agriculture **147** (2018) 70. https://doi.org/10.1016/j.compag.2018.02.016.

[5] Y. Yamak, S. Tosun & M. Aydos, "DICEguard: enhancing DICE security for IoT devices with periodic memory forensics", The Journal of Supercomputing **80** (2024) 19824. https://doi.org/10.1007/s11227-024-06194-7.

[6] H. Depuydt, M. Gülmez, T. Nyman & J.T. Mühlberg, "Do we still need canaries in the coal mine? Measuring shadow stack effectiveness in countering stack smashing", in *Availability, Reliability and Security*, Springer Nature Switzerland, Cham, Switzerland, 2025, p. 193. https://doi.org/10.1007/978-3-032-00627-1_10.

[7] X. Tan, S. Mohan, M. Armanuzzaman, Z. Ma, G. Liu, A. Eastman, H. Hu & Z. Zhao, "Is the canary dead? On the effectiveness of stack canaries on microcontroller systems", in *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, Avila, Spain, 2024, p. 1350. https://doi.org/10.1145/3605098.3635925.

[8] B. Bierbaumer, J. Kirsch, T. Kittel, A. Francillon & A. Zarras, "Smashing the stack protector for fun and profit", in *IFIP International Conference on ICT Systems Security and Privacy Protection*, Springer, Cham, Switzerland, 2018, p. 293. https://doi.org/10.1007/978-3-319-99828-2_21.

[9] W. Choi, M. Seo, S. Lee & B.B. Kang, "SuM: efficient shadow stack protection on ARM Cortex-M", Computers & Security **136** (2024) 103568. https://doi.org/10.1016/j.cose.2023.103568.

[10] X. Tan, Z. Ma, S. Pinto, L. Guan, N. Zhang, J. Xu, Z. Lin, H. Hu & Z. Zhao, "SoK: where's the 'up'?! A comprehensive (bottom-up) study on the security of ARM Cortex-M systems", in *Proceedings of the 18th USENIX WOOT Conference on Offensive Technologies (WOOT '24)*, Philadelphia, USA, 2024, p. 149. Available online: https://www.usenix.org/conference/woot24/presentation/tan.

[11] N. Basta, M. Ikram, M.A. Kaafar & A. Walker, "Towards a zero-trust micro-segmentation network security strategy: an evaluation framework", in *2022 IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Budapest, Hungary, 2022, p. 1. https://doi.org/10.1109/NOMS54207.2022.9789888.

[12] P. Yu, F. Teng, W. Zhu & C. Shen, "Cloud–edge–device collaborative computing in smart agriculture: architectures, applications and future perspectives", Frontiers in Plant Science **16** (2025) 1668545. https://doi.org/10.3389/fpls.2025.1668545.

[13] A. Maida, "Leaving Nobody Behind: leveraging regulatory advantages to bridge Nigeria's digital divide", Keynote Address, Rural Connectivity Summit, Radisson Blu Hotel, Ikeja, Lagos, Nigeria, Oct. 22, 2025, Nigerian Communications Commission (NCC). [Online]. https://ncc.gov.ng/sites/default/files/2026-02/Keynote-Address-by-Dr-Aminu-Maida-at-the-Rural-Connectivity-Summit.pdf.