Received 9 July 2025; revised 16 September 2025; accepted 19 September 2025. Date of publication 25 September 2025; date of current version 17 October 2025. The review of this article was arranged by Associate Editor Dr. Flavio Esposito.

Digital Object Identifier 10.1109/SR.2025.3614166

COTS Small Board Computers for Resilient Sensor Data Acquisition and Processing in Nanosatellites: Challenges and Opportunities

JOÃO ARAÚJO [1], NUNO FILIPE MONTEIRO [2], AND RUI PINTO [3] (Member, IEEE)

(Review Paper)

[1]Departamento de Informática, Faculdade de Engenharia, Universidade do Porto, 4099-002 Porto, Portugal
[2]Altice Labs, 3810-106 Aveiro, Portugal
[3]SYSTEC ARISE, Faculdade de Engenharia, Universidade do Porto, 4099-002 Porto, Portugal

CORRESPONDING AUTHOR: RUI PINTO (e-mail: rpinto@fe.up.pt)

This work was supported by the European Space Agency (ESA) and Altice Labs S.A. through the ARTES 4.0 Strategic Programme Line 5G/6G, under Grant N4000144031/24/NL/EG/If -5G Nanosatellite (Activity Code 3E.022).

ABSTRACT The proliferation of nanosatellites, particularly CubeSats, has not only revolutionized space access but has also driven the demand for efficient and resilient onboard sensor systems. Advanced sensor payloads, ranging from optical imagers to inertial measurement units, generate complex and high-volume data streams that must be reliably acquired, processed, and transmitted under stringent constraints. Commercial-off-the-shelf (COTS) small board computers (SBCs), such as Raspberry Pi and NVIDIA Jetson platforms, offer a cost-effective and computationally capable solution for this onboard data handling. However, their deployment in the harsh low Earth orbit environment presents significant challenges, including the coupled effects of radiation susceptibility, thermal management, and power limitations, which demand robust software resilience. This article reviews state-of-the-art methods for deploying COTS SBCs for sensor data acquisition and processing in space, synthesizing mitigation strategies and identifying critical research gaps. It highlights the necessity for standardized holistic testing frameworks, the development of hierarchical fault recovery architectures, and secure over-the-air update mechanisms tailored to sensor-driven nanosatellite missions. The findings lay the foundation for developing next-generation, resilient nanosatellite sensor systems leveraging COTS computing platforms.

INDEX TERMS Commercial-off-the-shelf (COTS) small board computers (SBCs), nanosatellites, over-the-air (OTA) updates, power management, radiation effects, reliability, software resilience, space environment, space qualification, thermal management.

I. INTRODUCTION

The past two decades have witnessed a significant transformation in space exploration and utilization, primarily driven by the advent of nanosatellites, particularly those adhering to the CubeSat standard [1]. These small spacecraft, typically weighing up to 2 kg per unit [1], [2], have dramatically lowered the barriers to space access, enabling a diverse range of actors, including universities, startups, and research institutions, to conduct missions previously feasible only for large governmental agencies [3]. Nanosatellites are now employed in various applications, from Earth observation and remote sensing to telecommunications, technology demonstration, and scientific research, fostering innovation through rapid development cycles and constellation deployments [3], [4].

A central driver of these missions is deploying increasingly sophisticated onboard sensor systems, including high-resolution imagers, hyperspectral sensors, radiation dosimeters, and environmental monitors, which generate vast volumes of complex data. Managing these sensor data onboard requires reliable acquisition and efficient real-time processing to optimize limited downlink bandwidth and increase mission autonomy.

Central to the operation of any satellite is its onboard computer (OBC), which is responsible for command execution, data handling, payload management, and communication [2], [5]. Traditionally, OBCs relied on expensive, radiation-hardened processors with relatively modest computational capabilities, representing a significant cost and development bottleneck, especially for budget-constrained nanosatellite missions [6]. A compelling alternative has emerged in the form of commercial-off-the-shelf (COTS) small board computers (SBCs), such as the popular Raspberry Pi, including its Compute Module variants, such as the Compute Module 4 (CM4) and Compute Module 5 (CM5) that contain error correction code (ECC) memory [7], and NVIDIA Jetson series [8], [9], [10], [11].

These platforms offer substantial advantages: drastically lower costs, significantly higher computational performance, including graphics processing unit (GPU)-dependent capabilities on some platforms and artificial intelligence (AI), vast software ecosystems, readily available development tools, and large user communities, enabling faster prototyping and integration [12]. Their small form factor, flexible sensor interfacing capabilities, and relatively low power consumption enhance their appeal for volume- and power-limited nanosatellite designs. Various missions demonstrate the potential and limitations of integrating COTS SBCs into space environments, providing valuable insights into their performance, adaptability, and resilience. This article's focus is to propose and validate a portable software architecture on a representative COTS platform (which includes Arm Cortex-A-based processors) rather than comparing systems themselves.

The *AstroPi* project by the European Space Agency (ESA) exemplifies the effective use of Raspberry Pi SBCs in space missions. Deployed aboard the International Space Station (ISS) in two instances, first in 2015 using the Raspberry Pi 1 Model B+ and later in 2021 with the Raspberry Pi 4 Model B+ (codenamed *Astro Pi Mark II*). These units were equipped with various sensors to enable students worldwide to develop experiments in the unique microgravity environment of the ISS [9]. While primarily educational, the project highlights the computational capabilities and adaptability of COTS SBCs, even in controlled but challenging environments. The success of the *AstroPi* is primarily attributed to the usage of 6063-grade aluminum for its casing (see Fig. 1), providing the effective conduction cooling and shielding needed, a critical design step that ensured reliable operation in space [9].

The Virtual SuperOptics with Reconfigurable Swarms (*VISORS*) CubeSat mission leverages Raspberry Pi technology in its payload avionics interface board to support a distributed telescope system. This mission demonstrates the feasibility of integrating COTS SBCs for highly specialized and collaborative scientific endeavors. Kolhof and Lightsey [13] emphasized the importance of robust design practices to

FIGURE 1. ***Astro Pi Mark II* Flight Case [9].**

mitigate risks associated with radiation and thermal extremes, which are critical for ensuring reliable operation in low-Earth orbit (LEO).

The *Aalto-1* nanosatellite, developed by Aalto University, implemented a Linux-based OBC to manage telemetry, payload operations, and data handling [5]. Using embedded Linux on custom hardware inspired by off-the-shelf SBC designs, *Aalto-1* demonstrated the viability of lightweight, modular software architectures for CubeSats. The mission highlighted the importance of balancing computational demands with power efficiency in constrained environments.

In the *SUCHAI* nanosatellite program, Gonzalez et al. [14] implemented a modular flight software architecture to enhance reusability and adaptability. The program showcased how COTS components, including SBCs, can be effectively integrated into nanosatellite missions with agile development methodologies. These systems emphasize fault tolerance, extensibility, and low power consumption, making them suitable for rapid prototyping and deployment.

The diverse applications of COTS SBCs in missions, such as *AstroPi*, *VISORS*, *Aalto-1*, and *SUCHAI*, illustrate their versatility and cost-effectiveness. Despite their terrestrial advantages, deploying COTS SBCs in space, even in LEO, presents great challenges [2], [12]. These devices are fundamentally designed for benign, ground-based environments and lack the inherent robustness of traditional space-grade components. The LEO environment exposes electronics to a harsh combination of factors: ionizing radiation, including total ionizing dose (TID), which causes cumulative degradation, and single event effects (SEEs), such as bit-flips (SEUs) or latch-ups (SELs) induced by high-energy particles, which can corrupt data or cause catastrophic hardware failure [15], [16].

Furthermore, the vacuum of space eliminates convective cooling, demanding alternative thermal management strategies to handle heat generated by increasingly powerful processors. At the same time, rapid transitions between sunlight and shadow impose extreme thermal cycling [12], [17]. Finally, nanosatellites operate under stringent power constraints dictated by small solar panels and batteries, limiting the energy available for computation and operations [5], [12]. This fundamental mismatch between COTS design and the space environment poses significant risks to computation reliability and onboard sensor data acquisition and processing.

Addressing the challenges of using COTS SBCs reliably in space, particularly for resilient sensor integration and data processing, requires a thorough understanding of the risks and available mitigation techniques. This review aims to synthesize the current state-of-the-art regarding the deployment of COTS SBCs in nanosatellite missions, explicitly focusing on their role in supporting sensor-driven operations. It examines the primary environmental and operational challenges encountered, evaluates the hardware and software mitigation strategies proposed and implemented in the literature, and identifies critical gaps in current research and understanding. The scope of this review primarily focuses on the LEO environment, commonly utilized COTS SBC platforms (e.g., Raspberry Pi and NVIDIA Jetson families), and the key challenges of radiation effects, thermal management, power efficiency, software resilience, the implementation of over-the-air (OTA) updates, and robust sensor system integration.

In addition to synthesizing the literature, this article draws on empirical lessons from a ground-based framework designed to simulate LEO conditions and serve as a crucial preflight validation methodology for the up-and-coming *Altice Labs' 5G Nanosatellite* mission, where a COTS SBC is tasked with managing a real-time video feed from a camera sensor to validate 5G nonterrestrial network (NTN) connectivity. This case study provides practical insights into sensor data acquisition systems' integration, testing, and operational resilience. This use case not only illustrates the practical integration of sensor-based COTS SBC payloads but also serves to validate and, in some aspects, challenge existing literature on mitigation strategies for nanosatellite environments.

The rest of this article is organized as follows. Section II outlines the methodology employed for the literature review. Section III delves into the significant challenges and discusses the state-of-the-art mitigation strategies in the literature, covering radiation, thermal, power, software resilience, OTA updates, and sensor data handling. Section IV presents lessons learned from the ongoing *Altice Labs' 5G Nanosatellite* project, emphasizing practical strategies for resilient sensor integration and operation. Section V provides a discussion synthesizing the identified research gaps and their implications. Section VI proposes future research directions based on these gaps. Finally, Section VII concludes this article, summarizing the findings and the outlook for COTS SBCs in space-based sensor systems.

II. LITERATURE REVIEW METHODOLOGY

A rigorous and structured literature review was conducted to assess the state-of-the-art deployment of COTS SBCs for sensor data acquisition and processing in nanosatellite missions. The scope of the review focuses on the application of COTS SBCs in nanosatellite missions, with particular emphasis on their integration with sensor systems, onboard data acquisition, real-time processing, and resilience to space environment challenges, such as radiation, thermal extremes, and power constraints. The objective was to identify current techniques, architectures, and mitigation strategies aligning with developing reliable, self-sustainable, high-performance small satellite platforms.

A hybrid methodology was adopted to ensure comprehensiveness and minimize bias, combining elements of the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework [18] with an exploratory search strategy. This dual approach enabled identifying established research and emerging relevant trends. Fig. 2 represents the literature review methodology followed.

A. SEARCH STRATEGY

The primary search was conducted across major scientific databases, including IEEE Xplore, Scopus, ScienceDirect, and ACM Digital Library. Search queries were formulated using a combination of keywords related to SBCs (*"SBC," "OBC," "Raspberry Pi," and "NVIDIA Jetson"*), nanosatellite platforms (*"CubeSat" and "LEO"*), sensor systems (*"sensor integration," "payload data handling," and "onboard processing"*), and environmental and operational challenges (*"radiation resilience," "thermal management," "fault tolerance," "OTA update"*).

The search period was restricted primarily to publications from 2019 onward to capture recent advancements. However, foundational works and technical guidelines from agencies, such as the National Aeronautics and Space Administration and ESA, were also included where relevant [9], [12].

B. SEARCH QUERIES AND KEYWORDS

Representative search queries included the following.

1) (*"small board computer" OR "onboard computer"*) AND (*nanosatellite OR CubeSat*) AND (*"sensor integration" OR "payload data acquisition"*).
2) (*"radiation resilience" OR "thermal management" OR "single event effects"*) AND (*nanosatellite OR CubeSat*).
3) (*"over-the-air update" OR "fault tolerance"*) AND (*"small board computer" OR "Raspberry Pi" OR "NVIDIA Jetson"*).

Queries were iteratively refined to balance precision and recall, ensuring coverage of relevant works while minimizing irrelevant results.

C. INCLUSION AND EXCLUSION CRITERIA

Articles were selected based on the following inclusion criteria.

1) Focus on COTS SBCs or closely related components in the context of nanosatellite or CubeSat missions.
2) Address sensor data acquisition, onboard processing, or system-level resilience (radiation, thermal, power, and software).
3) Published from 2019 onward, in English, with full-text availability.
4) *Publication type:* peer-reviewed journal articles, conference papers, technical reports, or book chapters.

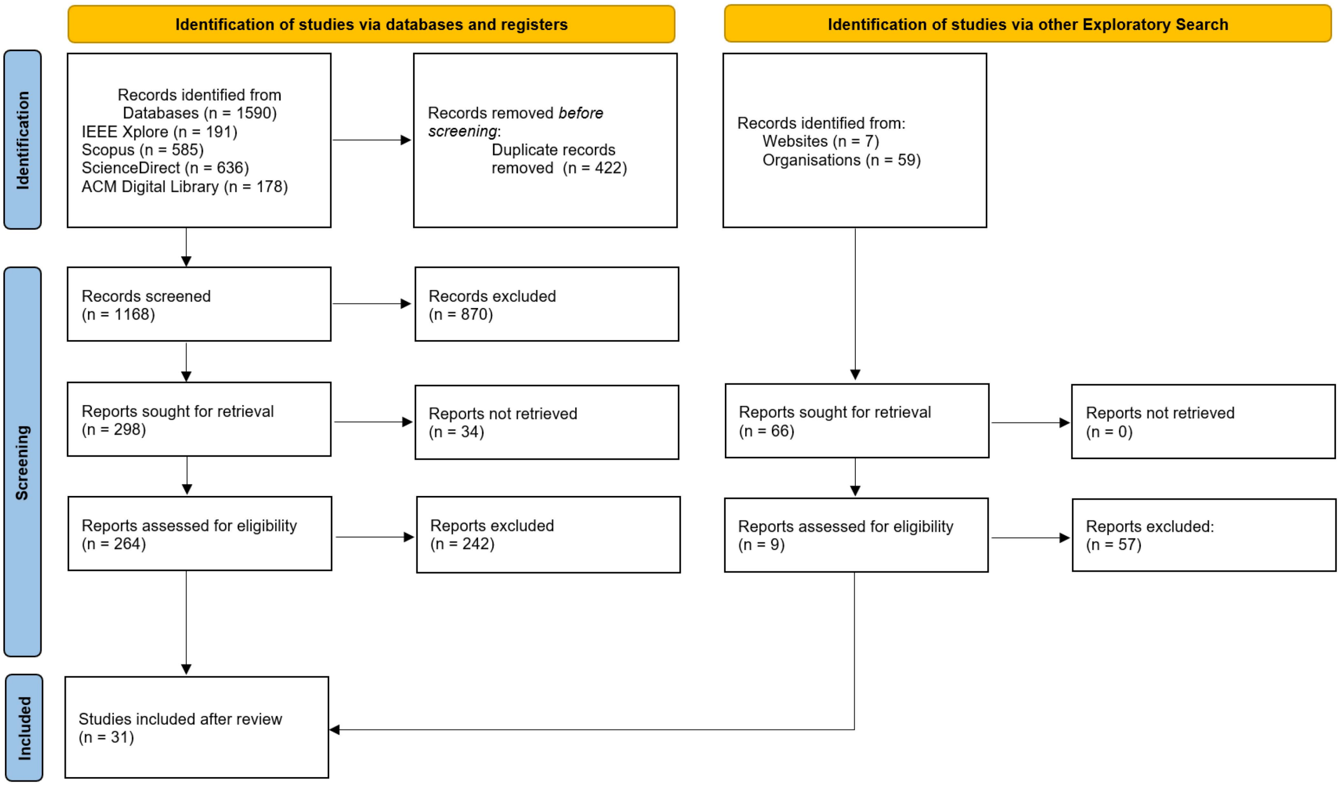

FIGURE 2. **PRISMA and exploratory search flowchart.**

Exclusion criteria were as follows:
1) focus solely on terrestrial applications;
2) studies centered on non-COTS space-grade hardware, unless providing a comparative analysis; and
3) non-English publications, retracted works, or documents without sufficient technical detail (e.g., abstracts only).

D. REVIEW PROCESS

The review process followed a multistage screening approach:
1) removal of duplicates across databases;
2) title and abstract screening based on inclusion and exclusion criteria;
3) full-text assessment for eligibility confirmation; and
4) thematic data extraction and synthesis aligned with identified challenges: radiation effects, thermal management, power efficiency, software resilience, OTA updates, and onboard sensor data handling.

This structured methodology ensured a balanced literature assessment, integrating both the rigor of systematic reviews and the breadth of exploratory searching, ultimately supporting the state-of-the-art synthesis presented in Section III and the research gap discussion in Section V.

III. KEY CHALLENGES AND MITIGATION STRATEGIES

The deployment of COTS SBCs in the demanding space environment necessitates addressing several critical challenges inherent to their terrestrial design origins. This section reviews the primary obstacles identified in the literature. It discusses the current state-of-the-art mitigation strategies employed or proposed for nanosatellite missions, primarily focusing on the LEO environment.

A. RADIATION ENVIRONMENT EFFECTS

The LEO radiation environment poses a significant threat to COTS electronics, which lack the inherent radiation hardening of traditional space-grade components [2], [12]. Key concerns include TID, which leads to cumulative degradation and eventual device failure, and SEEs, such as SEUs or bit-flips in memory and logic, and potentially destructive SELs in complementary metal–oxide–semiconductor devices [15], [16]. These effects can cause data corruption, system crashes, and permanent hardware damage [16], [19].

Mitigation approaches encompass both hardware and software techniques. Hardware strategies often involve component selection, shielding, and redundancy. While COTS SBCs are not inherently rad-hard, some components may exhibit better tolerance than others, necessitating careful selection and screening [11], [12], [20]. Radiation testing campaigns on specific COTS boards, such as the NVIDIA Jetson TX2i (see Fig. 3), have provided valuable data on TID and SEE sensitivity thresholds for components, such as the central process unit (CPU), double data rate synchronous dynamic random-access memory (DDR SDRAM), and Flash memory, identifying critical vulnerabilities, e.g., Flash memory

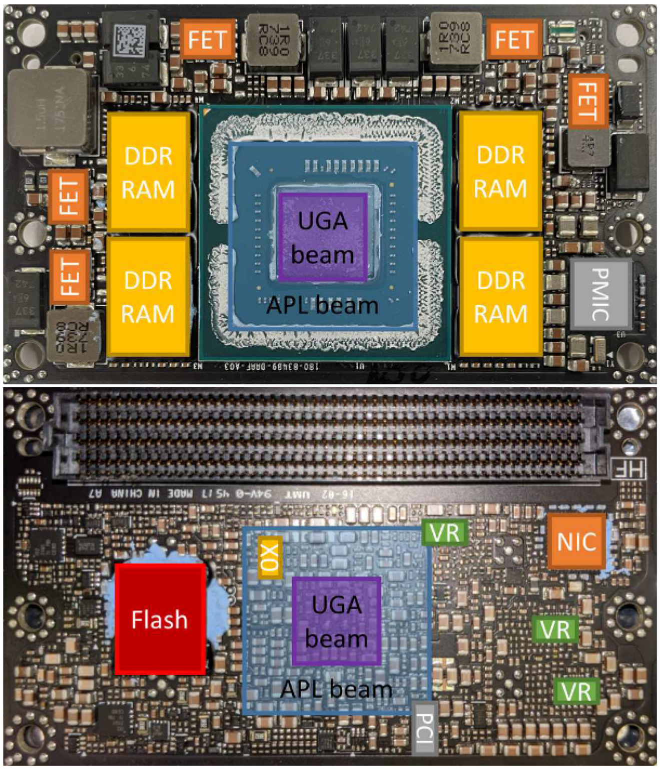

FIGURE 3. Radiation sensitive components on the front (top) and rear (bottom) of the NVIDIA TX2i module [11].

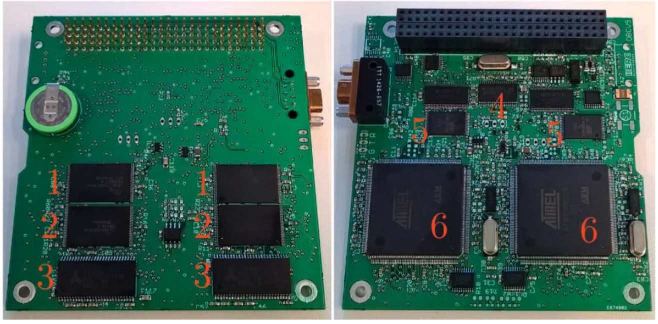

FIGURE 4. *Aalto-1* onboard Linux-based computer. (1) 256 MB NAND Flash, (2) 8 MB NOR Flash, (3) 32 MB SDRAM, (4) MSP430 arbiter, (5) 8 MB DataFlash, and (6) AT91RM9200 microcontroller [5].

failure requiring operating system (OS) reinstallation after 25 krad(Si) [10], [11], [21].

Shielding, often integrated into the satellite chassis or enclosure, e.g., the 6063-grade aluminum case for *AstroPi*, [8]), can reduce TID accumulation and lower SEE rates. However, it adds mass and cannot eliminate high-energy particles. Hardware redundancy, such as triple modular redundancy (TMR), usually implemented using field programmable gate array alongside COTS processors, can mask errors but adds complexity and power consumption [22], [23], [24]. Some newer COTS modules, such as the Raspberry Pi CM5, incorporate basic hardware mitigation, such as ECC on DDR SDRAM memory, offering protection against single-bit SEUs [7].

Software strategies provide complementary protection. Memory scrubbing routines periodically read and rewrite memory areas to correct accumulating SEUs, often in conjunction with hardware ECC [22]. Watchdog timers are commonly used to detect system hangs (potentially caused by SEEs) and trigger resets [14], [25]. Fault detection, isolation, and recovery (FDIR) routines at the software level aim to identify anomalous behavior, potentially isolate faulty processes or memory regions, and attempt recovery actions [14], [26]. Techniques like software-based TMR or process duplication can provide redundancy without specialized hardware [23], [27]. System-level observability tools and frameworks have also been proposed to diagnose radiation-induced faults during ground testing better and potentially in flight [28].

Despite these strategies, comprehensive radiation data across various available COTS SBCs are lacking. Software mitigation effectiveness requires further validation, particularly for complex errors or SELs without hardware power-cycling capabilities. The vulnerability of nonvolatile memory (NVM), such as Flash or Embedded MultiMediaCard (eMMC), during write cycles under radiation is a persistent concern [12], [21].

B. THERMAL MANAGEMENT IN VACUUM

COTS SBCs typically rely on convection for cooling, a mechanism unavailable in the vacuum of space [12]. The heat generated by the processor and other components must be managed primarily through conduction and radiation to prevent overheating, which can degrade performance, reduce component lifetime, and cause failures [2], [17]. Rapid and extreme temperature swings between orbital sunlight and shadow exacerbate thermal stress [17].

Passive thermal control is paramount. This involves careful design of conductive pathways using materials, such as aluminum or copper straps, thermal interface materials, and heat spreaders, to draw heat away from components, such as the CPU or system on chip (SoC), toward the satellite structure or dedicated radiators [2], [12]. The *AstroPi* project's aluminum enclosure is a practical example of conduction cooling [8]. The *Aalto-1* mission demonstrated passive conduction cooling for its Linux-based OBC [5], as represented in Fig. 4. For lower power SBCs (e.g., *Raspberry Pi Zero*) or processors (e.g., *AT91RM9200* used in *Aalto-1*), thermal challenges are less severe due to lower heat generation (sub-100mW range mentioned in [12]).

Active software techniques, such as dynamic voltage and frequency scaling (DVFS), sometimes assisted by onboard temperature sensors, can dynamically adjust performance to manage heat output and optimize thermal cycles, potentially extending system lifetime [17]. Temperature-aware task mapping on multicore processors aims to distribute load and prevent localized hotspots [17]. System-level testing under thermal–vacuum (TVAC) conditions is crucial for validation [2], [20].

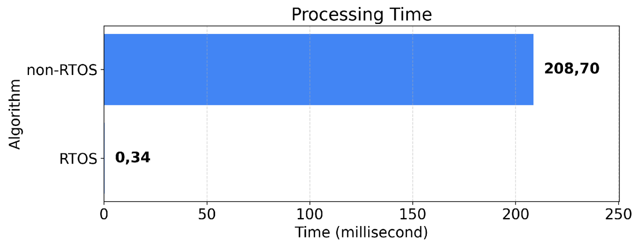

FIGURE 5. **Onboard data handling subsystem processing speeds using RTOS versus non-RTOS to fetch sensorial readings [32].**

Validated thermal models for specific COTS SBCs under vacuum conditions are often unavailable. The effectiveness of purely passive cooling for newer, higher power COTS SBCs needed for demanding tasks, such as running 5G user equipment (UE) [29], remains a challenge. Integrating effective conductive paths within highly constrained nanosatellite volumes requires careful design.

C. POWER EFFICIENCY AND CONSTRAINTS

Nanosatellites typically operate on minimal power budgets [30], [31], often less than 5–10 W, generated by small solar panels and stored in batteries [2], [5]. COTS SBCs, while often designed for low power in terrestrial applications, such as Internet of Things and mobile, may still consume significant power under computational load, potentially exceeding the available budget, especially when peripherals are active [12].

Minimizing power consumption involves hardware selection, power-aware software design, and operational strategies. Selecting energy-efficient processors and components is a primary step [2], [12]. Utilizing low-power sleep states during periods of inactivity is critical [12]. Software techniques include optimizing algorithms and employing lightweight OS or real-time operating systems (RTOS), which can offer more deterministic scheduling and potentially lower overhead compared to general-purpose OS, such as Linux, for specific tasks [32]. Fig. 5 illustrates how the usage of an RTOS allowed the subsystem to fetch parameters, such as temperature, gyroscope, and magnetometer readings, in 0.34 ms versus 208.7 ms with a non-RTOS.

Disabling unused peripherals, such as universal serial bus (USB), high-definition multimedia interface, or Wi-Fi, on a Raspberry Pi via software or hardware modifications is recommended [12]. Data compression techniques can reduce the power needed for data transmission [33]. Modular software architectures allow resource-intensive components to be selectively activated [14]. Collaborative hardening techniques aim to integrate fault tolerance with energy efficiency [26].

Balancing computational performance requirements, e.g., for advanced payloads or communication, with strict power limits is a constant tradeoff. Accurate power profiling of COTS SBCs under realistic, varying flight software loads is complex but necessary. Optimizing power consumption for tasks involving peripherals (e.g., radios and cameras) requires careful system-level integration.

D. SOFTWARE RESILIENCE AND SELF-HEALING

Beyond immediate radiation effects, ensuring long-term software stability and the ability to recover autonomously from unforeseen states (crashes, hangs, and corruptions) is crucial for mission success, especially given limited ground contact [14], [19]. COTS operating and file systems may not be robust against abrupt power losses or radiation-induced memory corruption.

Robustness is built through multiple layers. Watchdog timers (hardware or software) are fundamental for detecting system unresponsiveness and forcing a reboot [14], [25]. Techniques for ensuring reliable boot include redundant bootloaders and boot sequence validation [5], [14]. File system integrity is maintained using journaling file systems, such as the fourth extended filesystem (*ext4*) or checksumming, although corruption due to SEUs or power loss remains a risk [34]. As mentioned previously, ECC memory [7] and memory scrubbing [22] address RAM corruption. A key strategy for system-level resilience involves redundant software partitions: maintaining separate, validated partitions for the OS and application software, e.g., an active and inactive/backup partition. If an update fails or the active partition becomes corrupted, then the system can revert to the last-known-good state in the backup partition [5], [14]. This often includes a “golden image” or minimal rescue system. Process duplication or virtualization can provide redundancy at the process level [24], [35]. Self-healing strategies aim to detect anomalies, diagnose faults, and trigger automated recovery procedures, potentially including selective restarts or reinitialization of subsystems [26], [36].

Implementing robust dual-partitioning and rollback mechanisms requires careful design of the bootloader and update process. Ensuring file system integrity against all failure modes (especially power loss during writes on nonideal storage, such as SD cards) is difficult. The effectiveness of self-healing mechanisms often depends on the ability to accurately diagnose the root cause of a problem, which can be challenging in complex COTS systems.

E. OTA UPDATES IN SPACE

Beyond immediate self-healing from transient faults, long-term mission viability often depends on the ability to update software remotely. This OTA update capability serves two critical resilience functions: first, as a mechanism to deploy patches for latent software bugs or security vulnerabilities that cannot be resolved by a simple reboot, and second, as a means to adapt to changing mission parameters or add new capabilities, extending mission lifetime and value [2]. However, implementing reliable OTA updates for nanosatellites is challenging due to low-bandwidth, high-latency, and intermittent communication links, coupled with the severe consequences (bricking) of a failed update [19], [37].

Robust OTA mechanisms are therefore an extension of the system's fault tolerance strategy, often leveraging techniques discussed under software resilience. Dual-partitioning is fundamental, allowing updates to be downloaded and verified on an inactive partition before attempting a switch, with a clear rollback path if the new software fails [5], [14]. File transfer protocols need to handle interruptions and potential corruption; lightweight protocols, sometimes based on the user datagram protocol with forward error correction or trivial file transfer protocol/secure shell file transfer protocol adaptations with retransmission capabilities, are used [5], [38], [39]. Update packages are often split into smaller chunks to manage limited bandwidth and allow resumption after interruptions [5]. Verification using cryptographic hashes (e.g., *SHA-256*) ensures update integrity before application [14]. Modular software architectures facilitate smaller, targeted partial updates rather than replacing the entire system image [14]. In addition, delta update mechanisms, which transmit only the binary differences between software versions, are utilized to reduce the amount of data significantly transferred [14].

Despite its importance, OTA capability is often deprioritized or absent in many research-focused CubeSat missions [40]. There is a lack of standardized, widely adopted, lightweight, and secure OTA protocols specifically designed for the constraints of nanosatellite communication links. Ensuring atomicity (the update either fully succeeds or fails, leaving the system stable) across the entire process (download, verify, deploy, reboot, and confirm) is complex. Security aspects, while sometimes deprioritized for academic missions, are becoming increasingly important for commercial constellations but add overhead [2].

F. ONBOARD SENSOR INTEGRATION AND DATA PROCESSING

Nanosatellite missions increasingly rely on sophisticated sensor payloads, including high-resolution cameras, hyperspectral imagers, radiation dosimeters, and Global Navigation Satellite System receivers, to fulfill mission objectives, such as Earth observation, environmental monitoring, and scientific experimentation [3], [4], [41]. These sensors generate large volumes of data, creating significant acquisition, storage, and transmission challenges, especially under the stringent constraints of LEO operations.

Onboard data processing has emerged as a crucial strategy to reduce downlink requirements and enhance mission autonomy [42]. COTS SBCs offer an appealing platform for this purpose, providing sufficient computational capability to perform real-time preprocessing, compression, and even preliminary data analysis onboard [10], [12], [43], [44]. SBCs enable flexibility in interfacing with diverse sensors through standardized protocols, such as interintegrated circuit, serial peripheral interface, universal asynchronous receiver–transmitter, and Ethernet, facilitating rapid integration of heterogeneous payloads [5].

While the core resilience challenges for the SBC platform are universal, the specific nature of the sensor payload introduces unique data processing and reliability requirements that must be addressed at the system level. For optical sensors, such as the charge-coupled device (CCD) detectors used in space-borne imaging and lidar missions, a primary challenge is mitigating radiation-induced artifacts. A detailed postmission analysis of the Aeolus satellite's ALADIN instrument by Lux et al. [45] provides a comprehensive case study. The study documents the emergence of "hot pixels," pixels with anomalously high dark current, caused by high-energy particles in the LEO environment. Mitigating their impact on the final wind data products required the development of dedicated, regular in-orbit calibration campaigns and sophisticated ground processing to characterize their complex behavior, including random telegraph signal (RTS) noise.

Furthermore, the study demonstrates the critical interplay between the radiation and thermal environments, a coupled effect noted as a key challenge in system reliability. They performed dedicated in-orbit tests to analyze the temperature dependence of the dark current originating from these radiation-damaged pixels. The findings showed that the thermal behavior of these "hot pixels" differed significantly from that of nominal pixels, underscoring the necessity of precise thermal control and characterization to maintain the calibration and data integrity of sensitive optical payloads. This illustrates that a resilient onboard system must not only handle radiation effects but also manage their thermal implications to ensure reliable sensor performance.

For inertial sensors, such as inertial measurement units (IMUs) critical for attitude determination and control systems, the primary challenges for an OBC involve real-time data fusion and filtering to compensate for inherent sensor noise, drift, and external disturbances. A study by Wei et al. [46] highlights this challenge, particularly the degradation of attitude accuracy under dynamic acceleration, where external forces corrupt the accelerometer's ability to provide a stable gravity reference while gyroscope data accumulate integration errors over time. To address this, the authors propose a robust adaptive error state Kalman filter. The OBC is tasked with executing this computationally intensive algorithm to fuse sensor data, adaptively adjusting its trust in the accelerometer data based on real-time motion detection. The resilience of the SBC is therefore paramount, as a processing failure or delay could lead to a loss of attitude knowledge, jeopardizing the entire mission, including the ability to point other scientific payloads or communicate with the ground.

These examples highlight that a successful sensor-driven mission requires a holistic design where the SBC's resilience strategies are tailored to the specific needs of the sensor payload. Table 1 synthesizes these critical interdependencies, incorporating the specific challenges identified for different sensor modalities.

As these interdependencies show, emerging trends advocate for edge computing approaches, where intelligent processing is performed directly onboard using lightweight machine

TABLE 1. **Relationship of Onboard Sensor Integration With Key Challenges and Mitigation Strategies**

Challenge Area	Impact on Onboard Sensor Integration	Relevant Mitigation Strategy
Radiation Environment Effects	• Causes ”hot pixels” and RTS noise in optical sensors (e.g., CCDs), degrading data quality [45]. • Flash storage for sensor data is highly vulnerable during write operations. • Potential for SEUs in SBC memory to corrupt in-flight sensor processing algorithms.	• In-orbit calibration routines and advanced ground processing to characterize and mitigate sensor artifacts. • RAM buffering and robust filesystems to protect stored sensor data. • Selective shielding for critical sensor and SBC components.
Thermal Management in Vacuum	• Alters dark current characteristics in optical sensors, requiring thermal characterization to maintain calibration [45]. • Induces bias and drift in inertial sensors (IMUs), affecting attitude determination accuracy. • SBC overheating risks instability in real-time data processing pipelines.	• Passive thermal control (heat spreaders, thermal straps). • Onboard thermal modeling and compensation algorithms. • Comprehensive TVAC testing [47] to qualify sensor and SBC performance across orbital temperatures.
Power Efficiency and Constraints	• Continuous or high-rate sensor acquisition increases power demand. • Computationally intensive filtering (e.g., Kalman filters for IMUs) consumes significant power. • SBC and sensor duty cycles must balance data needs with energy availability.	• DVFS on SBC to match processing power to sensor needs. • Event-driven sensing and duty-cycling of sensor operations. • Power-aware algorithm design for onboard processing.
Software Resilience and Self-Healing	• Onboard execution of complex filtering algorithms (e.g., Kalman filters for IMUs) is required to mitigate sensor noise and drift [46]. • Faults in SBCs can disrupt continuous sensor data acquisition and corrupt data streams. • Sensor drivers and acquisition processes must autonomously recover from software-induced errors.	• Robust implementation of state estimation filters within a fault-tolerant software framework. • Watchdog timers for critical sensing and processing tasks. • Health monitoring of sensor data pipelines with automated restart capabilities.
OTA Updates in Space	• Sensor calibration parameters or onboard processing algorithms (e.g., filter tuning) may require updates post-launch. • OTA failures risk continuity of sensor data collection and processing.	• Secure, redundant OTA update frameworks (A/B partitioning). • Delta updates to minimize bandwidth use for updating sensor firmware or algorithm parameters. • Cryptographic validation of update bundles before deployment.

learning models to detect relevant features or anomalies before data transmission [48], [49]. SBCs equipped with GPU accelerators (e.g., NVIDIA Jetson series) or specialized AI processors are particularly suited for such tasks, enabling real-time analytics on sensor data streams [2], [11]. Techniques, such as onboard image classification, change detection, and event-based data prioritization, are actively explored to maximize the scientific return under limited bandwidth constraints [31], [50].

Ultimately, reliable onboard sensor data handling requires a holistic approach combining robust hardware design, modular and fault-tolerant software architectures, energy-efficient data acquisition pipelines, and integrated health monitoring systems. The practical insights from missions are crucial to bridge the gap between these theoretical frameworks and operational success. In the following section, we present lessons learned from the Altice Labs' 5G Nanosatellite project, where a COTS SBC is employed to manage real-time video acquisition from a camera sensor for 5G NTN communication validation, illustrating the tangible challenges and mitigation strategies in integrating and operating these systems in orbit.

IV. ONGOING LESSONS FROM A SENSOR-BASED SBC USE CASE

While theoretical reviews provide valuable insights into system-level challenges and mitigation strategies, practical deployments are essential to validate these concepts under real-world conditions. The *Altice Labs' 5G Nanosatellite* project with ESA [29] aims to validate 5G connectivity in NTNs by demonstrating live video transmission from orbit, imposing stringent computational and reliability demands on its COTS SBC payload. The SBC must execute a full 5G UE stack and manage and route high-throughput video data from an onboard camera. Sustained, reliable operation of these functions is paramount for mission success, directly conflicting with the known COTS vulnerabilities. Thus, the central problem of this project is the development of a resilient software architecture that enables a COTS Arm Cortex-A-based system-on-module (SoM) to meet these specific, demanding mission objectives within the LEO environment. The main hypotheses of this project are as follows.

1) *A COTS SBC, when integrated with a multilayered, software-defined radiation mitigation strategy—encompassing error detection and correction techniques for memory, robust hardware watchdog integration, comprehensive process monitoring, and selective reboot capabilities—will demonstrate enhanced operational resilience. This resilience will be characterized by the sustained execution of critical functionalities, including 5G UE operation and continuous video transmission, maintaining a predefined high level of service availability under simulated LEO thermal and radiation conditions representative of the target mission profile.*
2) *The implementation of a secure and robust OTA update mechanism, featuring atomic update procedures, cryptographic verification of software packages, and a dual-partition scheme complemented by a verified golden fallback image stored in NVM, will enable the system to achieve a high success rate for software updates and rollbacks. This reliability is anticipated even in the presence of simulated communication link intermittency or nondestructive radiation-induced faults occurring during the update process.*
3) *Despite its COTS nature, the selected Arm Cortex-A-based SoM, when augmented with the proposed software-defined resilience techniques and appropriate thermal management strategies, can meet or exceed the specified performance benchmarks for its primary mission objectives. These objectives include reliably handling 5G UE functionalities and consistently transmitting a high-quality live video feed over a 5G NTN link under the combined stresses of the LEO environment for an operationally significant mission duration.*

Although the *Altice Labs' 5G Nanosatellite* project has not yet flown, its ground-based integration and test campaign has already yielded several practical insights. This mission relies on a COTS-based payload tasked with real-time video acquisition and transmission over a 5G NTN link. The payload architecture, which pairs a Raspberry Pi CM5 for video processing (*PL2*) with another for 5G communications (*PL1*), offers a practical basis for evaluating the integration, qualification, and operational resilience of sensor-driven SBC systems in a representative LEO environment. Fig. 6 contains a mock board of the dual-payload architecture.

The payload's primary objective is to validate the performance of 5G NTN connectivity for high-throughput, real-time data applications from LEO. The payload comprises two interconnected COTS SBCs.

1) *An Observation Payload (PL2):* A Raspberry Pi Compute Module responsible for acquiring a real-time video stream from a camera sensor and encoding it for transmission.
2) *A Communications Payload (PL1):* A second Raspberry Pi Compute Module running a complete 5G UE software stack, which receives the video stream from *PL2* and manages its transmission to the ground.

PL2 orchestrates the entire data acquisition pipeline: capturing the video stream from the camera sensor, encoding it in real time to *H.264*, and using PL1 to stream it via real-time streaming protocol to the ground stations relying on the communications payload. This demanding, sensor-driven task subjects the COTS system to significant computational, thermal, and storage stresses. It is an excellent case study for evaluating the resilience strategies discussed in this article.

A. CHALLENGES AND LESSONS LEARNED IN SENSOR–SBC INTEGRATION

Integrating the camera, SBCs, and 5G software-defined radio (SDR) reveals the critical interplay between the environmental and operational challenges outlined in Section III .

1) RADIATION HARDENING BY DESIGN—MITIGATING EFFECTS ON NONHARDENED COMPONENTS

Nonradiation-hardened components, particularly the eMMC flash memory, exhibit a known susceptibility to ionizing radiation, consistent with findings from prior studies on similar COTS hardware [11]. A SEU affecting the eMMC during a write operation or corrupting the bootloader could lead to catastrophic data loss or render the system inoperable. A strategy combining a minimized bootloader with a multitiered NVM architecture was implemented to mitigate these risks.

The immutable bootloader's footprint on the unprotected eMMC is minimized, containing only the Master Boot Record and essential first-stage boot code. Critical components, including the Linux kernel, are stored in protected, redundant partitions to reduce the single-point-of-failure attack surface. While embedding the kernel within the bootloader could reduce storage redundancy, it would significantly enlarge the

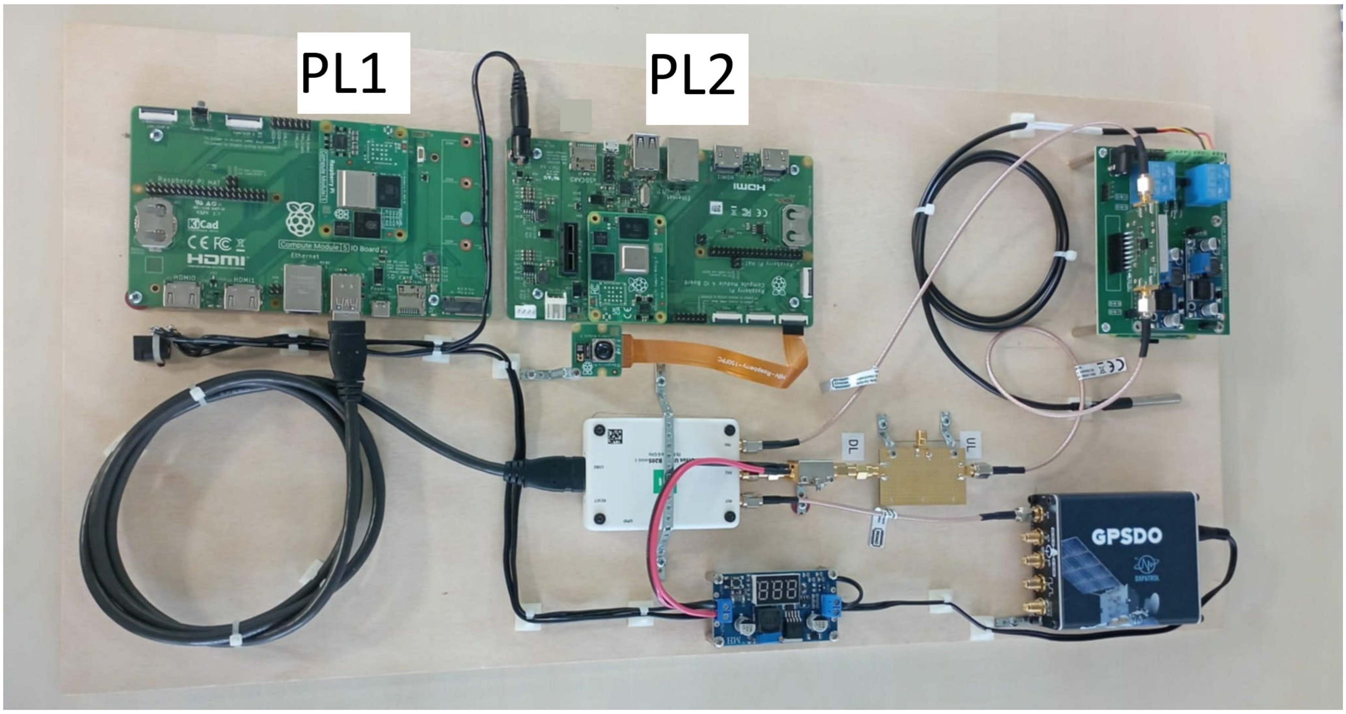

FIGURE 6. Dual-payload mock of the dual-payload Architecture for SBC-based 5G and sensor integration. It contains a CM5 as PL1, CM4 as PL2, a USRP B205-mini acting as the SDR, a Raspberry Pi Camera Module V3, and a GPS disciplined oscillator.

unprotected bootloader section, thereby increasing vulnerability to irrecoverable faults.

Regarding storage resilience testing, we manually corrupted partitions (A/B) to validate boot recovery. CRC-verified rollback from A to B successful. The bootloader fallback and golden image recovery were validated through induced fault simulations. We conclude that the system successfully recovered from partition-level corruption with 100% rollback success in six trials, confirming the robustness of the A/B partition and golden image architecture.

1) *Lesson 1—A multilayered NVM architecture is non-negotiable for data integrity:* A multilayered NVM architecture was determined to be essential for data integrity, as reliance on a single robust filesystem is insufficient. The implemented architecture combines an A/B redundant root filesystem for atomic updates, a minimal read-only rescue partition, and a pristine "golden image" for complete system restoration. Laboratory validation demonstrated that this configuration enables autonomous recovery to a known-good state even if both primary partitions are corrupted. Fig. 7 illustrates the repair flow when the system is fully booted, as the process begins by verifying the integrity of the golden image before proceeding with an iterative restoration of the damaged system partitions.
2) *Lesson 2—The bootloader is a critical single point of failure:* The bootloader requires robust hardening as a critical single point of failure. The boot process is fortified using a redundant U-Boot environment protected by *CRC32* checksums. U-Boot automatically fails to a redundant copy if the primary environment is corrupted. Should both copies fail, it reverts to a default, immutable environment compiled into the bootloader binary. This layered defense ensures the system can reliably initiate a boot sequence, a prerequisite for any subsequent recovery action.

2) THERMAL MANAGEMENT AND ENVIRONMENTAL STRESS

Thermal characterization has thus far been limited to HVac, with TVAC testing pending. Under sustained 1080p30 video encoding, preliminary measurements indicate that the SoC temperature stabilizes at a delta of 7 °C–8 °C above ambient for temperatures below 60 °C. Above 60 °C, this delta increases to 10 °C–11 °C, approaching the SoC's thermal throttling threshold (80 °C) with passive dissipation, with power usage being increased from 1.38 to 1.56 W. These observations, which validate operational feasibility under moderate thermal loads, are supported by the existing literature [12], [17], inform the following mitigation.

1) *Lesson 3—Dynamic load-aware throttling mitigates hotspots:* A *systemd*-managed daemon continuously monitors system temperature, resource usage, and proactively throttles or terminates noncritical, low-priority services when thermal thresholds are breached, proving an essential active mitigation strategy. This prevents a thermal runaway condition from a secondary task from jeopardizing the primary mission under critically high temperatures.

Future work includes developing a high-fidelity thermal model of the SoC to predict temperature profiles throughout

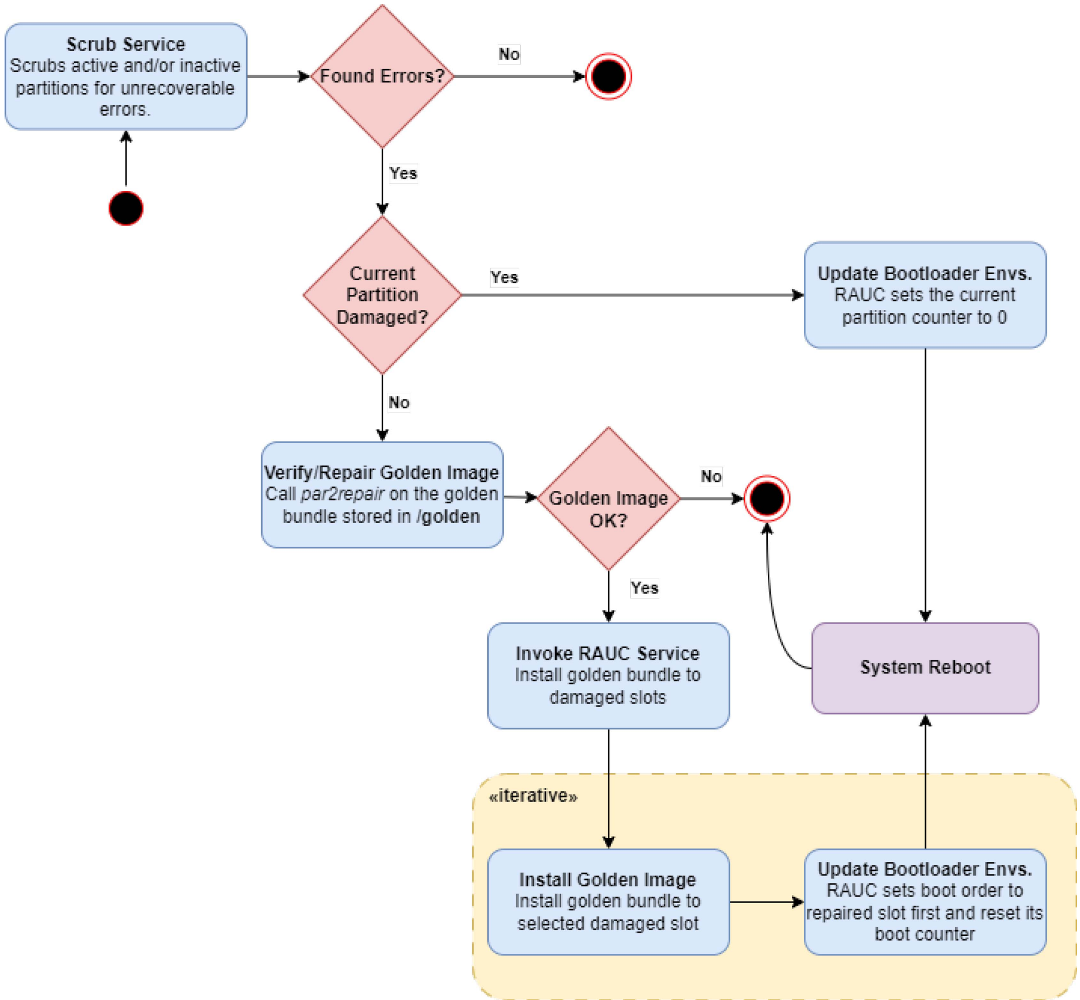

FIGURE 7. Flowchart of the automated repair workflow. The process begins by verifying the integrity of the PAR2-protected golden image before proceeding with an iterative restoration of damaged system partitions, ensuring a robust disaster recovery capability.

LEO sunlight/eclipse cycles, enabling predictive load balancing across CPU cores. Further, coupling thermal tests with radiation will likely demonstrate the need for a load-balancing profile.

B. HIERARCHICAL RESILIENCE: FROM PROCESS RESTART TO OTA UPDATE

Ensuring data integrity and continuous operation requires a multilayered resilience strategy that can gracefully handle faults of increasing severity. In our architecture, this is achieved through a hierarchical FDIR system where the OTA update mechanism serves as the ultimate recovery tool. All resilience mechanisms have been validated in ground tests by inducing SEU-like damage and service-level failures.

To test software resilience, service crashes were induced by stress testing the camera pipeline. We observed that the first level of the FDIR system restarted the service in all scenarios relying on systemd. For more severe, system-level hangs, a watchdog-induced reboot and partition repair took approximately 251 s for the worst-case scenario of repairing both partitions on the slower system (the CM4). These ground tests provide quantitative data on the system's recovery timeline. In addition, to formally describe the system's recovery hierarchies, the architecture's behavior is modeled as the finite state machine (FSM) depicted in Fig. 8. This FSM defines the primary operational modes and the automated transitions that govern system behavior in response to faults, commands, and recovery actions.

a) *Lesson 4—A Hierarchical FDIR Provides Graduated Fault Response:* A centralized FDIR daemon, complemented by a hardware watchdog timer (HWT), provides a tiered response to failures. This hierarchy is crucial for maximizing uptime.
   a) *Level 1 (Process Restart):* For isolated service failures (e.g., the camera pipeline crashing), the FDIR daemon first attempts a simple process restart. This is the fastest recovery, resolving transient software glitches with minimal system disruption.
   b) *Level 2 (System Reboot):* If a service fails repeatedly, or if the HWT detects a full system hang

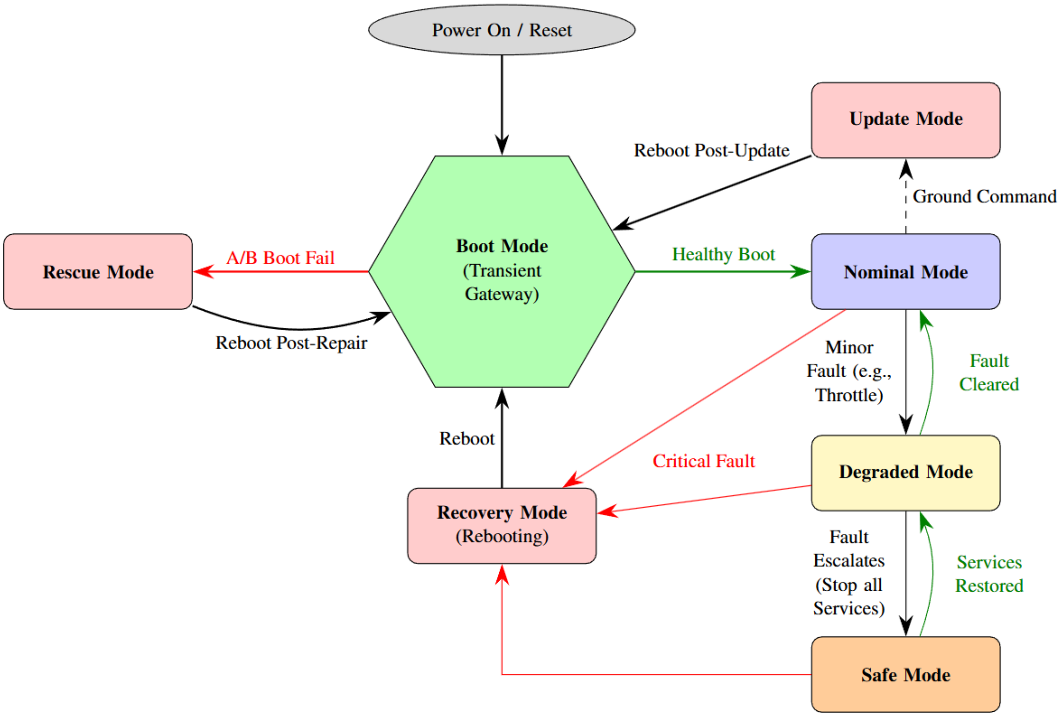

FIGURE 8. High-level system's operational state machine. The FSM defines the autonomous transitions between modes based on fault detection, providing a formal model for the hierarchical recovery logic.

(e.g., due to memory corruption or a kernel panic), then the system escalates to a full reboot. The robust bootloader (see Lesson 2) ensures the system can return to a known-good state on a primary or backup partition.

c) *Level 3 (OTA Recovery):* If a persistent software bug causes repeated crashes even after reboots, then the fault is considered unrecoverable by autonomous means. In this scenario, the OTA update mechanism becomes the final layer of the FDIR system. By deploying a patched software image to the inactive partition, ground operators can perform the ultimate recovery, fixing the root cause of the fault. This transforms the OTA system from a simple feature-update tool into the cornerstone of long-term mission resilience.

The update strategy itself must balance the constraints of limited satellite contact windows, available storage, and onboard processing power. Our implementation uses the RAUC framework [51], which supports both full-image and delta-based updates, allowing operators to choose the optimal method based on the situation. For OTA validation, we deployed three update cycles, achieving 100% successful update/rollback in all test cases, with recovery from an induced mid-transfer failure taking under 60 s. The atomicity of this process, guaranteed by cryptographic verification before switching partitions, prevents the satellite from being "bricked" by a partial or corrupted update.

C. DATA INTEGRITY AND PROCESSING RESILIENCE

All resilience mechanisms have been validated only in ground tests, by inducing SEU-like damage to eMMC and RAM. Ensuring the end-to-end reliability of both systems requires a multilayered software resilience approach, moving beyond hardware considerations.

Service crashes were induced by stress testing the camera pipeline to test software resilience. We observed that the FDIR system restarted service in under 10 s in all scenarios. Also, a complete reboot + dual-partition repair on the CM4 in worst-case recovery took under 12 min. In sum, the average FDIR service recovery time is 8.6 s. Full system recovery via watchdog-induced reboot takes 72 s.

1) *Lesson 5—A hierarchical watchdog and FDIR system complements an HWT:* A centralized FDIR daemon works with the HWT on the implemented solution to provide a graduated response. It first attempts to restart a failed service (e.g., the camera pipeline). If that fails repeatedly, then it will escalate to a full system reboot and a partition repair, based on a local golden image. Furthermore, suppose the system becomes unresponsive due to a major corruption within a service that originates memory leaks. In that case, the HWT will eventually trigger continuous reboots, returning to the passive or rescue partitions. This tiered approach maximizes service uptime while providing an ultimate fail-safe against catastrophic system hangs.

Furthermore, long-term mission viability depends on securely updating software in orbit to address latent defects or deploy new capabilities. The update strategy must balance the constraints of limited satellite contact windows, available storage, and onboard processing power. Two primary update methodologies exist: 1) full-image updates and 2) delta-based updates. Full images are robust and straightforward to deploy on the target, but their large size consumes significant bandwidth. Conversely, delta updates drastically reduce the data volume for transmission but impose greater computational overhead and require additional temporary storage on the device, as a new image must be reconstructed from the old version and the received patch. The optimal choice is mission- and stage-dependent, requiring a flexible mechanism to support both approaches.

For OTA update validation, we used the RAUC framework [51] to deploy three update cycles. We achieved 100% successful update/rollback in all test cases, with recovery from induced fault within 60 s. One update induced failure mid-transfer, which successfully rolled back to the previous image.

1) *Lesson 6—Secure and atomic OTA updates are essential for long-term mission viability:* Acknowledging that postlaunch bugs are inevitable, a robust OTA update mechanism was developed using the RAUC framework and the A/B partitioning scheme, relying on full or delta-based images, allowing operators to select the most appropriate method based on operational conditions. Updates are installed to the inactive partitions and cryptographically verified before a boot attempt is made. This ensures the update process is atomic, i.e., it succeeds completely, or the system automatically rolls back to the previous known-good partition, preventing the satellite from being "bricked" by a partial or corrupted update.

D. PRELIMINARY RESULTS FROM GROUND TESTING

To validate the feasibility and resilience of the sensor-driven SBC payload, ground-based integration and stress testing were conducted, yielding valuable insights into thermal behavior, storage resilience, software fault handling, and OTA update reliability. These preliminary results provide early empirical support for the proposed architecture and mitigation strategies, demonstrating operational viability for nanosatellite deployment. A summary of the key performance metrics and their implications is presented in Table 2.

The thermal results indicate that under realistic computational loads, such as real-time video acquisition and encoding, the CM5 SBC operates within safe temperature margins without active cooling. However, TVAC chamber testing is still required to validate performance under LEO-alike thermal conditions.

The storage resilience tests showed that both soft and hard corruption of primary partitions could be effectively mitigated using a multitiered fallback strategy. CRC-validated U-Boot environments and golden image partitions allowed autonomous recovery without manual intervention.

TABLE 2. Summary of Preliminary Ground Test Results

Metric	Result	Implication
Partition Recovery (A/B + Golden Image)	5/5 successful recovery scenarios	Demonstrates robustness of bootloader and storage architecture under induced corruption scenarios.
Average FDIR Service Restart Time	8.6 s	Ensures high system availability for critical processes (e.g., camera pipeline), with minimal downtime.
Full System Recovery Time (Watchdog Reboot)	∼72 s	Confirms system can autonomously recover from unrecoverable faults and return to safe operational state.
OTA Update + Rollback Success Rate	3/3 update scenarios successful	Validates secure and atomic update mechanism, including automatic fallback to prior image in case of fault.

Service resilience was evaluated by inducing failures in key services. The FDIR daemon demonstrated rapid recovery of camera pipelines and proper escalation when failures persisted, providing a multitiered protection layer against software anomalies.

Finally, the update mechanism was successfully applied, and the RAUC framework verified software packages. Rollbacks occurred reliably in cases of induced failure, reinforcing the architecture's robustness for long-duration missions requiring periodic software maintenance.

Although preliminary and obtained in a laboratory environment, these results strongly support the proposed design choices and underscore the potential of resilient COTS-based architectures for future smart sensor platforms in nanosatellites.

E. TOWARD RESILIENT SENSER-DRIVEN NANOSATELLITE SYSTEMS: KEY IMPLICATIONS AND DESIGN PRINCIPLES

The *Altice Labs' 5G Nanosatellite* project demonstrates that COTS SBCs can support demanding, real-time sensor processing workloads in the harsh space environment, but only when embedded within a rigorously engineered resilience framework. This experience reveals that hardware selection alone is insufficient: system reliability emerges from a software-defined resilience stack that mitigates radiation-induced faults, thermal drift, limited fault observability, and constrained update mechanisms.

In particular, the combination of hierarchical FDIR, a layered NVM architecture, and atomic, secure OTA updates forms the backbone of operational integrity. These strategies are critical to the ongoing mission success of the *Altice Labs' 5G Nanosatellite* payload and represent transferable design principles for future innovative nanosatellite systems.

As sensor-driven tasks grow in complexity, encompassing onboard machine learning, edge analytics, and autonomous

TABLE 3. Summary of Practical Design Challenges and Mitigation Strategies From the Altice Labs COTS SBC Sensor Integration Use Case

Design Challenge	Strategy Implemented	Literature Basis	Preliminary Insight / Lesson
Radiation-induced data corruption	Multi-layered NVM architecture (A/B partitions, rescue, golden image)	Katz et al. [11]; Guertin [12]	Redundancy enables autonomous recovery from SEU-induced corruption
Bootloader vulnerability	Redundant bootloader environments with automatic fallback	Guertin et al. [12]; Leppinen et al. [5]	Minimal bootloader footprint + layered redundancy prevents deadlocks
Thermal hotspots under load	Dynamic load-aware thermal throttling daemon	Kim and Yang [17]; Guertin et al. [12]	Thermal-aware service management avoids throttling and runaway temperatures
System hangs or degraded services	Hierarchical FDIR integrated with HWT	Guertin et al. [12]; Leppinen et al. [5]	Tiered FDIR maintains uptime and guards against persistent failure
Remote software updates in orbit	RAUC-based OTA with cryptographic verification and rollback	Kassing et al. [4]; Weston et al. [2]	OTA ensures system recoverability even with partial/corrupt updates

decision-making, the importance of resilient, upgradeable, and intelligent software frameworks will intensify. Table 3 summarizes the practical design lessons from the *Altice Labs' 5G Nanosatellite* use case project and proposes generalized guidelines for future missions targeting similar sensor integration and processing challenges.

V. SYNTHESIZED GAPS AND IMPLICATIONS

The review of the state-of-the-art mitigation strategies highlights the significant potential of COTS SBCs for nanosatellite missions and the remaining hurdles. While various techniques address individual challenges, such as radiation effects or thermal management, synthesizing the limitations identified across the literature reveals several critical, overarching gaps that hinder the widespread adoption and reliable long-term operation of these components in space. Addressing these gaps is crucial for advancing cost-effective and high-performance nanosatellite capabilities. This section enhances the literature synthesis with empirical insights drawn from the *Altice Labs' 5G Nanosatellite* case study, directly validating and contextualizing key theoretical challenges and strategies.

A. OVERARCHING NEED FOR STANDARDIZED AND HOLISTIC TESTING

A recurring theme emerging from the review is the fragmented nature of COTS SBC evaluation for space applications. While valuable data exist for specific boards under certain conditions (e.g., NVIDIA Jetson series [10], [11]), there is a distinct lack of standardized, comparative, and holistic testing frameworks. Current studies often focus on isolated aspects, such as TID tolerance, SEE sensitivity, or thermal performance in a vacuum. However, the space environment presents these challenges concurrently. There is insufficient research characterizing the performance and reliability of different COTS SBCs under realistic, combined environmental stresses—simultaneous radiation exposure, TVAC cycling, and representative flight software workloads. This creates a significant gap between ground-based characterization and true in-orbit performance, as the full spectrum of environmental stressors cannot be fully replicated preflight.

Empirical Insight: The *Altice Labs' 5G Nanosatellite* project highlights this limitation: although a comprehensive ground-based testbed was developed, TVAC testing remains pending, and only individual stress conditions (e.g., thermal load and eMMC corruption) were tested in isolation. This reflects the difficulty of reproducing combined space-like conditions and reinforces the need for integrated validation campaigns. Ultimately, it underscores that even the most rigorous ground testing serves as a necessary but incomplete proxy for the validation that only in-orbit operation can provide.

B. SOFTWARE MITIGATION FRONTIER: REAL-TIME RESILIENCE

While hardware mitigation like shielding and basic ECC offers a baseline level of protection, the literature reveals significant untapped potential and a necessity for more advanced software-based resilience techniques, particularly those operating in real time with minimal overhead. Current software approaches often rely on lagging indicators (e.g., watchdog timeouts after a crash) or periodic checks (e.g., memory scrubbing). There is limited exploration of dynamic, real-time fault detection mechanisms capable of identifying transient errors before they lead to system failure, especially within the context of non-RTOS commonly used on COTS SBCs, such as Linux [5]. Also, fault detection mechanisms remain underexplored for COTS platforms in space [14].

Empirical Insight: The *Altice Labs' 5G Nanosatellite* use case implemented a hierarchical FDIR system combining service monitoring, a hardware watchdog, and automatic reboot strategies using U-Boot failover and golden images. This demonstrates that resilience can be achieved on COTS Linux-based platforms, yet predictive or proactive detection of faults (e.g., using anomaly detection algorithms) remains absent, validating this gap.

TABLE 4. Cross-Analysis of Literature Gaps and Empirical Findings from the Altice Labs Use Case

Challenge Area	Identified Gaps in Literature	Empirical Evidence from Altice Use Case
Standardized Testing	Lack of holistic, multi stressor benchmarks; no systematic board-to-board comparisons	Ground validation were conducted for thermal and SEU stressors, but no integrated stress testing has been done yet. Confirms the complexity of reproducing space-like multi factor scenarios
Software Resilience	Insufficient real-time mitigation mechanisms; mostly basic watchdogs or ECC	Hierarchical FDIR system demonstrated, showing practical feasibility but lacking predictive mechanisms or adaptive fault handling
OTA Updates	Limited adoption of secure, atomic, and lightweight OTA methods tailored to CubeSat constraints	RAUC-based dual-partition OTA with rollback implemented and validated locally; still lacks in-orbit stress-tested evidence
System Integration	Sparse research on interplay between environmental stressors and software/hardware strategies	Integration challenges noted between thermal, processing load, fault recovery, and OTA timelines; supports need for co-design frameworks

C. ENABLING IN-ORBIT MAINTAINABILITY AND EVOLUTION: ROBUST OTA

Despite the clear benefits for mission longevity and adaptability, implementing reliable OTA update capabilities remains a significant gap, often deprioritized in academic missions [40]. Key challenges include ensuring update atomicity across the entire process [5], efficiently handling interruptions [14], minimizing bandwidth consumption [14], and implementing appropriate security measures [40]. The literature lacks evidence of widely adopted, standardized, lightweight, secure, and demonstrably fault-tolerant OTA frameworks tailored explicitly for nanosatellite constraints.

Empirical Insight: The *Altice Labs' 5G Nanosatellite* adopted the RAUC framework for secure OTA updates with A/B partitioning and cryptographic verification, validated in lab-based induced fault scenarios. While the framework supports both full and delta updates, real-world testing of OTA under constrained bandwidth or intermittent links is still ongoing, highlighting the operational complexity of achieving truly robust OTA.

D. CHALLENGE OF COUPLED ENVIRONMENTAL AND SYSTEM-LEVEL CONSTRAINTS

While this study analyzed stressors in isolation for methodological clarity, in practice, these factors are deeply coupled. The system-level integration and validation of the various mitigation strategies must be studied holistically. Interdependencies between thermal behavior, power draw, radiation effects, and operational procedures, such as OTA updates, create complex failure modes that are difficult to predict. For example, a radiation-induced error can increase processing load, leading to thermal hotspots and increased power consumption, which in turn could impact the stability of other subsystems. Thus, the FDIR system's response, such as throttling or rebooting, must therefore be designed to manage these cascading effects, highlighting the need for holistic, multidomain resilience.

Empirical Insight: The *Altice Labs' 5G Nanosatellite* payload underscores these interactions: thermal-induced performance throttling impacts video encoding, which may in turn affect OTA update windows or fault detection latency. The test campaign confirmed that resilience strategies (e.g., dynamic throttling and watchdog escalation) must be coordinated across subsystems [12], [17]. Still, deeper cross-domain covalidation remains an open need.

Table 4 summarizes the alignment between the synthesized literature gaps and the empirical observations from the *Altice Labs' 5G Nanosatellite* use case. This comparative analysis illustrates how practical deployments can validate and refine theoretical mitigation strategies proposed for COTS SBCs in nanosatellite applications.

These cross-domain insights underline the necessity of a coengineered approach that merges system-level testing, predictive software resilience, and operational robustness. The *Altice Labs' 5G Nanosatellite* project provides preliminary, yet compelling, evidence that COTS SBCs can be viably deployed in space missions if accompanied by an integrated strategy tailored to the environmental and mission-specific constraints. Nevertheless, as this review highlights, this preflight validation remains incomplete without the ultimate test of in-orbit data, reinforcing the urgency for missions to not only implement these strategies but also to report on their flight performance to close the community's knowledge gap.

VI. FUTURE RESEARCH DIRECTIONS

The identified gaps in the current body of literature highlight several key areas where further research is needed to enhance the reliability, capability, and accessibility of COTS SBCs for nanosatellite missions. Table 4 presents a summary of these gaps and the corresponding empirical evidence from the Altice Labs use case. Addressing these areas will be crucial for unlocking the full potential of this cost-effective technology in space. A notable limitation of the current research landscape, and of this article, is the lack of direct, quantitative performance comparisons between different COTS SBC platforms under space-like conditions. This gap is a direct consequence of nonstandardized testing methodologies. Based on the discussion in Section V, the following research directions are proposed to address this and other key challenges.

A. DEVELOPMENT OF STANDARDIZED HOLISTIC EVALUATION FRAMEWORKS

A primary obstacle hindering the reliable adoption of COTS SBCs is the absence of standardized evaluation frameworks. This makes it extremely difficult for mission designers to perform objective, "apples-to-apples" comparisons between different platforms. There is a pressing need to move beyond isolated component testing toward integrated evaluation. Future research should focus on the following.

1) *Designing and validating a standardized test methodology* for COTS SBCs intended for space use. This framework should encompass simultaneous testing under combined conditions relevant to LEO, including realistic radiation exposure (both TID and SEE), thermal–vacuum cycling, and representative electrical loading based on flight software benchmarks.
2) *Conducting systematic comparative benchmarking studies:* Using a standardized framework, researchers should perform direct comparisons of contemporary COTS SBCs (e.g., Raspberry Pi CM4/CM5, various NVIDIA Jetson modules, and other relevant ARM-based platforms). This would provide the community with invaluable and currently unavailable data on critical performance tradeoffs, including radiation tolerance (TID, SEE sensitivity), thermal efficiency under vacuum, and computational performance per watt (i.e., energy-per-instruction metrics). This is essential for informed hardware selection.
3) *Investigating synergistic effects* between different environmental stressors. Research is needed to quantify how operating temperature influences SEE rates or TID tolerance in COTS components and how power cycling strategies impact overall system longevity under combined stress.

B. ADVANCING REAL-TIME SOFTWARE-BASED RESILIENCE

Given the inherent vulnerabilities of COTS hardware, enhancing software resilience is paramount. Key research avenues include the following.

1) *Developing novel, lightweight, real-time fault detection algorithms* suitable for COTS processors (e.g., Arm Cortex-A series) and non-RTOS, such as Linux. These algorithms should aim to detect transient errors (e.g., control flow errors and critical data corruption) with minimal performance overhead.
2) *Designing and implementing advanced autonomous recovery mechanisms* beyond simple reboots. This includes techniques for dynamic memory isolation (detecting and cordoning off faulty RAM or Flash blocks), adaptive task scheduling to work around detected hardware issues, and robust software-level redundancy schemes (e.g., advanced process pairing or micro-virtualization).
3) *Validating software resilience techniques* through extensive fault injection campaigns (both software-based and using radiation testing) to quantify their effectiveness in preventing system failures under realistic error conditions.

C. CREATING ROBUST AND STANDARDIZED OTA UPDATE SOLUTIONS

To improve mission maintainability and longevity, dedicated research into OTA updates for nanosatellites is required.

1) *Developing and standardizing a lightweight, secure, and fault-tolerant OTA protocol* specifically tailored for nanosatellite communication constraints (low bandwidth, high latency, intermittent connectivity). This protocol should ensure updated atomicity and handle interruptions gracefully.
2) *Integrating and validating efficient update mechanisms*, such as delta updates (binary patching), within the context of nanosatellite flight software and resource constraints. Frameworks, such as RAUC or SWUpdate [52], could be adapted and evaluated for space use.
3) *Investigating secure OTA procedures* suitable for nanosatellites, balancing security requirements (authentication, integrity, and confidentiality) with the limited computational and power resources available.

D. FOCUSING ON SYSTEM-LEVEL INTEGRATION AND VALIDATION

Bridging the gap between component-level analysis and mission reliability requires a system-level perspective.

1) *Developing integrated simulation and modeling tools* that capture the complex interactions between thermal, power, radiation, and software behavior in a COTS-based nanosatellite OBC.
2) *Conducting system-level validation campaigns* that test the integrated OBC (hardware and software) under realistic mission scenarios, including nominal operations, stress conditions, fault scenarios, and operational procedures, such as OTA updates, within a simulated space environment (TVAC, radiation).
3) *Exploring co-design methodologies* that explicitly consider the interplay between hardware selection, software architecture (including OS choice, middleware, and resilience mechanisms), thermal design, and power management from the outset of the OBC design process.

VII. CONCLUSION

The increasing sophistication and accessibility of nanosatellite missions are intrinsically linked to advancements in onboard computing and sensing capabilities. COTS SBCs have emerged as promising platforms for supporting complex sensor data acquisition and real-time processing tasks in space, offering significant benefits in computational performance, cost efficiency, and development flexibility compared to traditional space-grade hardware.

This article synthesized the state-of-the-art challenges and mitigation strategies for deploying COTS SBCs in nanosatellite missions, with particular attention to their integration with advanced sensor payloads. Key operational challenges, including radiation effects, thermal management, power constraints, software resilience, and secure OTA updates, were analyzed regarding their impact on reliable sensor operation and data integrity. A comprehensive relationship mapping between sensor system requirements and environmental constraints was provided to guide system designers.

To move beyond theory, this article presented practical lessons from the ongoing *Altice Labs' 5G Nanosatellite* project, offering a grounded, empirical validation of the mitigation strategies discussed. The case study reveals the feasibility of using a dual-SBC COTS architecture to manage real-time video acquisition and 5G NTN communication in a laboratory setup, with expectations to run flawlessly in a space-representative context. These empirical results reinforce the applicability of multilayered software resilience techniques, bootloader hardening, thermal-aware throttling, and OTA frameworks in enabling sustained sensor operation with COTS platforms.

While the presented empirical results stem from a specific mission scenario (a camera-based 5G NTN validation payload), many design insights, architectural strategies, and identified bottlenecks are broadly generalizable. Specifically, the integrated mitigation stack and system-level resilience principles can inform the design of other sensor-driven missions involving COTS SBCs, particularly those with high data throughput or autonomous processing requirements. However, mission-specific variations in orbit profiles, sensing modalities, and fault tolerance thresholds should be considered when adapting these strategies to other use cases.

Combining a comprehensive state-of-the-art analysis with concrete empirical insights lays the groundwork for the systematic development of resilient nanosatellite architectures centered on sensor data acquisition and processing using COTS SBCs. This integrated approach advances the understanding of current challenges and provides a validated reference for system-level design, fault mitigation, and operational strategies. The identified gaps and future research directions offer a roadmap for the evolution of intelligent, autonomous sensor systems in future space missions.

Despite promising advances, critical research gaps remain. There is a pressing need for standardized, holistic testing frameworks that evaluate COTS SBC and sensor systems under combined environmental stresses and for developing lightweight, fault-tolerant, and energy-efficient onboard data processing architectures. As nanosatellite missions increasingly demand intelligent sensing and autonomy, these gaps must be addressed to support the integration of edge computing and onboard analytics under constrained conditions.

Addressing these challenges is essential to fully realize the potential of COTS SBCs as enablers for resilient, adaptive, and high-performance sensor systems in space, paving the way for the next generation of smart nanosatellite missions.

REFERENCES

[1] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation," in *Proc. AIAA/USU Conf. Small Satell.*, 2000, pp. 1–19. Accessed: Oct. 8, 2025. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2069&context=smallsat

[2] NASA Ames Research Center, "State-of-the-art small spacecraft technology," *Nat. Aeronaut. Space Administration, Ames Res. Center*, Moffett Field, CA, USA: NASA Tech. Publication NASA/TP—20250000142, Feb. 2025. Accessed: Oct. 8, 2025. [Online]. Available: https://www.nasa.gov/smallsat-institute/sst-soa

[3] F. Davoli, C. Kourogiorgas, M. Marchese, A. Panagopoulos, and F. Patrone, "Small satellites and CubeSats: Survey of structures, architectures, and protocols," *Int. J. Satell. Commun. Netw.*, vol. 37, no. 4, pp. 343–359, 2019.

[4] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the "Internet from space" with Hypatia," in *Proc. ACM Internet Meas. Conf.*, 2020, pp. 214–229.

[5] H. Leppinen et al., "Developing a Linux-based nanosatellite on-board computer: Flight results from the Aalto-1 mission," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 1, pp. 4–14, Jan. 2019.

[6] P. de Yra et al., "Next-generation spacecraft command and data handling system based on the RAD750 processor," in *Proc. AIAA SPACE 2010 Conf. & Expos., Anaheim*, CA, Sep. 2010, Paper no. AIAA 2010-8888, doi: 10.2514/6.2010-8888.

[7] Raspberry Pi Ltd., *Raspberry Pi Compute Module 5 Datasheet. Raspberry Pi Ltd.*, Cambridge, UK, Oct. 2025. Accessed: Oct. 8, 2025. [Online]. Available: https://datasheets.raspberrypi.com/cm5/cm5-datasheet.pdf

[8] D. Honess and O. Quinlan, "Astro PI: Running your code aboard the International Space Station," in *Acta Astronautica*, vol. 138, pp. 43–52, 2017.

[9] European Space Agency and Raspberry Pi Foundation "Astro PI: Raspberry Pi computers aboard the International Space Station," *Astro Pi Challenge*, 2023. Accessed: Oct. 08, 2025. [Online]. Available: https://astro-pi.org/

[10] R. B. Wyrwas, "NEPP single event effects test report for the NVIDIA Jetson TX2 system on module (SOM)," NASA Electronic Parts and Packaging (NEPP) Program, Washington, DC, USA, Tech. Rep. NEPP-TR-19-021, Jul. 2019.

[11] S. L. Katz, C. Heistand, and E. N. Miller, "NVIDIA Jetson TX2I TID and proton SEE testing: Results and a comparison of two proton beam facilities," in *Proc. IEEE Radiat. Effects Data Workshop* (in conjunction with 2023 NSREC), 2023, pp. 1–6.

[12] S. M. Guertin, "Raspberry Pis for space: Guideline," *Radiation Effects Group, Jet Propulsion Lab., California Inst. Technol.*, Pasadena, CA, USA, NASA Electronic Parts and Packaging (NEPP) Program, Tech. Rep. CL-21-5641, 2021. Accessed: Oct. 8, 2025. [Online]. Available: https://nepp.nasa.gov/docs/papers/2021-Guertin-Raspberry-Pi-Guideline-CL-21-5641.pdf

[13] M. Kolhof and E. G. Lightsey, "Design of a payload avionics interface board for the VISORS (virtual superoptics with reconfigurable swarms) CubeSat mission," Georgia Tech, Atlanta, GA, USA, Tech. Rep., Dec. 2021. Accessed: Oct. 8, 2025. [Online]. Available: https://repository.gatech.edu/entities/publication/607ce325-39d4-47cd-bc1f-bece5ecf75f5/full

[14] C. E. Gonzalez, C. J. Rojas, A. Bergel, and M. A. Diaz, "An architecture-tracking approach to evaluate a modular and extensible flight software for CubeSat nanosatellites," *IEEE Access*, vol. 7, pp. 126409–126429, 2019.

[15] A. Johnston et al., "Cosmic radiation environment data sheets," *J. Spacecraft Rockets*, vol. 50, pp. 999–1010, 2013.

[16] W. Weber, "Radiation effects in advanced microelectronics technologies," *Microelectronics Rel.*, vol. 58, pp. 1–10, 2016.

[17] B. Kim and H. Yang, "Temperature sensor assisted lifetime enhancement of satellite embedded systems via multi-core task mapping and DVFs," *Sensors (Switzerland)*, vol. 19, no. 22, 2019, Art. no. 4902.

[18] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *BMJ*, vol. 339, 2009, Art. no. b2535.

[19] D. Siewiorek and R. Swarz, *Reliable Computer Systems: Design and Evaluation*. Natick, MA, USA: AK Peters, 2002.

[20] A. Coronetti et al., "Mixed-field radiation qualification of a COTS space on-board computer along with its CMOS camera payload," in *Proc. 19th Eur. Conf. Radiat. Its Effects Compon. Syst.*, 2019, pp. 1–8.

[21] M. Henkel et al., "Mitigating and recovering from radiation induced faults in non-hardened spacecraft flash memory," in *Proc. IEEE Aerosp. Conf. Proc.*, 2024, pp. 1–8.

[22] B. Shashidhara, S. Jadhav, and Y. S. Kim, "Reconfigurable fault tolerant processor on a SRAM based FPGA," in *Proc. IEEE Int. Conf. Electro Inf. Technol.*, 2020, pp. 151–154.
[23] H. Su, T. Lu, and C. Feng, "Evaluation of modified collaborative triple module redundancy based on cots heterogeneous processor," in *Proc. Int. Conf. Comput. Inf. Big Data Appl.*, 2020, pp. 606–610.
[24] C. M. Fuchs et al., "A fault-tolerant MPSoC for CubeSats," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, 2019, pp. 1–6.
[25] V. Kumar, V. Pandey, K. Arun Kumar, and S. Sandya, "Redundancy management of on-board computer in nanosatellites," in *Proc. Adv. Intell. Syst. Comput.*, 2019, pp. 565–570.
[26] A. Martıń-Ortega et al., "Early SEU sensitivity assessment for collaborative hardening techniques: A case study of OPTOS processing architecture," *Microelectronics Rel.*, vol. 95, pp. 36–47, 2019.
[27] P. R. C. Villa, R. Travessini, R. C. Goerl, F. L. Vargas, and E. A. Bezerra, "Fault tolerant soft-core processor architecture based on temporal redundancy," *J. Electron. Testing: Theory Appl.*, vol. 35, no. 1, pp. 9–27, 2019.
[28] I. C. Lopes, V. Pouget, F. Wrobel, F. Saigne, A. Touboul, and K. Roed, "Development and evaluation of a flexible instrumentation layer for system-level testing of radiation effects," in *Proc. 21st IEEE Latin-Amer. Test Symp.*, 2020, pp. 1–6.
[29] European Space Agency, "5G nanosatellite project," *ESA Connectivity and Secure Communications Directorate*, Activity Code 3E.022, 2024. Accessed: Oct. 8, 2025. [Online]. Available: https://connectivity.esa.int/projects/5g-nanosatellite
[30] B.-Y. Choi, D. Boyd, and D. Wilkerson, "Toward reliable and energy efficient wireless sensing for space and extreme environments," in *Proc. IEEE Int. Conf. Wireless Space Extreme Environ.*, 2017, pp. 156–157.
[31] Q. Li, S. Wang, X. Ma, A. Zhou, and F. Yang, "Towards sustainable satellite edge computing," in *Proc. IEEE Int. Conf. Edge Comput.*, 2021, pp. 1–8.
[32] A. R. Harfian, B. Satriyotomo, and D. Arseno, "An investigation of RTOS-based sensor data management performance for tel-USAT on board data handling (OBDH) subsystem," in *Proc. Int. Conf. Inf. Commun. Technol.*, 2019, pp. 634–638.
[33] A. Prasad, Y. Jain, N. Joshi, N. Gupta, V. Singhania, and Y. Sreedharan, "Interfacing architecture between telemetry and on-board computer for a nanosatellite," in *Proc. IEEE Aerosp. Conf.*, 2020, pp. 1–6.
[34] S. L. Johansson, H. O. Said, H. Forsberg, N. Tsog, and O. Flordal, "Comparing EXT4 and ZFS for onboard data processing: A systematic mapping and experimental evaluation," in *Proc. 2023 Eur. Data Handling Data Process. Conf.*, 2023, pp. 1–8.
[35] D. D. Niz et al., "Mixed-trust computing for real-time systems," in *Proc. IEEE 25th Int. Conf. Embedded Real-Time Comput. Syst. Appl.*, 2019, pp. 1–25.
[36] L. A. Aranda, P. Reviriego, and J. A. Maestro, "Toward a fault-tolerant star tracker for small satellite applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3421–3431, Oct. 2020.
[37] A. Scholz, F. König, S. Fröhlich, and J. Piepenbrock, "Flight results of the compass-1 picosatellite mission," *Acta Astronautica*, vol. 66, pp. 1195–1201, 2010.
[38] L. Wood, W. Eddy, C. Smith, W. Ivancic, and C. Jackson, "Saratoga: A scalable data transfer protocol," *Internet-Draft draft-wood-tsvwg-saratoga-22, Internet Engineering Task Force*, Dec. 2017. Accessed: Oct. 8, 2025. [Online]. Available: https://datatracker.ietf.org/doc/draft-wood-tsvwg-saratoga/22/
[39] Q. Sun and H. Li, "Research and application of a UDP-based reliable data transfer protocol in wireless data transmission," in *Proc. 2011 Int. Conf. Comput. Sci. Serv. Syst.*, 2011, pp. 1514–1516.
[40] D. J. F. Miranda, M. Ferreira, F. Kucinskis, and D. McComas, "A comparative survey on flight software frameworks for 'new space' nanosatellite missions," *J. Aerosp. Technol. Manage.*, vol. 11, 2019, Art. no. e4619.
[41] A. Poghosyan and A. Golkar, "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions," *Prog. Aerosp. Sci.*, vol. 88, pp. 59–83, 2017.
[42] A. D. George and C. M. Wilson, "Onboard processing with hybrid and reconfigurable computing on small satellites," *Proc. IEEE*, vol. 106, no. 3, pp. 458–470, 2018.
[43] G. Furano and A. Menicucci, "Roadmap for On-Board Processing and Data Handling Systems in Space," in *Dependable Multicore Architectures at Nanoscale*. Berlin, Germany: Springer, 2018, pp. 253–281.
[44] G. Gao, L. Yao, W. Li, L. Zhang, and M. Zhang, "Onboard information fusion for multisatellite collaborative observation: Summary, challenges, and perspectives," *IEEE Geosci. Remote Sens. Mag.*, vol. 11, no. 2, pp. 40–59, Jun. 2023.
[45] O. Lux et al., "CCD detector performance of the space-borne Doppler wind Lidar ALADIN during the aeolus mission," *Appl. Opt.*, vol. 63, pp. 6754–6775, Sep. 2024.
[46] X. Wei et al., "A robust adaptive error state Kalman filter for MEMS IMU attitude estimation under dynamic acceleration," *Measurement*, vol. 242, 2025, Art. no. 116097.
[47] ECSS Secretariat, "Space engineering - thermal analysis handbook," *ECSS-E-HB-31-03A, European Space Agency Requirements & Standards Division, ESTEC, Noordwijk*, The Netherlands, Nov. 2016. Accessed: Oct. 8, 2025. [Online]. Available: https://www.ecss.nl/wp-content/uploads/2016/11/ECSS-E-HB-31-03A15November2016.pdf
[48] B. Zhang et al., "Progress and challenges in intelligent remote sensing satellite systems," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1814–1822, Feb. 2022, doi: 10.1109/JSTARS.2022.3148139.
[49] Q. T. Ngo, K. T. Phan, W. Xiang, A. Mahmood, and J. Slay, "On edge caching in satellite – IoT networks," *IEEE Internet Things Mag.*, vol. 4, no. 4, pp. 107–112, Dec. 2021.
[50] S. Xi, B. Shang, H. Zhang, J. Ma, and P. Fan, "Energy optimization in multisatellite-enabled edge computing systems," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 21715–21726, Jun. 2024.
[51] Pengutronix, "RAUC: Safe and secure OTA updates for embedded linux." Accessed: Oct. 8, 2025. [Online]. Available: https://rauc.io/
[52] A.K. Srivastava, K. CS, D. Lilaramani, R. R, and K. Sree, "An open-source SWUpdate and Hawkbit framework for OTA updates of RISC-V based resource constrained devices," in *Proc. 2nd Int. Conf. Commun., Comput. Ind. 4.0 (C2I4)*, 2021, pp. 1–6.

JOÃO ARAÚJO received the B.Sc. degree in informatics and computing engineering in 2023 from the University of Porto, Porto, Portugal, where he is currently working toward the master's degree in informatics and computing engineering.

He is currently a Researcher of software-defined architectures for embedded systems for space with Altice Labs, Aveiro, Portugal.

NUNO FILIPE MONTEIRO received the B.Sc. degree in electrical engineering from the Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal, in 2001, and the M.Sc. degree in electrical engineering from Columbia University, New York, NY, USA, in 2008.

He is currently a Senior Consultant with Altice Labs, Aveiro, Portugal.

RUI PINTO (Member, IEEE) received the master's and Ph.D. degrees in electrical and computer engineering from the Faculty of Engineering of the University of Porto (FEUP), Porto, Portugal, in 2013 and 2022, respectively.

He is a Doctoral Researcher with FEUP and an Integrated Member of the Research Center for Systems & Technologies. He is also an Invited Teaching Assistant with FEUP in the Informatics and Computing Engineering program. He has contributed to 20 national and European R&D projects and has authored 50 scientific publications. He has supervised or co-supervised more than 30 bachelor's, master's, and Ph.D. dissertations in topics related to the smart industry paradigm. His research focuses on the digitalization of industrial processes, WSAN, edge/cloud computing, smart components, bio-inspired approaches, and Education 4.0.