AegisSat: A Satellite Cybersecurity Testbed

Roee Idan[1,*], Roy Peled[1,*],Aviel Ben Siman Tov[1], Eli Markus[1], Boris Zadov[1], Ofir Chodeda[1], Yohai Fadida[1], Oliver Holschke[2], Jan Plachy[2], Asaf Shabtai[1], Yuval Elovici[1]

[1]Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Israel
[2]T-Labs (Research & Innovation)
[*]These authors contributed equally.

***Abstract*—The rapid increase in satellite deployment, and particularly nanosatellite deployment, has heightened their exposure to cybersecurity threats, making the task of safeguarding sensitive operations and data challenging. and making the task of safeguarding sensitive operations and data increasingly challenging. To address these challenges, we developed AegisSat, an open-source satellite cybersecurity testbed to study satellite resilience to cyberattacks and test dedicated detection and defense mechanisms, including machine learning-based solutions. Our testbed includes a physical CubeSat (Earth-based) and an environment emulator that mimics realistic orbital conditions such as sunlight, and magnetic fields. We also created a comprehensive dataset consisting of telemetry data and labeled attack data from experiments conducted using different scenarios. The data was collected during hundreds of experiments we performed in the testbed. By making both the design of the testbed and the dataset accessible to the research community, this work advances understanding of satellites' vulnerability to cyberattacks, drives the development of robust cybersecurity defenses, and establishes a platform for future research.**

I. Introduction

In the last decade, the use of satellite technology has evolved from being employed in exclusive applications, primarily by governments and large corporations, to widespread adoption by private companies and individuals, and it has become a critical component in global infrastructure supporting communication, navigation, Earth observation, and scientific research [1]. This transformation has been fueled by the NewSpace sector, driving advancements in reusable launch vehicles and streamlined satellite manufacturing; such advancements have reduced launch costs and expanded access to space, driving a surge in satellite deployment. The number of operational satellites has increased from a few dozen in 2012 to several thousand by 2023 [2]. Most of these satellites are deployed in low Earth orbit (LEO), defined as up to 2,000 km above Earth, and offer reduced latency and support applications in various domains, including Internet of Things (IoT), urban infrastructure management, and disaster response [3], [4]. There is a growing trend of launching CubeSats and nanosatellites (1–10 kg) into this orbit, as their small size and low weight allow them to be launched more cost-effectively and frequently than larger satellites.

Given their compact size and reduced resource requirements, CubeSats and nanosatellites are well-suited for deployment in LEO orbit. However, their small size and limited weight significantly restrict onboard resources. With minimal power generation capabilities, reduced battery capacity, and restricted space for processing components, these small satellites must optimize size, weight, and performance, making efficient design a necessity [5], [6]. The constrained resources and design limitations of CubeSats and nanosatellites make them particularly vulnerable to cyber threats. Despite these threats, nanosatellites have been widely adopted for scientific and commercial missions due to their versatility and lower costs.

The limited computational resources of these satellites make the implementation of strong security measures difficult, and their reliance on exposed communication channels increases the risk of attacks like jamming, eavesdropping, and spoofing [3], [7]. In addition, in many cases, the software deployed on these satellites originates from open-source repositories that may include one-day and zero-day vulnerabilities [8], [9]. These limitations and vulnerabilities, combined with factors such as the lack of cybersecurity regulations in the satellite industry, have left many small satellites exposed to potential threats [5], [10].

Recent reports have highlighted satellites' vulnerability to various cyberattacks, which can disrupt global communication, navigation, and data services [6], [10]–[12]. One notable incident occurred in February 2022, during the Russian-Ukrainian conflict, when a cyberattack targeted Viasat's KA-SAT satellite network using the "AcidRain" wiper malware. This attack disrupted broadband satellite Internet services across Ukraine and Europe, disabling tens of thousands of modems and leaving some users without connectivity for over two weeks. The impact extended beyond Internet outages, disrupting operations in critical sectors, including the energy section, where it affected a German company's ability to remotely monitor 5,800 wind turbines [13], [14]. This event underscored the vulnerabilities of satellite networks in geopolitical conflicts and highlighted the urgent need for robust cybersecurity measures to protect critical infrastructure.

Historically, the satellite industry has relied on an approach of *security through obscurity*, assuming that the proprietary nature and isolated operation of satellite systems would protect them from cyberattacks [6], [11]. This approach has limited

Workshop on Security of Space and Satellite Systems (SpaceSec) 2025
24 February 2025, San Diego, CA, USA
ISBN 979-8-9919276-1-1
https://dx.doi.org/10.14722/spacesec.2025.23069
www.ndss-symposium.org

researchers' ability to develop systems aimed at improving satellite cybersecurity, and the satellite industry's secretive practices have prevented researchers from assessing satellite vulnerabilities. The security through obscurity approach has also hindered the development of public and non-proprietary datasets on satellite cyber incidents, impeding research aimed at comprehensively analyzing and addressing cybersecurity issues and improving space systems' security [13], [14].

While various testbeds have been proposed to advance research in this area [15], [16], most existing testbeds primarily focus on simulating satellite attacks, with limited attention given to analyzing the consequences of successful attacks. Furthermore, these testbeds often lack the physical components necessary to emulate realistic satellite behavior under operational conditions and are generally not fully open source, particularly in aspects related to hardware and physical implementation.

To address these challenges, we present AegisSat, a novel satellite cybersecurity testbed that features a controlled emulator environment, a physical CubeSat model, and physical components, such as a magnetic field generator and solar lighting emulator, to emulate realistic space conditions. The testbed, which is presented in Fig. 1, provides a controlled environment for researchers to simulate satellite operations and launch cyberattacks at specified intervals, enabling the analysis of various threats and their potential impact.

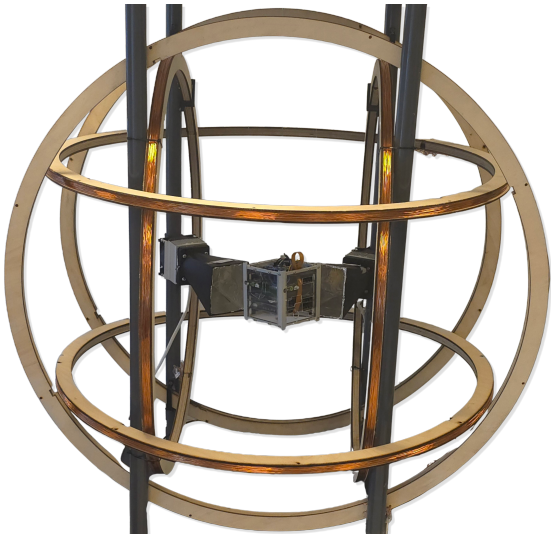

Fig. 1. The AegisSat testbed.

The testbed consists of five primary components: the *satellite*, *emulation environment*, *operations*, *attack manager*, and *simulation manager* components. The testbed's architecture, which includes both physical elements and simulation-based systems, is presented in Fig. 2.

The physical components, such as the CubeSat, magnetic field generator, and solar lighting emulator, replicate real-world satellite environments. The simulation-based components model dynamic orbital mechanics, environmental conditions, and cyberattack scenarios. Collectively, these testbed components enable realistic simulation of satellite functionalities, including environmental factors, operational scenarios, attack simulations, and coordinated management of all subsystems, creating a robust platform for assessing satellite operations and examining satellites' vulnerabilities to cyber threats.

The data collected by performing experiments in the testbed includes telemetry data from the satellite, encompassing information about its operations, components, and surrounding environments, as well as attack scenarios and the corresponding ground truth values. This data can be used in research for the development of robust mechanisms for satellite security, including attack detection and defense strategies.

Leveraging the capabilities of the AegisSat testbed, we also collected a dataset based on the hundreds of experiments we conducted in the testbed. To demonstrate the dataset's potential, we trained a machine learning model on a subset of the data, obtaining promising results that highlight the dataset's ability to advance the security of space systems and evaluate satellite defense mechanisms.

By making the testbed's design and dataset accessible to the research community, this work enables researchers to examine the impact of cyberattacks in different scenarios, develop and validate advanced cybersecurity solutions, and evaluate satellite defense mechanisms, serving as a foundation for future research in the satellite cybersecurity domain and fostering collaboration in this critical area[1].

II. Related Work

In response to the emerging threats, projects like the SPARTA project by the Aerospace Corporation [17] and ESA's SpaceShield program [18] were initiated to develop comprehensive satellite cybersecurity frameworks and standardized security practices. The growing number of initiatives highlights the need to create a testbed to evaluate and enhance such satellite security frameworks.

In light of these developments, several organizations have developed satellite cybersecurity testbeds to evaluate and enhance the resilience of space systems. The MITRE Corporation's satellite testbed [15] focuses on developing advanced cybersecurity measures for commercial satellites. It uses non-space-rated proprietary and commercial off-the-shelf (COTS) components to assess vulnerabilities and test mitigation strategies, particularly through external attack simulations that provide valuable insights into satellite weaknesses. However, it lacks realistic environmental emulation capabilities, such as magnetic fields or an Earth view. It treats the satellite as a black box, emphasizing external attack vectors rather than internal defense mechanisms.

The Merge/Space testbed [16], developed by USC-ISI, simulates multi-agent security scenarios using virtual satellite networks. It explores cyberattacks such as denial-of-service (DoS), network scanning, and data exfiltration attacks, in a controlled environment. This testbed focuses on the simulation

[1]The open-source testbed and dataset are available at https://github.com/texydo/satellite_security_testbed

of attacks on satellite constellations by modeling network vulnerabilities and testing potential exploits, providing valuable insights into the unique security challenges interconnected satellite systems face. However, it does not explicitly address the simulation of physical aspects of satellite operations, such as power constraints and environmental interactions, which is crucial for understanding the full scope of potential cyber threats in space.

The QPEP project [19] introduced a novel quantum-resistant communication protocol designed to secure data transmissions between satellites and ground stations. This testbed focuses on evaluating advanced cryptographic techniques but does not provide a physical satellite platform for testing.

The Dominant Systems Corporation developed a satellite lab platform emphasizing practical cybersecurity training and solution testing [20]. However, it operates within a proprietary environment and is not an open-source platform. ESA's OPS-SAT [21] features an experimental CubeSat for testing new technologies, including cybersecurity protocols, in an actual space environment. While OPS-SAT includes some publicly available datasets, they are limited in scope and contain partial telemetry data.

While existing testbeds have advanced satellite cybersecurity research, they often lack the ability to perform realistic environment emulation and the ability to run onboard machine learning (ML) models and do not provide access to datasets designed for research purposes. Additionally, most testbeds do not provide open-source code or data.

The key features and capabilities of state-of-the-art satellite cybersecurity testbeds include a **physical satellite**, which enables hardware-based testing under real-world conditions, and a **simulation environment**, which provides controlled setups for modeling operations and threats. **Realistic environment emulation** replicates orbital conditions like magnetic fields, solar radiation, and power constraints to assess satellite resilience. The ability to simulate **cyberattacks** enables the evaluation of vulnerabilities in scenarios involving DoS attacks, unauthorized access, and other threats. **Shared datasets**, generated during simulations, support collaboration and benchmarking, complementing the accessibility offered by **open-source platforms**.

The simulation of **malware Simulation** enables the analysis of the impact of malicious code on satellite functionality and the effects of unauthorized or altered commands on system operations. **Telemetry data collection** offers real-time monitoring of system performance, while **power constraint simulation** examines the effects of energy limitations. Finally, the ability to **run onboard ML/DL (deep learning) models** facilitates the development of AI-driven security strategies.

Table I compares our testbed and existing similar satellite security testbeds, in terms of the key features and capabilities essential for state-of-the-art research. As can be seen, our testbed includes several key features, such as a physical satellite platform, the ability to run onboard satellite ML models, realistic environmental simulations that accurately reproduce orbital conditions (e.g., magnetic fields, sunlight exposure, and power constraints), and an accessible dataset. Furthermore, our testbed offers open-source code, fostering collaboration and enabling researchers to benchmark and develop innovative cybersecurity strategies.

TABLE I
COMPARISON OF SATELLITE CYBERSECURITY TESTBEDS.

Feature/Capability	MITRE Testbed [15]	Merge/Space Testbed [16]	**AegisSat Testbed**
Physical Satellite	✓		✓
Simulation Environment	✓	✓	✓
Realistic Environment Emulation			✓
Cyberattack Simulation	✓	✓	✓
Open Source		✓	✓
Telemetry Data Collection	✓	✓	✓
Shared Datasets			✓
Malware Simulation	✓	✓	✓
Power Constraint Simulation			✓
Running Onboard ML/DL Models			✓
Target Audience	Industry-focused	Academic-focused	Academic-focused

III. TESTBED ARCHITECTURE

AegisSat's architecture is designed to provide a realistic and controlled environment for the simulation of satellite operations and cyberattack scenarios. The testbed is comprised of five main components: the satellite, emulation environment, operations, attack manager, and simulation manager components, as shown in Fig. 2. This section provides an overview of each component and their interconnections.

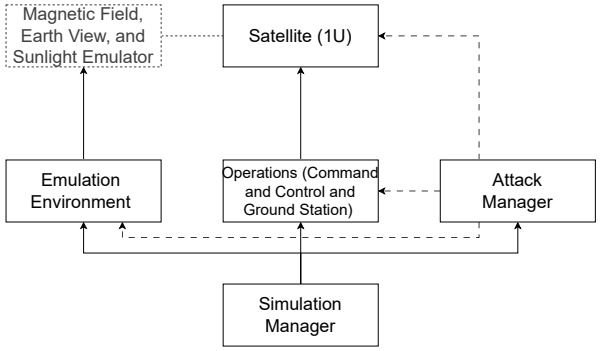

Fig. 2. Testbed architecture. The boxes bordered in a solid line represent core components, while the box outlined in a dashed line is part of the emulation environment and influences the satellite without a physical link. The dashed arrows from the attack manager component point to components that can be targeted during cyberattacks.

A. Satellite

At the core of the testbed is a physical 1U nanosatellite integrated with various subsystems to emulate real satellite operations. The onboard computer (OBC) subsystem handles command execution and data processing, while the electrical power system (EPS) manages the power generation by the solar panels and a battery for storage. This system ensures the satellite's operation during conditions of limited sunlight and distributes power to the satellite's subsystems. The attitude determination and control system (ADCS) stabilizes the satellite and controls its orientation using sensors, actuator data, and a magnetometer for orientation based on the magnetic field. A basic thermal control system (TCS) is included to maintain

optimal temperatures. The communication system, equipped with radio and Wi-Fi capabilities, facilitates data exchange with the ground station. Additionally, the satellite features a payload with imaging capabilities and enhanced computing power for processing tasks.

B. Emulation Environment

The emulation environment component consists of modules that replicate some of the conditions a satellite would encounter in space to create a realistic testing and simulation setup. It includes modules that emulate the Earth's magnetic field, sunlight exposure, and the Earth view at every point in time along the satellite's trajectory. The magnetic field module simulates the space environment, supports the examination of the satellite's attitude control systems, and enables the use of the telemetry data it generates. The sunlight module adjusts for varying angles and intensities of sunlight, assessing solar panel efficiency and thermal management. The Earth view module provides visual simulation to validate payload operations, enabling imaging and data collection. These emulations collectively mimic orbital dynamics, and specifically the satellite's movement and interactions, to test its responses in space.

C. Operations Component

The operations component serves as the testbed's command and control center, interfacing with the satellite through a ground station. It manages the flow of commands, telemetry, and data, ensuring interaction between the satellite and ground systems. The ground station communicates with the satellite using radio signals or Wi-Fi, enabling remote monitoring and control of missions and effective data exchange. Data transmission is an integral part of nominal operations, modeled in the simulation to replicate both uplink and downlink processes. The data transmission handles data packets and potential transmission errors, ensuring the standard flow of communication between the satellite and the ground station.

Nominal operations refer to the satellite's regular functioning without intentional disruptions, including the routine execution of commands sent from the ground station. The process replicates real-world communication protocols, focusing on command validation and precise execution timing. The nominal operations simulation also incorporates regular maintenance activities, such as software updates and system health checks, to ensure that the satellite's systems remain functional and perform as expected.

This component also includes scenario simulation capabilities, generating predefined or random scenarios to simulate satellite operations based on external data sources. Standard operational procedures, such as runtime limits to prevent hardware wear and overheating and command limits to avoid system overloads, are implemented to ensure safe and efficient operations. Rest periods are integrated to allow the satellite to cool down and recharge, simulating natural downtime during missions.

D. Attack Manager

The attack manager is responsible for preparing and executing cyberattacks within the testbed environment. Attacks are initialized and executed from different points in the simulation environment depending on the specific attack scenario. The types of attacks simulated are malware insertion, data poisoning, command injection, and jamming attacks. As an example, one attack scenario involves a rogue satellite operator uploading anomalous commands to the satellite. Further details on the different attack scenarios and their implementation are provided in Section IV.

E. Simulation Manager

The simulation manager oversees all testbed activities and ensures that all the components remain synchronized. It initiates the testbed and sets the parameters and conditions for the simulation scenarios, configuring its standard operations and mission-specific contexts, as well as coordinating with the attack manager to initialize the attacks planned for the scenario. During the scenario's execution, the simulation manager continuously updates the satellite's position and the simulation time, ensuring that all components receive the precise data required for their roles, delivered consistently and in sync. Moreover, it actively monitors and logs the scenario in real time, recording comprehensive data throughout the scenario's execution.

The simulation manager includes a web application (presented in Fig. 3) that incorporates live data visualization. This includes visual graphs of telemetry data, command history, real-time monitoring of the main components, and orbit tracking. Such visualization helps in monitoring the execution of the scenario.

Fig. 3. The simulation manager component's web application, which incorporates live data visualization, including graphs of telemetry data and command logs, and facilitates real-time monitoring of the primary components and the scenario's execution.

The modular and flexible design of the satellite cybersecurity testbed allows for easy integration of new components and adaptations to meet evolving research requirements and advancements in satellite design and technology. For example, the testbed can accommodate the addition of new payloads, simulate emerging attack vectors, and incorporate advanced

defense mechanisms as they are developed. This adaptability ensures that the testbed will remain a valuable platform for investigating and developing cybersecurity solutions for satellite systems in the years to come.

IV. Cyberattacks

The satellite testbed's cyberattack component is designed to simulate a wide range of cyber threats and evaluate the satellite's resilience in the face of these threats. This section provides detailed information on the different types of attacks that can be simulated, the mechanisms used to execute the attacks, and the scenarios they aim to replicate.

A. Malware Insertion

Malware insertion involves injecting malicious software into the satellite's systems. The malware can execute specific actions, such as turning on/off different subsystems, altering telemetry data, or disrupting communication, and it can be triggered by specific events or command patterns. For example, in a nominal operation scenario, the malware could be triggered during the transition from day to night, causing the payload to capture an excessive number of images and deplete the satellite's battery.

The following steps are performed when simulating a malware insertion attack in the testbed:

- **Preparation:** Configuring the malware to execute predefined actions based on specific triggers.
- **Loading:** Integrating the malware into the satellite's systems through the testbed environment.
- **Execution:** Triggering the malware based on defined scenarios, such as when a satellite communicates with the ground station or reaches a specific position.
- **Monitoring and Logging:** Observing the satellite's response and logging relevant data for analysis.

B. Data Poisoning

Data poisoning aims to compromise the integrity of the data being processed or transmitted by the satellite. Telemetry data can be altered to include false information, or images captured during an Earth observation mission can be manipulated. For instance, if a satellite loses control and starts spinning, false information could be injected into the ADCS data, falsely indicating that the satellite is stable and operating nominally.

The following steps are performed when simulating a data poisoning attack in the testbed:

- **Preparation:** Setting up parameters for altering telemetry data or images.
- **Loading:** Injecting the malicious data into the satellite's data streams.
- **Execution:** Implementing data poisoning during critical data transmissions or imaging operations.
- **Monitoring and Logging:** Tracking the changes in the data and logging the satellite's response.

C. Command Injection

Command injection involves manipulating or disrupting the command sequences sent to the satellite. This can include injecting false commands, delaying or blocking legitimate commands, or altering the command history to disrupt normal operations. For example, a malicious operator could briefly activate the ADCS, causing the satellite to deviate from its desired orbit.

The following steps are performed when simulating a command injecting attack in the testbed:

- **Preparation:** Defining the specific command manipulation to be performed.
- **Loading:** Uploading the manipulated command to the satellite.
- **Execution:** Executing the unintended commands during orbit operations.
- **Monitoring and Logging:** Observing the impact of the unintended commands on satellite operations and recording the resulting disruptions.

D. Jamming Attacks

Jamming attacks aim to disrupt the communication link between the ground station and the satellite. This can block or interfere with commands being uploaded to the satellite or telemetry being downloaded from it. For example, jamming can prevent commands from reaching the satellite, causing it to miss necessary instructions and fail to capture the desired images.

The following steps are performed when simulating a jamming attack in the testbed:

- **Preparation:** Configuring the frequency and timing of the jamming signals.
- **Execution:** Initiating the jamming during critical communication windows.
- **Monitoring and Logging:** Observing the effects on satellite communication and control, and logging the results of the operation.

V. AegisSat Design and Operational Flow

The testbed's design and operational flow involve several integral elements designed to simulate and monitor satellite system operations and manage the cyberattack. These elements are organized into five main components, as illustrated in Fig. 2, which outlines the logical structure of the testbed. The elements within these components are depicted in Fig. 4, while the physical testbed setup is shown in Fig. 1, where it is arranged on a custom-designed table that houses all of the components.

Table II provides a brief description of each element.

The design of our satellite cybersecurity testbed is presented in Fig. 4 We present an overview of how the testbed's elements interact and function together to simulate and evaluate satellite operations and cyberattacks. The simulation manager orchestrates the entire operation of the testbed, coordinating all components. The simulation manager provides the satellite's orbital data to the emulation computer, which uses it to

TABLE II
DESCRIPTION OF TESTBED ELEMENTS AND FUNCTIONS.

Elements	Description
Satellite	For the satellite, we use EAST [22], an educational nanosatellite designed to demonstrate and validate key satellite subsystems. It includes essential systems such as EPS, data handling, TT&C, and ADCS, along with interfaces for user-developed software and hardware modules. For detailed information, refer to the appendix.
Raspberry Pi Zero 2W	The Raspberry Pi Zero 2W enhance the satellite's capabilities by serving as the CPU. It enables system enhancements and simulates attacks and defenses using ML/DL.
V-I Sampler	The V-I sampler monitors voltage (V) and current (I) in electronic systems while also functioning as a power supply.
Magnetic Field Emulation	The magnetic field emulation system is a custom-built 3-axis Helmholtz coil system that emulates the Earth's magnetic field. It generates controlled, uniform magnetic fields in three orthogonal directions, enabling precise validation of satellite systems.
Sunlight Emulation Lamps	The sunlight emulation lamps consist of two custom-built 100W LED lamps used to emulate sunlight conditions. Each lamp emits 9000 lumens, simulating the satellite's exposure to solar radiation.
Earth View Simulation	The earth view simulation is a screen-based system that emulates the view of Earth as seen by the satellite. It provides a realistic visual environment for payloads like the camera.
Emulation Computer	The emulation computer manages the emulation environment by coordinating and synchronizing magnetic fields, the Earth view, and sunlight emulation, ensuring precise environmental simulation.
Ground Station (GS)	The GS facilitate satellite communication via radio frequencies and hash transmission (TX) and reception (RX) capabilities.
Command and Control (C&C) Center	The C&C center operates from a dedicated computer that manages the satellite's C&C operations. It ensures effective mission management using COSMOS version 4 [23].
Camera Payload	The camera payload, connected to the Raspberry Pi Zero 2W, captures the emulated Earth view. It facilitates the evaluation of image capturing and processing as the satellite experiences simulated orbital perspectives.
Attack Manager	The attack manager is a specialized computer that orchestrates and manages cyberattacks targeting the satellite system.
Simulation Manager	The simulator manager runs simulation scenarios and manages the overall simulation environment. It ensures control of Attack Manager and C&C Center interactions.

generate the magnetic field, sunlight, and Earth view emulation, ensuring these conditions accurately mimic the space environment encountered by satellites in orbit.

The emulation computer provides power data to the V-I sampler, which supplies the power needed for the magnetic field and sunlight emulation systems. The V-I sampler is directly connected to the satellite through an out-of-band channel, enhancing the security and reliability of power monitoring.

The Raspberry Pi Zero 2W acts as the satellite's central processing unit (CPU) and is connected to the ESAT using UART, a serial communication protocol that facilitates reliable data exchange between devices. This integration allows for detailed analysis using ML and DL models and testing of the satellite's functionalities and additional payloads, such as the camera and communication payload with the C&C center through Wi-Fi.

The C&C center, operated from a dedicated computer, manages the execution of command and control strategies to ensure effective mission management. The ground segment facilitates communication by maintaining a reliable link with the satellite through Wi-Fi and radio, while the attack manager orchestrates simulated cyberattacks on the ground segment and Raspberry Pi Zero 2W.

VI. DATASET COLLECTION

We used the testbed to create a comprehensive dataset, the main purpose of which is to support further research on satellites in the cybersecurity domain. In this section, we describe the data collection process and the dataset's structure and provide additional details on the data. We also present several potential cyberattack scenarios and demonstrate the application of our dataset's telemetry data in an anomaly detection use case.

A. Dataset Contents

The dataset collected for the evaluation presented in this paper comprises data from hundreds of simulations designed to emulate satellite operations in space. The simulations range from one lap, which is 90 minutes long, to a few hours, and approximately 5,000 records are generated per lap. The dataset consists of: (1) satellite telemetry data from onboard sensors, recorded at a frequency of one reading per second; (2) details on commands issued from the ground station; and (3) records corresponding to attack scenarios, which are explicitly labeled with the attack type and associated parameters.

B. Dataset Structure

Each sample contains general information, including the simulation ID, timestamp, satellite name, and telemetry data, with other fields available to store relevant details when operational commands are issued and attacks are simulated. The dataset is stored in a MongoDB database.

1) Telemetry Data: The telemetry data includes readings from sensors monitoring various components of the satellite. The components we collect data on are: (a) **TCS:** provides information on the TCS, including its operational mode and temperature; (b) **Payload:** captures data on the camera status, CPU usage, and RAM usage percentage; (c) **EPS:** captures readings from sensors in the satellite's EPS; and (d) **OBC:** contains data from the OBC, such as processor temperature and RAM utilization data.

2) Commands: This sample field is only used when a command is sent from the ground station to the satellite during the simulation. The values in this field represent the commands as strings.

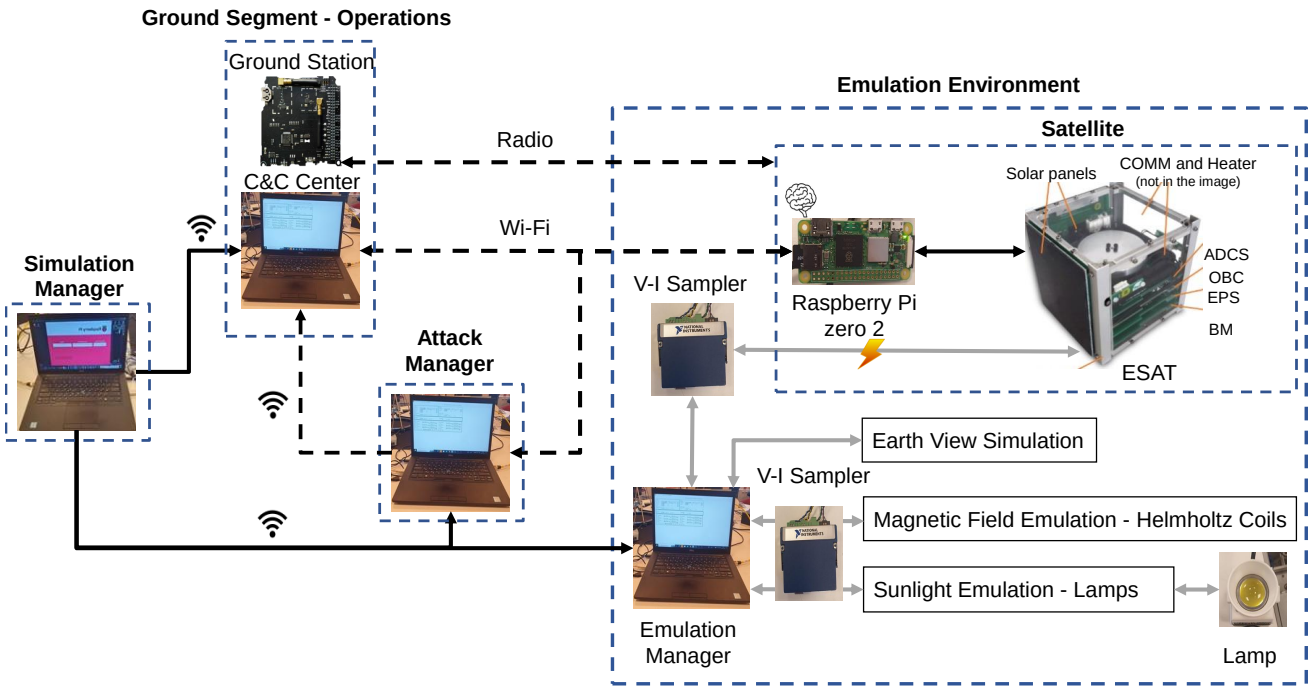

Fig. 4. Testbed's implementation.

3) Attack: When attacks are simulated in a given sample, a field describing the names of the attacks and relevant parameters is populated.

C. Cyberattack Scenarios in the Testbed

The testbed's design enables the simulation and analysis of a wide range of attack scenarios. Below, we describe several attack scenarios that can be executed in the AegisSat testbed.

1) Human-Triggered: An attacker with the ability to send operational commands to the satellite could execute a sequence of commands that activate pre-installed malware injected via a supply chain attack. This type of attack can be identified by detecting an abnormal sequence of commands or by analyzing telemetry data for abnormal CPU usage spikes immediately following the execution of these commands, deviating from typical CPU activity patterns.

2) Environment-Triggered: An attacker could deploy malware that is activated in response to environmental conditions, such as day/night cycles or variations in the Earth's magnetic field. The malware could deplete CPU resources, fill the satellite's limited memory with junk data, or manipulate files, creating, editing, corrupting, or deleting them.

3) Insider Threat: An attacker, either with direct access to the satellite or through a connection to a malicious satellite operator with the ability to send harmful operational commands from the ground station, could deliberately activate multiple components and payloads simultaneously and continuously. This could generate excessive heat, potentially disrupting the satellite's operation or causing damage to its electrical components.

D. Dataset's Use

To demonstrate how the dataset can be used, we trained two Variational Auto Encoders (VAE) with different architectures on a small portion of the telemetry data in our dataset to detect anomalies (attacks). The experiments show how data from even a few simulations can support the development and evaluation of satellite anomaly detection systems, emphasizing the potential benefits of using the complete dataset for training more advanced models.

1) Experimental Setup: The training dataset was generated using six satellite simulations based on the two-line element (TLE) file from the International Space Station (ISS), resulting in approximately 30,000 data records. We selected nine features from this dataset that monitor the satellite's electrical usage and OBC for detailed information refer to the appendix. The test dataset was created by performing an additional simulation of the same satellite while executing several attacks designed to cause excessive resource consumption.

We employed two unsupervised Variational Auto Encoder (VAE) models for anomaly detection: (1) a Light-VAE, which utilizes a recurrent architecture, and (2) a Heavy-VAE, which integrates convolutional, recurrent, and attention mechanisms for enhanced feature extraction. Both models were implemented using PyTorch [24]. The Light-VAE consists of a GRU-based encoder-decoder structure that captures temporal dependencies and reconstructs the input sequence using a compact latent representation. In contrast, the Heavy-VAE employs a CNN feature extractor to capture local temporal patterns, followed by a bidirectional LSTM encoder that models long-

TABLE III
PERFORMANCE COMPARISON OF LIGHTWEIGHTVAE AND HEAVYVAE MODELS

Model	AUC	Detection Rate (%)	False Alarm Rate (%)	Precision	Recall	F1-Score
Lightweight VAE	0.80	47.71	**4.53**	0.48	0.47	0.47
Heavy VAE	**0.92**	**82.11**	11.85	**0.59**	**0.82**	**0.69**

range dependencies. Additionally, a multihead self-attention mechanism refines the latent representations before decoding with an LSTM-based structure.

Hyperparameters for both models were determined through empirical experimentation, evaluating different configurations to optimize reconstruction accuracy while maintaining stable latent space representations. The models were trained using a loss function that combines Mean Squared Error (MSE) reconstruction loss with a KL divergence term, ensuring a balance between accurate reconstruction and effective latent space regularization.

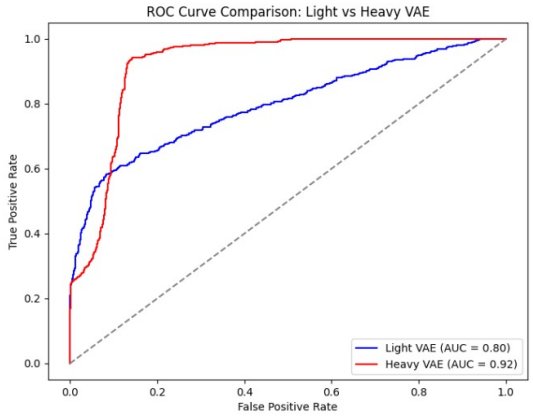

Fig. 5. ROC curve and AUC for the examined anomaly detection models.

2) Experimental Results: The results of our experiments are presented in Table III and Fig.5. The AUC values for the Light-VAE and Heavy-VAE models are 0.80 and 0.92, respectively, demonstrating their ability to detect abnormal patterns caused by cyberattacks. The Heavy-VAE achieves a significantly higher detection rate of 82.11%, compared to 47.71% for the Light-VAE, indicating its superior capability in identifying attacks. However, this comes at the cost of a higher false alarm rate (11.85%) compared to the LightweightVAE (4.53%), highlighting the trade-off between sensitivity and specificity.

The ROC curves in Fig. 5 illustrate this difference in performance, showing that the Heavy-VAE consistently achieves higher true positive rates across different false positive rate thresholds. Additionally, precision and F1-score values reinforce the trade-off observed, with the Heavy-VAE achieving an F1-score of 0.69, compared to 0.47 for the Light-VAE.

We used a subset of our dataset and hypothesized that increasing its diversity through additional simulations would enhance model robustness and anomaly detection. These findings underscore the value of our dataset and testbed telemetry analysis in cyberattack detection.

VII. CONCLUSION

The AegisSat testbed and the accompanying dataset presented in this paper provide a platform to study satellites' resilience to cyberattacks under realistic conditions in a lab environment. The testbed allows researchers to simulate satellite operations, test the impact of environmental factors, and diverse attack scenarios. The open-source dataset provides labeled telemetry and attack data to support further research, foster collaboration, and help new researchers entering the field. Together, the testbed and dataset will contribute to improved satellite cybersecurity.

AegisSat serves as both a blueprint and an open-source platform, fostering collaboration among the research community. The design, along with the supporting code, is freely accessible, enabling researchers to replicate, modify, and enhance the testbed in their own facilities. This approach eliminates the need for physical access to the original testbed, allowing researchers to build their own tailored versions using the provided blueprint and design, adapting it to meet their specific research needs. By using the open-source design as a foundation, researchers can enhance the testbed's features, ensuring that it remains a dynamic resource for satellite cybersecurity research.

The dataset provided with this paper includes extensive telemetry data and labeled attack data from numerous simulations and experiments conducted using different scenarios. It enables researchers to analyze the impact of cyberattacks and develop robust detection and defense mechanisms. It encourages researchers in fields like ML and cybersecurity to more easily perform studies in the emerging domain of satellite security and contribute to its advancement. In future research, we plan to expand the dataset, enhancing its value as a resource for satellite cybersecurity research.

Future work can focus on expanding the testbed's capabilities to address emerging challenges. This could include simulating more sophisticated multi-satellite networks to explore inter-satellite communication and coordinated operations, as well as enhancing the emulation environments to account for dynamic orbital mechanics and complex space phenomena such as South Atlantic Anomaly. In addition, more advanced attack models, such as zero-day exploits and coordinated multi-vector threats, could be integrated in the testbed to provide deeper insights into satellites' cybersecurity risks. Continued contributions from the research community will ensure that the testbed and dataset platform presented evolves to address the growing complexity of satellite systems and remains a relevant tool for improving satellites' resilience to cyber threats.

REFERENCES

[1] M. Mesich, "Satellite cybersecurity act of 2022 highlights growing importance of satellite networks in critical infrastructure," 2022, accessed: 2024-10-07. [Online]. Available: https://www.industrialdefender.com

[2] N. Database, "Nanosats database: The comprehensive database of nanosatellites," 2023, accessed: 2024-11-16. [Online]. Available: https://www.nanosats.eu/

[3] P. Yue, J. An, J. Zhang, J. Ye, G. Pan, S. Wang, P. Xiao, and L. Hanzo, "Low earth orbit satellite security and reliability: Issues, solutions, and the road ahead," *IEEE Communications Surveys & Tutorials*, 2023.

[4] M. Xie, W. Zhang, and L. Chen, "Key technologies and challenges for leo mega-constellations in 6g global coverage," *IEEE Wireless Communications*, 2023.

[5] M. Strohmeier, J. Pavur, and G. Tresoldi, "Space odyssey: An experimental software security analysis of satellites," *IEEE Symposium on Security and Privacy*, 2023.

[6] J. Pavur and I. Martinovic, "Building a launchpad for satellite cybersecurity research: lessons from 60 years of spaceflight," *Journal of Cybersecurity*, vol. 8, no. 1, p. tyac008, 2022.

[7] F. Guo, Y. Chen, and L. Zhang, "Security threats in space-air-ground integrated networks (sagins)," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[8] E. U. A. for Cybersecurity (ENISA), "Low earth orbit (leo) satcom cybersecurity assessment," 2023, accessed: 2024-12-06. [Online]. Available: https://www.enisa.europa.eu/publications/low-earth-orbit-leo-satcom-cybersecurity-assessment

[9] P. Hansen, W. C. Henry, M. G. Reith, R. Thummala, and G. Falco, "Guarding the galaxy: Satellite ransomware and countermeasures," in *2024 IEEE Aerospace Conference*. IEEE, 2024, pp. 1–6.

[10] M. Manulis, C. P. Bridges, R. Harrison, V. Sekar, and A. Davis, "Cyber security in new space: analysis of threats, key enabling technologies and challenges," *International Journal of Information Security*, vol. 20, pp. 287–311, 2021.

[11] R. Peled, E. Aizikovich, E. Habler, Y. Elovici, and A. Shabtai, "Evaluating the security of satellite systems," *arXiv preprint arXiv:2312.01330*, 2023.

[12] J. Willbold, M. Schloegel, M. Vögele, M. Gerhardt, T. Holz, and A. Abbasi, "Space odyssey: An experimental software security analysis of satellites," in *IEEE Symposium on Security and Privacy*, 2023.

[13] C. P. Institute, "Case study viasat," June 2022. [Online]. Available: https://cyberconflicts.cyberpeaceinstitute.org/law-and-policy/cases/viasat

[14] N. Boschetti, N. G. Gordon, and G. Falco, "Space cybersecurity lessons learned from the viasat cyberattack," in *Proceedings of ASCEND 2022*. American Institute of Aeronautics and Astronautics (AIAA), 2022, p. 4380.

[15] J. Finke, R. Thummala, R. Elbasheer, P. Hansen, W. Henry, D. Mamula, A. M. Noor, T. York, K. Zheng, and G. Falco, "Satellite cybersecurity testbed to improve commercial space security," in *Proceedings of ASCEND 2023*. American Institute of Aeronautics and Astronautics (AIAA), 2023, p. 4768.

[16] M. P. Collins, A. Hussain, J. Walters, C. Ardi, C. Tran, and S. Schwab, "Merge/space: A security testbed for satellite systems," in *SpaceSec*, 2024.

[17] T. A. Corporation, "Sparta," 2023, accessed: 2024-11-16. [Online]. Available: https://sparta.aerospace.org/

[18] E. S. Agency, "Spaceshield," 2023, accessed: 2024-11-16. [Online]. Available: https://spaceshield.esa.int/

[19] J. Huwyler, J. Pavur, G. Tresoldi, and M. Strohmeier, "Qpep in the real world: A testbed for secure satellite communication performance." in *SpaceSec*, 2023.

[20] D. S. Corporation, "Satellite cyber security," 2023, accessed: 2024-11-16. [Online]. Available: https://www.dominantisc.ca/satellite-cyber-security

[21] E. S. Agency, "Ops-sat," 2023, accessed: 2024-11-16. [Online]. Available: https://www.esa.int/Enabling_Support/Operations/OPS-SAT

[22] P. S. Sánchez, I. Tinao, J. Ezquerro, J. Fernández, J. Rodríguez, A. Bello, and K. Olfe, "Educational nanosatellites for hands-on learning in aerospace engineering education," in *INTED2021 Proceedings*. IATED, 2021, pp. 2587–2596.

[23] R. Melton, "Ball aerospace cosmos open source command and control system," 2016.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

APPENDIX

ESAT's Subsystem

1) **Electrical Power Subsystem (EPS):** The EPS is responsible for managing power generation, storage, and distribution in ESAT.
   - **Solar Panels:** Two fixed panels, each providing **5.5V at 180mA**.
   - **Battery Pack:** Li-ion pack operating at **6.0–8.4V**.
   - **Power Outputs:** Provides **3.3V and 5V** regulated outputs.
   - **Battery Management:** Overvoltage, undervoltage, overcurrent, and overtemperature protections.
2) **On-Board Computer (OBC):** The OBC acts as the main processing unit of ESAT, responsible for handling telemetry, executing commands, and managing software.
   - **Microcontroller:** MSP430F5529 MCU.
   - **Memory:** 64KB Flash storage.
   - **Interfaces:** Supports I2C, SPI, and UART communication.
   - **WiFi Module:** Enables remote connectivity via 802.11 b/g/n.
   - **Real-Time Clock (RTC):** Maintains system time for telemetry synchronization.
3) **Communication Subsystem (COM):** Handles wireless data transmission and reception between ESAT and the ground station.
   - **Frequency Range:** Operates within **425–525 MHz** (default: 433 MHz, ISM band).
   - **Alternative Bands:** Supports operation in **142–175 MHz** and **above 850 MHz**, but communication is not guaranteed due to potential crosstalk effects.
   - **Modulation Types:** Supports multiple digital modulation schemes for uplink and downlink communication:
     - **OOK (On-Off Keying)**
     - **2FSK (Two-Frequency Shift Keying)**
     - **2GFSK (Two-Gaussian Frequency Shift Keying)**
     - **4FSK (Four-Frequency Shift Keying)**
     - **4GFSK (Four-Gaussian Frequency Shift Keying)**
     - **Carrier Mode (Continuous Wave)**
   - **Channel Selection:** Supports channels **0–31**, each spaced **250 kHz** apart.
   - **Power Control:** Transmission power is adjustable from **0% to 100%** (default: 100%).
4) **Attitude Determination and Control Subsystem (ADCS):** Manages satellite orientation using sensors and actuators.
   - **Sensors:** Includes a 3-axis gyroscope, 3-axis magnetometer, coarse sun sensors (CSS), and an IMU.

- **Magnetometer:**
  - Only X- and Y-axis readings are available in the default ADCS software. The software needs to be upgraded for the z-axis.
  - Provides magnetic azimuth determination based on the Earth's magnetic field.
  - Includes a geometry correction function that interpolates angles at 0, 45, 90, 135, 180, 225, 270, and 315 degrees for more accurate readings.
- **Actuators:** Uses magnetorquers and a reaction wheel to control orientation.
- **Reaction Wheel:**
  - Operates on the Z-axis for attitude stabilization.
  - Speed can be controlled in the range of 0 to 7000 RPM via PID control.
  - Uses a tachometer for precise speed measurements.
  - Requires a minimum start speed of 3000-4000 RPM for optimal operation.
- **Control Algorithm:** Implements a configurable PID controller.
- **Telemetry Data:** Outputs angular velocity, magnetic field measurements, sun sensor readings, and reaction wheel speed.

5) **Thermal Payload (TPL):** The ESAT is equipped with a thermal payload (TPL) for experimentation.
   - **Heating Element:** Capable of precise temperature modulation.
   - **Control:** Uses PWM modulation for accurate thermal adjustments.
   - **Power Supply:** Operates on a regulated 5V line.
6) **Ground Station (GS):** Facilitates telemetry reception and command transmission to ESAT.
   - **Radio Interface:** Utilizes the same transceiver hardware as the COM subsystem.
   - **Telemetry Link:** Supports full-duplex data exchange.
   - **Data Processing:** Connected to a microcontroller handling commands and telemetry storage.

Feature Descriptions

- **EPS_3V3_LINE_CURRENT**: Total 3.3V buses current.
- **EPS_3V3_LINE_VOLTAGE**: 3.3V buses voltage.
- **EPS_5V_LINE_CURRENT**: Total 5V buses current.
- **EPS_5V_LINE_VOLTAGE**: 5V buses voltage.
- **EPS_BATTERY_CURRENT**: Battery output current.
- **EPS_BATTERY_TOTAL_VOLTAGE**: Battery pack voltage.
- **EPS_BATTERY_TEMPERATURE**: Battery temperature.
- **OBC_PROCESSOR_LOAD**: Processor running time after last reset.
- **OBC_PROCESSOR_TEMPERATURE**: Processor temperature in degrees Celsius.