

Temporal and Multimodal Deep Learning for Cyberattack Detection in LEO Satellite Systems

Kyle Stein, Guillermo Francia, III, Eman El-Sheikh, Hossain Shahriar

Center for Cybersecurity and AI

University of West Florida

Pensacola, FL, USA

{kstein, gfranciaiii, eelsheikh, hshahriar}@uwf.edu

Abstract—The growing reliance on Low-Earth Orbit (LEO) satellite communication systems has increased the need for intelligent methods capable of detecting cyberattacks across complex and dynamic space environments. Unlike conventional network intrusion detection, satellite systems generate heterogeneous information across radio-frequency (RF) links, onboard hardware, and orbital operations. However, many existing approaches either rely on terrestrial intrusion datasets or evaluate individual observations independently, limiting their ability to capture temporal attack behavior specific to LEO satellites. In this work, we conduct a systematic study of deep-learning-based cyberattack detection using the recently introduced satellite-specific UNSW-IoTSAT dataset. We investigate structured learning architectures that preserve hardware, orbital, and RF information, including a Subsystem-Fusion MLP and a hierarchical multimodal Transformer that models both cross-subsystem interactions and temporal evolution. We further evaluate leakage-resistant row-level and temporal settings, along with cross-satellite generalization, to characterize how model architecture and evaluation protocol influence satellite cyberattack detection. Experimental results demonstrate the value of structured multimodal modeling and rigorous evaluation, with the hierarchical Transformer achieving up to 91.66% accuracy and 85.63% macro F1 under the leakage-resistant evaluation protocol.

Index Terms—Deep Learning, LEO Satellite Communications, Malicious Detection, Intrusion Detection Systems

I. INTRODUCTION

Space-based infrastructure has become an increasingly important component of modern communication systems [1]. Low-Earth Orbit (LEO) satellite communication networks now support a wide range of services, including: broadband connectivity, Internet of Things (IoT) applications, maritime operations, emergency and disaster response, and government and defense services [2], [3]. As dependence on satellite-enabled services continues to grow, disruptions to these systems have serious cascading consequences. This risk was demonstrated in the 2022 Viasat KA-SAT cyberattack, where attackers compromised ground infrastructure used to manage satellite terminals and rendered tens of thousands of customer modems inoperable across Europe [4]. Incidents such as these highlight the urgent need for intelligent systems capable of detecting malicious activity within LEO satellite communication systems. Figure 1 provides an overview of this context, illustrating (a) critical services enabled by LEO satellite systems and (b) the communication architecture and cyberattack surface.

However, securing LEO satellite communications from adversarial threats involves more than protecting the contents of

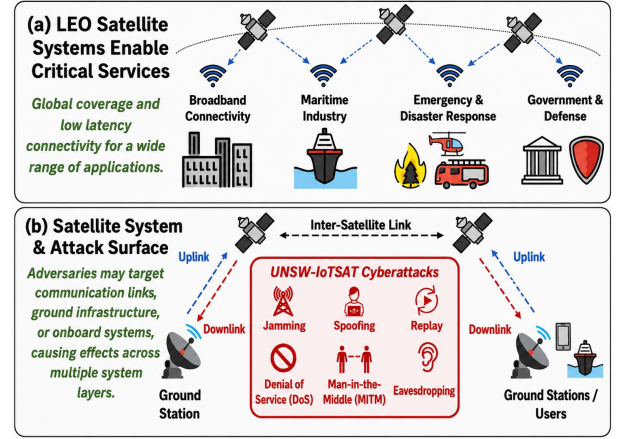


Fig. 1. LEO satellite services and cyberattack surface. (a) Critical services supported by LEO satellite systems. (b) Satellite communication links and cyberattacks represented in the UNSW-IoTSAT dataset.

communication links. Satellite operations rely on complex interactions among communication hardware, onboard systems, orbital state, and radio-frequency (RF) links. Consequently, malicious activity may produce observable effects across multiple sources of system information. For example, RF measurements may capture changes associated with communication-layer attacks, while hardware and orbital telemetry can provide complementary information about the state of the satellite system. Therefore, AI-enabled intrusion detection systems (IDSs) that jointly analyze these heterogeneous measurements may provide a more complete representation of malicious behavior than approaches based on a single source of information.

Recent advances in deep learning have transformed cybersecurity by enabling models to identify complex patterns of malicious behavior across large volumes of system and network data [5], [6]. These capabilities are promising for satellite security since learning-based IDSs can model relationships among multiple measurements rather than relying solely on predefined signatures or isolated indicators. However, progress has been constrained by the limited availability of public datasets that support *multiclass cyberattack* analysis while *jointly representing* LEO satellite operational and communication behavior. Consequently, much of the existing deep learning literature has focused on individual communication-layer threats, such as spoofing and jamming [7], [8]. The recently introduced UNSW-IoTSAT dataset [9], [10] provides

an important foundation for broader evaluation by combining labeled hardware, orbital, and RF measurements under benign and multiclass adversarial conditions.

Although this dataset enables broader evaluation of satellite cyberattack detection, several questions remain regarding how such data should be modeled and evaluated. First, satellite telemetry is inherently temporal, as each individual measurement captures only the system state at a single point in time, while cyberattacks may evolve across a sequence of observations. Treating each observation independently at the row level may therefore discard important information contained in the recent behavior of the system. Second, evaluation methodology can significantly influence reported detection performance. Consecutive observations generated during the same attack occurrence are often highly related, and randomly distributing these observations across train and test sets may lead to strong generalization on paper, but fail to reflect in real-world operational environments. Similar concerns arise when observations from the same satellite appear throughout training and testing, since a model may partially rely on satellite-specific characteristics rather than learning attack behavior that generalizes across different satellite systems.

To address these challenges, this work conducts a systematic study of row-level, temporal, and cross-satellite deep-learning based cyberattack detection on the UNSW-IoTSAT dataset. We develop two structured architectures, a Subsystem-Fusion MLP and a hierarchical multimodal Transformer, that preserve hardware, orbital, and RF information while modeling their relationships within and across satellite observations. We further introduce leakage-resistant data partitioning, matched row-level and temporal evaluation, and bidirectional cross-satellite testing to examine how model architecture, temporal context, and satellite-specific characteristics influence cyberattack detection and generalization. Overall, our key contributions can be summarized as follows:

- We propose both a subsystem-fusion MLP and hierarchical multimodal Transformer that jointly model hardware/environmental, orbital/kinematic, and RF telemetry across consecutive satellite observations for multiclass cyberattack detection.
- We introduce a leakage-resistant evaluation framework and a matched row-level versus temporal comparison to quantify whether temporal context provides meaningful detection gains beyond individual observations.
- We evaluate cross-satellite generalization by training on one satellite and testing on a completely unseen satellite, assessing whether learned representations capture transferable cyberattack behavior.

II. RELATED WORK

LEO satellite communication systems consist of interconnected spacecraft, communication links, ground infrastructure, and user terminals, creating attack surfaces that extend across both cyber and physical components [11], [12]. Recent studies have characterized these threats from both adversarial and system-level perspectives. Peled et al. [13], for example,

extend concepts from the MITRE ATT&CK framework to describe adversarial behavior throughout the satellite attack lifecycle, while Willbold et al. [14] demonstrate practical vulnerabilities in onboard firmware and telecommand interfaces. Other studies identify potential attack surfaces across individual spacecraft subsystems, including attitude determination and control, onboard computing, communications, power, and mission payloads [15], [16]. Therefore, malicious activity affecting one component may also produce observable changes elsewhere in the cyber-physical satellite system.

Consequently, machine and deep learning have been investigated for automated detection of malicious attacks in satellite systems. Conventional approaches include support vector machines for satellite interference recognition [17] and decision-tree intrusion detection for integrated satellite-terrestrial networks [18]. Artificial neural-network feature extraction has also demonstrated promising intrusion-detection performance [19], [20]. These studies demonstrate the potential of learned representations for satellite cybersecurity; however, many existing approaches focus on a specific communication threat, such as spoofing or jamming, or are evaluated using conventional network and IoT datasets that provide limited representation of satellite-specific cyber-physical behavior. Moreover, satellite telemetry is inherently sequential, and comparatively limited attention has been given to determining whether explicitly modeling consecutive observations improves attack detection over row-wise classifiers.

Dataset availability further limits the evaluation of satellite intrusion-detection methods. Operational spacecraft datasets such as the ESA anomaly-detection benchmark [21] and OPS-SAT [22] provide valuable real-world telemetry, but primarily characterize operational anomalies rather than labeled cyberattacks, which is necessary for proper machine/deep learning experimentation. Furthermore, evaluation on closely related observations from the same simulated environment makes it difficult to determine whether a detector has learned transferable attack characteristics across satellite systems. In particular, consecutive observations belonging to the same attack event may be highly similar, and evaluation on the same satellite used during model development does not directly measure generalization to a different satellite platform.

UNSW-IoTSAT [9], [10] provides a unique opportunity to investigate these limitations by combining hardware, orbital, and RF measurements with labeled normal behavior and multiclass cyberattack categories across two satellite nodes. Rather than introducing another attack-specific detector, this work leverages UNSW-IoTSAT to systematically examine how model architecture, temporal context, and evaluation setting influence satellite multiclass cyberattack detection. We compare machine-learning/deep-learning models under leakage-resistant attack-instance partitions, evaluate row-level and temporal models on matched prediction targets, and measure cross-satellite generalization when the target satellite is excluded from model training.

TABLE I
SATELLITE FEATURE GROUPS USED FOR ATTACK DETECTION.

Feature Group	Count	Measurements
Hardware/Environmental	12	Shunt voltage, current, power, three-axis magnetic field, proximity, ambient light, magnetic magnitude, power density, light-to-proximity ratio, and power-to-magnetic-field ratio.
Orbit/Kinematic	9	Latitude, longitude, altitude, north/east/up velocity, total speed, distance from the reference origin, and velocity bearing.
Radio Frequency	10	CRC errors, synchronization-word detections, received signal strength, SNR, bit-error rate, packet-error rate, throughput, frequency offset, Doppler shift, and constellation error.
Total	31	

III. DATASET AND PREPROCESSING

The UNSW-IoTSAT dataset was developed to support cybersecurity research for IoT-enabled satellite communication systems. The dataset was generated using a hybrid satellite IoT testbed that combines real onboard sensor measurements, simulated orbital trajectories, and RF communication between satellite nodes and a ground station. The dataset captures normal operation and six cyberattack categories: jamming, spoofing, replay, denial-of-service (DoS), man-in-the-middle (MITM), and eavesdropping. Each observation contains measurements describing hardware/environmental conditions, orbital/kinematic behavior, and RF/communication characteristics. Furthermore, UNSW-IoTSAT contains observations from two distinct satellite nodes, Satellite1 and Satellite2, enabling evaluation of whether learned attack-detection models generalize across satellite systems.

Dataset Cleaning and Satellite Features: We preserve the original ordering of the observations and normalize the attack labels into seven target classes: normal, jamming, spoofing, DoS, MITM, replay, and eavesdropping. Rather than automatically retaining all numerical fields provided by the dataset, we define an explicit 31-feature list representing three complementary sources of satellite information. The hardware features characterize power consumption, magnetic-field behavior, and environmental sensing. The orbital features describe simulated position and motion. Lastly, the RF features characterize communication quality and receiver behavior. The three feature groups used throughout this study are summarized in Table I. Timestamps, satellite identifiers, source-row identifiers, attack-instance identifiers, and attack-related metadata are retained only for partitioning and auditing and are never supplied to the classifiers. This restricts the models to learn representations from the measurements of underlying physical, orbital, and communication state.

Leakage-Resistant Data Partitioning: Consecutive observations generated during the same cyberattack may be highly dependent, and placing samples from the same attack occurrence in both training and testing can introduce data leakage and inflate estimates of generalization [23]. To reduce this form of leakage, malicious observations are grouped according to their attack timeline, while normal observations are organized into

TABLE II
CLASS DISTRIBUTION OF MATCHED TEMPORAL-WINDOW TARGETS FOR THE PRIMARY EVALUATION PROTOCOL.

Class	Train	Test
Normal	129,670	33,286
Jamming	8,780	2,277
Spoofing	51,616	7,064
DoS	13,592	4,505
MITM	6,069	2,893
Replay	1,652	345
Eavesdropping	47,954	16,185
Total	259,333	66,555

contiguous operating blocks independently for each satellite. Complete attack instances and normal blocks are assigned to a single partition and are never divided across training and testing. For temporal evaluation, these partitions are fixed before any temporal windows are constructed. Sliding windows are then generated independently within each partition and satellite, ensuring that observations contained in training windows cannot also appear in test windows. For cross-satellite evaluation, one satellite instead serves as the source and training domain and the other as a completely held-out testing domain. Both Satellite1-to-Satellite2 and Satellite2-to-Satellite1 transfer directions are evaluated.

Temporal Window Construction: Temporal examples are constructed only after the data partitions have been fixed. Within each partition, observations are ordered independently for each satellite according to their timestamps, with the original source-row index used to resolve ties. For a temporal window of length L ending at observation t :

$$\mathcal{W}_t = [\tilde{\mathbf{x}}_{t-L+1}, \dots, \tilde{\mathbf{x}}_t] \in \mathbb{R}^{L \times D}, \quad (1)$$

where $\tilde{\mathbf{x}}_t \in \mathbb{R}^D$ denotes the preprocessed telemetry vector at observation t , and D denotes the number of retained input features. Temporal windows are constructed with a stride of one, and the class of the final observation is used as the prediction target. Labels from earlier observations are not supplied to the model, and windows may contain transitions between normal and malicious behavior or between different attack categories. This continuous-timeline construction allows

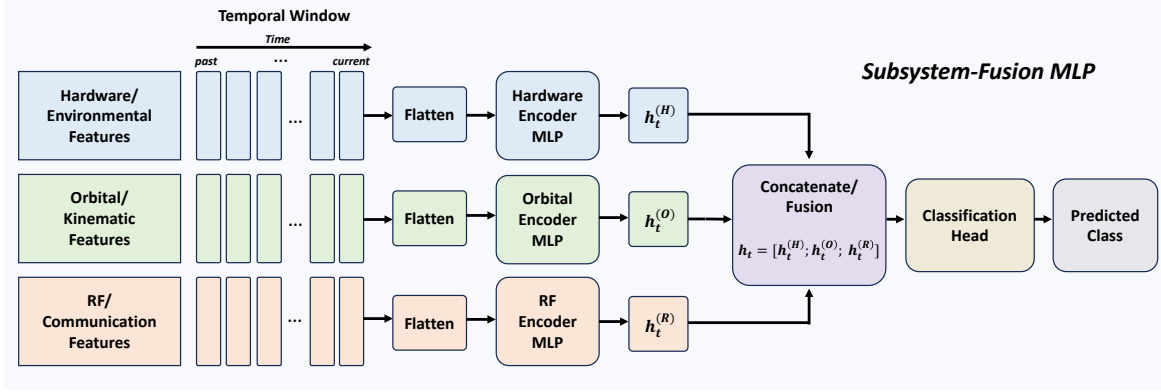


Fig. 2. Overview of the Subsystem-Fusion MLP cyberattack detection framework.

the models to observe how satellite behavior evolves around attack transitions rather than restricting temporal sequences to class-pure observations.

The resulting dataset is naturally imbalanced, with normal operation and several attack categories occurring substantially more frequently than others. We preserve this distribution rather than artificially balancing the training or test sets so that evaluation reflects the underlying dataset composition. Table II summarizes the class distribution of the matched temporal training and test targets. The *same* prediction endpoints are used for the corresponding row-level experiments, allowing temporal and non-temporal models to be compared on identical target observations.

IV. PROPOSED METHOD

In this section, we describe the model architectures and training procedure used to detect benign and malicious behavior in LEO satellite systems. We investigate two structured deep-learning models: a subsystem-fusion MLP and a hierarchical multimodal Transformer. These two architectures constitute the primary methodological implementations of this work. Both proposed architectures explicitly organize the input according to hardware, orbital, and RF measurements. The subsystem-fusion MLP processes these feature groups through separate modality-specific encoders before combining their learned representations for classification. The hierarchical Transformer instead uses self-attention [24] to model interactions among the three satellite subsystems within each observation and then applies a second Transformer stage to model how the resulting satellite-state representations evolve over time. Therefore, the two architectures provide complementary approaches for evaluating the importance of subsystem information and temporal context in satellite cyberattack detection.

A. Subsystem-Fusion MLP

We first develop a Subsystem-Fusion MLP that preserves the three satellite information sources before classification. Rather than directly concatenating all telemetry measurements into a single input vector, hardware, orbital, and RF measurements are processed independently. This allows each encoder to learn representations specialized to the characteristics of

its corresponding satellite subsystem. An overview of the proposed subsystem-fusion MLP is shown in Fig. 2

For observation t , let $\mathbf{x}_t^{(m)} \in \mathbb{R}^{D_m}$ denote the modality-specific subset of $\tilde{\mathbf{x}}_t$, where $m \in \{H, O, R\}$ represents hardware, orbital, and RF information. For a temporal window of length L , let $\mathbf{X}_t^{(m)} \in \mathbb{R}^{L \times D_m}$ contain the corresponding measurements. Each modality is encoded as:

$$\mathbf{h}_t^{(m)} = f_m(\text{vec}(\mathbf{X}_t^{(m)})), \quad (2)$$

where $f_m(\cdot)$ is a modality-specific MLP encoder and $\text{vec}(\cdot)$ flattens the temporal measurements while preserving their chronological ordering. The resulting subsystem representations are concatenated to form:

$$\mathbf{h}_t = [\mathbf{h}_t^{(H)}; \mathbf{h}_t^{(O)}; \mathbf{h}_t^{(R)}]. \quad (3)$$

The fused representation is then supplied to a classification network $g_{\text{cls}}(\cdot)$ which maps the concatenated subsystem representation to logits over the target classes:

$$\hat{\mathbf{y}}_t = g_{\text{cls}}(\mathbf{h}_t), \quad g_{\text{cls}}: \mathbb{R}^{d_f} \rightarrow \mathbb{R}^C, \quad (4)$$

where d_f denotes the dimensionality of the fused representation $\mathbf{h}_t$, and C is the number of target classes. This architecture provides a useful complementary comparison to the proposed Transformer since both models explicitly preserve the hardware, orbital, and RF structure of the satellite measurements. However, the Subsystem-Fusion MLP combines fixed subsystem representations through concatenation and does not explicitly learn attention-based relationships among modalities or across individual satellite states.

B. Hierarchical Multimodal Transformer

The proposed hierarchical multimodal Transformer is designed to capture two complementary forms of structure in satellite telemetry. First, each observation contains measurements describing different parts of the satellite system, including hardware and environmental conditions, orbital and kinematic state, and RF communication behavior. Second, satellite behavior evolves over time, meaning that the significance of the current observation may depend on how the system

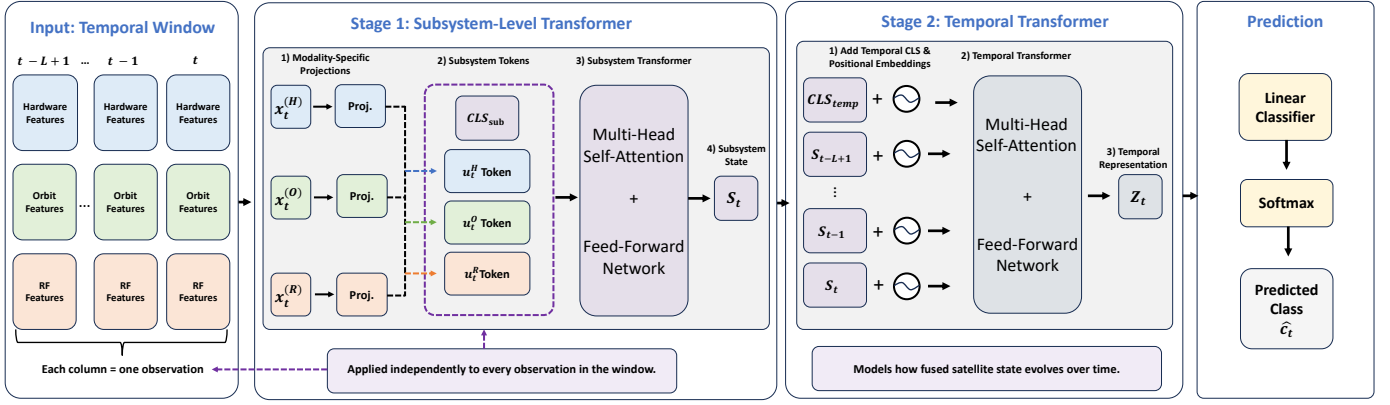


Fig. 3. Overview of the Hierarchical Multimodal Transformer cyberattack detection framework.

state has changed over the preceding observations. Therefore, the architecture operates in two stages: a subsystem-level Transformer that models subsystem relationships within each observation and a temporal Transformer that models relationships across consecutive satellite states. An overview of this architecture is displayed in Figure 3.

Stage 1: Subsystem-Level Transformer. For observation t , let $\mathbf{x}_t^{(H)}$, $\mathbf{x}_t^{(O)}$, and $\mathbf{x}_t^{(R)}$ denote the hardware/environmental, orbital/kinematic, and RF feature vectors, respectively. Since the three groups contain different types and values of measurements, each is independently mapped to a common d -dimensional embedding space using a learnable modality-specific linear projection:

$$\mathbf{u}_t^H = \phi_H(\mathbf{x}_t^{(H)}), \quad \mathbf{u}_t^O = \phi_O(\mathbf{x}_t^{(O)}), \quad \mathbf{u}_t^R = \phi_R(\mathbf{x}_t^{(R)}), \quad (5)$$

where ϕ_H , ϕ_O , and ϕ_R denote the learnable projection functions for the three satellite information sources. The resulting vectors $\mathbf{u}_t^H$, $\mathbf{u}_t^O$, and $\mathbf{u}_t^R$ form compact representations of the hardware, orbital, and RF conditions observed at the same point in time and are treated as subsystem tokens. A learnable subsystem classification token CLS_{sub} is prepended to these tokens to form the subsystem token sequence $[\text{CLS}_{\text{sub}}; \mathbf{u}_t^H; \mathbf{u}_t^O; \mathbf{u}_t^R]$. The resulting sequence contains four tokens: one learnable classification token and one token representing each satellite information source, preserving the subsystem organization of the telemetry while allowing the model to learn relationships among hardware, orbital, and RF conditions within the same observation.

Self-attention is then applied across the tokens using the standard scaled dot-product formulation:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}, \quad (6)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value representations obtained from learned projections of the input tokens, and d_k denotes the dimensionality of the key representations. Multi-head self-attention allows these relationships to be modeled across multiple representation subspaces. In the satellite setting, this enables an RF condition, for example, to be

interpreted jointly with the corresponding hardware and orbital conditions rather than independently. After the subsystem tokens interact through self-attention, the output corresponding to CLS_{sub} is retained as $\mathbf{s}_t$, the fused representation of the satellite state at observation t . Therefore, $\mathbf{s}_t$ summarizes information from all three satellite subsystems and serves as the input representation for the temporal modeling stage.

Stage 2: Temporal Transformer. The subsystem-level Transformer from Stage 1 is applied independently to each observation in a temporal window of length L . This produces L integrated satellite-state representations, $\mathbf{s}_{t-L+1}, \dots, \mathbf{s}_t$, where each $\mathbf{s}_t$ summarizes the combined hardware, orbital, and RF state at a single observation. Unlike the subsystem tokens used in Stage 1, these representations describe the complete satellite state rather than an individual information source. Now, Stage 2 of this architecture models how the complete satellite state changes across consecutive observations.

A learnable temporal classification token CLS_{temp} is prepended to the sequence of satellite-state representations. Learnable positional embeddings $\mathbf{P}$ are also added to preserve the chronological ordering of the observations, producing the temporal token sequence $[\text{CLS}_{\text{temp}}; \mathbf{s}_{t-L+1}; \dots; \mathbf{s}_t] + \mathbf{P}$. The resulting sequence contains $L+1$ tokens: one temporal classification token and L consecutive satellite-state representations.

A second Transformer encoder applies multi-head self-attention across this sequence. This allows changes in communication conditions, hardware behavior, and orbital context to be interpreted relative to preceding observations rather than from the current observation alone. After temporal self-attention, the output corresponding to CLS_{temp} is retained as $\mathbf{z}_t$, which summarizes the complete temporal window. Finally, $\mathbf{z}_t$ is passed to a linear classification head that produces the logit vector:

$$\hat{\mathbf{y}}_t = \mathbf{W}_c \mathbf{z}_t + \mathbf{b}_c. \quad (7)$$

The resulting classification corresponds to the final observation t in the temporal window. Overall, Stage 1 models relationships among satellite subsystems within each observation, while Stage 2 models how the resulting fused satellite state evolves across consecutive observations.

C. Training and Inference

Both architectures are trained end-to-end for supervised classification for benign and malicious cyberattack detection using the class label associated with the final observation of each temporal window. Given a temporal window $\mathcal{W}_t$ ending at observation t , the model produces a logit vector $\hat{\mathbf{y}}_t \in \mathbb{R}^C$, where C denotes the number of target classes. The logits are converted into class probabilities using the softmax function:

$$p_{t,c} = \frac{\exp(\hat{y}_{t,c})}{\sum_{j=1}^C \exp(\hat{y}_{t,j})}, \quad (8)$$

where $p_{t,c}$ denotes the predicted probability that the final observation belongs to class c , and j indexes over all C target classes. During training, the model parameters are optimized using weighted multiclass cross-entropy to account for class imbalance:

$$\mathcal{L}_{\text{WCE}} = -w_{y_t} \log p_{t,y_t}, \quad (9)$$

where y_t denotes the ground-truth class associated with the final observation of the temporal window and w_{y_t} denotes the corresponding class weight. For each class c , the weight is computed as $w_c = N_{\text{train}}/(Cn_c)$, where N_{train} is the total number of training targets, C is the number of target classes, and n_c is the number of training targets belonging to class c .

During inference, the models receive only the satellite measurements contained within the current input window; labels from earlier observations are never provided as model inputs. The Subsystem-Fusion MLP processes the temporal measurements through its modality-specific encoders before combining the resulting subsystem representations for classification. The hierarchical Transformer instead constructs a fused satellite-state representation for each observation and then models the sequence of resulting states through its temporal attention stage. In both architectures, the prediction corresponds to the final observation t in the input window. The final predicted class is obtained by selecting the class with the highest predicted probability:

$$\hat{c}_t = \arg \max_{c \in \{1, \dots, C\}} p_{t,c}. \quad (10)$$

Therefore, each temporal window produces a single classification for its final observation, while the preceding observations provide contextual information that may improve the resulting decision.

D. Row-Level Variations

In addition to the temporal architectures, we construct matched row-level variations that receive only the current observation t . The row-level and temporal models are evaluated on the same target observations, where the row-level model uses only the current observation, while the temporal model also uses preceding observations. This provides a controlled comparison between pointwise and temporal cyberattack detection.

For the Subsystem-Fusion MLP, the row-level variation retains the same modality-specific encoders and fusion mechanism, but removes the temporal window from each subsystem

input. Specifically, the temporal model encodes the vectorized modality window $\text{vec}(\mathbf{X}_t^{(m)})$, while the row-level model receives only the current subsystem feature vector $\mathbf{x}_t^{(m)}$:

$$\mathbf{h}_t^{(m)} = f_m(\mathbf{x}_t^{(m)}), \quad m \in \{H, O, R\}. \quad (11)$$

The resulting hardware, orbital, and RF representations are concatenated using the same fusion operation as the temporal model and passed to the classification head. Therefore, MLP variants differ primarily in whether each subsystem encoder receives a sequence of measurements or a single observation.

For the hierarchical Transformer, the subsystem-level attention stage (Stage 1) is retained and unchanged. The hardware, orbital, and RF measurements from observation t are projected into subsystem tokens and fused through self-attention to obtain the satellite-state representation $\mathbf{s}_t$. The row-level variation then passes $\mathbf{s}_t$ directly to the classification head:

$$\hat{\mathbf{y}}_t = \mathbf{W}_c \mathbf{s}_t + \mathbf{b}_c, \quad (12)$$

rather than constructing the temporal sequence $[\mathbf{s}_{t-L+1}, \dots, \mathbf{s}_t]$. Consequently, the temporal Transformer, temporal classification token, and temporal positional embeddings are omitted (Stage 2). The subsystem-level Transformer is now responsible for modeling relationships among hardware, orbital, and RF measurements, while no attention is performed across preceding observations. Since the row-level and temporal variations are evaluated on identical target observations, differences in performance reflect the contribution of temporal context rather than differences in the evaluated sample set.

V. EXPERIMENTAL RESULTS

Baselines and Metrics. We compare the proposed structured architectures against representative learning-based approaches commonly used for satellite and cyberattack detection [20], [25], [26]. Specifically, we evaluate Random Forest and XGBoost as conventional machine-learning baselines and implement a monolithic MLP as a deep-learning baseline. The monolithic MLP processes all hardware, orbital, and RF measurements jointly *without* preserving the subsystem-specific organization used by the proposed models. To provide matched temporal comparisons, we also construct temporal variants of each baseline. For Random Forest, XGBoost, and the monolithic MLP, the measurements from all L observations in a temporal window are ordered chronologically and flattened into a single feature vector before being supplied to the model. These temporal variants receive the same observation history and predict the same final-observation targets as the proposed temporal architectures, but do not explicitly model subsystem or sequential structure through attention.

All methods are evaluated using the same data partitions and matched prediction targets. We report accuracy, macro precision, macro recall, and macro F1 across five independent runs with different random seeds. Given the imbalanced class distribution, macro F1 is used as the primary evaluation metric

TABLE III
ATTACK-CLASSIFICATION PERFORMANCE AND INFERENCE EFFICIENCY UNDER MATCHED ROW-LEVEL AND TEMPORAL EVALUATION. PERFORMANCE METRICS ARE REPORTED AS MEAN $\pm$ STANDARD DEVIATION ACROSS RUNS.

Input	Method	Acc.	Prec.	Rec.	F1	Infer. (ms/sample)
Row	Random Forest [20]	87.34 $\pm$ 0.33	78.08 $\pm$ 0.68	75.73 $\pm$ 0.22	76.08 $\pm$ 0.42	0.246
	XGBoost [25]	76.64 $\pm$ 0.18	61.32 $\pm$ 0.31	73.58 $\pm$ 0.11	62.39 $\pm$ 0.24	0.011
	Monolithic MLP [26]	90.18 $\pm$ 0.74	83.90 $\pm$ 2.17	76.32 $\pm$ 3.11	76.51 $\pm$ 3.49	0.008
	Subsystem-Fusion MLP Transformer	91.35$\pm$0.10	94.44$\pm$0.59	78.52$\pm$0.91	84.65$\pm$0.92	0.144
Temporal	Random Forest-T [20]	83.82 $\pm$ 0.15	72.66 $\pm$ 0.08	71.41 $\pm$ 0.63	71.15 $\pm$ 0.40	0.419
	XGBoost-T [25]	75.50 $\pm$ 0.21	63.87 $\pm$ 0.10	74.46 $\pm$ 0.04	65.30 $\pm$ 0.15	0.012
	Monolithic MLP-T [26]	89.77 $\pm$ 0.50	80.24 $\pm$ 2.64	76.38 $\pm$ 2.14	74.90 $\pm$ 2.01	0.009
	Subsystem-Fusion MLP-T Transformer-T	90.13 $\pm$ 0.44	83.46 $\pm$ 0.11	75.98 $\pm$ 2.25	75.66 $\pm$ 2.20	0.132
		91.66$\pm$0.02	95.50$\pm$0.21	81.21$\pm$0.13	85.63$\pm$0.14	0.041

since it assigns equal importance to each target class regardless of its frequency in the dataset.

Implementation Details. The neural models were implemented in PyTorch and trained using the MPS backend and conducted on an Apple MacBook Pro equipped with an M5 Pro processor. Neural models were trained for a maximum of 10 epochs with a batch size of 64 using AdamW with a learning rate of 1×10^{-3} and weight decay of 1×10^{-4} . Weighted cross-entropy loss was used to address class imbalance. The Transformer uses an embedding dimension of 64, four attention heads, and a feed-forward dimension of 128.

A. Main Experimental Results

Table III presents the primary attack-classification results under matched row-level and temporal evaluation. This experiment evaluates each model using the same data partitions and prediction targets *without* holding out an entire satellite for testing. In the row-level setting, each model predicts the class of an observation using only the measurements available at that observation. In the temporal setting, the same target observations are evaluated using an eight-observation window with a stride of one, allowing the models to incorporate the previous satellite measurements when predicting the class of the final observation. Furthermore, inference latency is also reported for each predicted sample.

Among the row-level methods, the proposed Subsystem-Fusion MLP achieves the highest macro F1 at 84.65%, outperforming the row-level Transformer (82.09%) and monolithic MLP (76.51%). This suggests that preserving the subsystem structure of hardware, orbital, and RF can improve the quality of learned representations over a single unstructured input. Under the temporal evaluation experiments, the hierarchical Transformer performs best overall, reaching 91.66% accuracy and 85.63% macro F1. Relative to its row-level counterpart, it improves macro precision by 4.8% and macro recall by 2.5%, indicating fewer false-positive predictions relative to true positives and fewer missed class instances.

In contrast, temporal context does not consistently improve the other models. Random Forest-T, monolithic MLP-T, and Subsystem-Fusion MLP-T all achieve lower macro F1 than

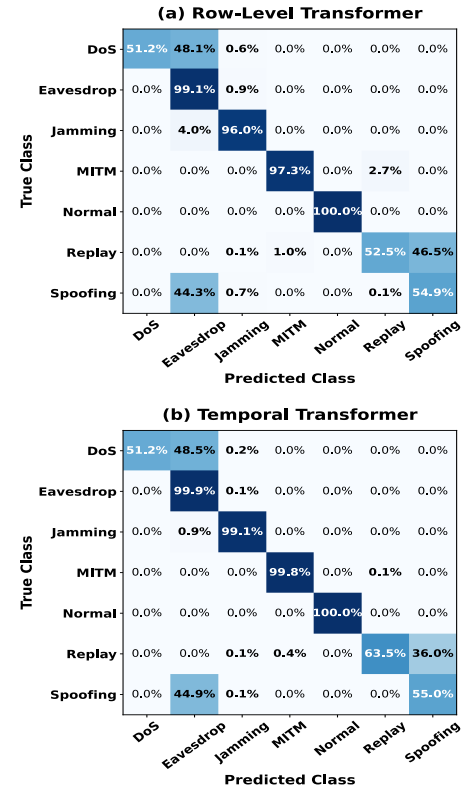


Fig. 4. Confusion matrices for the (a) row-level and (b) temporal Transformer across five runs. Values represent the percentage of each true class assigned to each predicted class.

their row-level counterparts. These models receive preceding observations as a fixed input representation, while the hierarchical Transformer explicitly models relationships across satellite states using temporal self-attention. This suggests that temporal information is most useful when the architecture can explicitly model relationships across observations. Inference latency further highlights the performance-efficiency trade-off. Although the monolithic MLP and XGBoost are fastest, Transformer-T requires only 0.041 ms per prediction, compared with 0.132 ms for Subsystem-Fusion MLP-T and 0.419 ms for Random Forest-T. Therefore, the temporal Transformer

TABLE IV

CROSS-SATELLITE GENERALIZATION PERFORMANCE. SOURCE $\rightarrow$ TARGET DENOTES THE SATELLITE USED FOR TRAINING AND THE UNSEEN SATELLITE USED FOR TESTING, RESPECTIVELY. RESULTS ARE REPORTED AS MACRO F1 MEAN $\pm$ STANDARD DEVIATION ACROSS RUNS.

Input	Method	Sat. 1 $\rightarrow$ Sat. 2	Sat. 2 $\rightarrow$ Sat. 1
Row	Random Forest [27]	78.78 $\pm$ 1.20	88.78$\pm$0.78
	XGBoost [25]	80.08 $\pm$ 0.22	79.67 $\pm$ 1.52
	Monolithic MLP	78.92 $\pm$ 5.79	22.26 $\pm$ 10.67
	Subsystem-Fusion MLP	80.19$\pm$4.01	87.29 $\pm$ 5.02
	Transformer	77.63 $\pm$ 6.68	88.33 $\pm$ 3.86
Temporal	Random Forest-T	70.06 $\pm$ 1.05	81.55 $\pm$ 0.64
	XGBoost-T	80.95 $\pm$ 0.26	84.69 $\pm$ 1.48
	Monolithic MLP-T	61.23 $\pm$ 4.74	14.74 $\pm$ 4.44
	Subsystem-Fusion MLP-T	88.47$\pm$0.63	85.86 $\pm$ 3.00
	Transformer-T	76.53 $\pm$ 5.38	90.06$\pm$1.23

achieves the strongest classification performance while maintaining relatively low inference latency.

To further examine class-specific performance, Fig. 4 presents the mean normalized confusion matrices for the row-level and temporal Transformer models. Temporal modeling provides the clearest improvements for Replay, Jamming, and MITM attack detection. Replay increases from 52.5% to 63.5%, primarily through a reduction in confusion with Spoofing from 46.5% to 36.0%. Jamming improves from 96.0% to 99.1%, while MITM increases from 97.3% to 99.8%. In contrast, DoS remains difficult for both models, with approximately half of its observations classified as Eavesdropping, while Spoofing also remains frequently confused with Eavesdropping. These results indicate that the overall temporal improvement is concentrated in specific attack classes rather than being uniformly distributed across all categories.

B. Cross-Satellite Generalization

Table IV reports macro F1 when each model is trained and validated on one satellite and evaluated on the other as an unseen target satellite. This setting provides a more strict generalization test than the primary evaluation, preventing the target satellite from contributing any observations during model development. In addition to measuring transfer between the two satellites represented in UNSW-IoTSAT, the experiment provides an initial indication of how well learned attack representations may transfer to previously unseen satellites. Although this setting does not fully reproduce the variability of an operational LEO constellation, it offers a useful experimental setup for evaluating whether a detector relies primarily on characteristics of the source satellite or captures cyberattack behavior that remains informative across satellite systems.

Cross-satellite performance varies considerably across both model architecture and transfer direction. For Satellite 1 $\rightarrow$ Satellite 2, the temporal Subsystem-Fusion MLP achieves the highest macro F1 of 88.47%, substantially improving over its row-level counterpart at 80.19%. For Satellite 2 $\rightarrow$ Satellite 1, the temporal Transformer performs best, reaching 90.06% macro F1. In contrast, the monolithic MLP exhibits substantial

TABLE V

MODALITY ABLATION UNDER TEMPORAL EVALUATION. ALL MODELS USE AN EIGHT-OBSERVATION WINDOW. RESULTS ARE REPORTED AS MACRO F1 MEAN $\pm$ STANDARD DEVIATION ACROSS RUNS.

Input Modalities	#Feat.	Fusion MLP-T	Transformer-T
Hardware (H)	12	5.62 $\pm$ 1.04	9.43 $\pm$ 0.10
Orbit (O)	9	5.78 $\pm$ 0.71	6.95 $\pm$ 0.24
RF	10	75.03 $\pm$ 1.12	82.38 $\pm$ 4.63
$H + \text{RF}$	22	73.97 $\pm$ 1.02	85.47 $\pm$ 0.19
$O + \text{RF}$	19	76.46 $\pm$ 3.43	80.50 $\pm$ 4.54
$H + O + \text{RF}$	31	75.66 $\pm$ 2.20	85.63 $\pm$ 0.14

degradation in both directions, particularly for Satellite 2 $\rightarrow$ Satellite 1, where macro F1 falls to 22.26% for the row-level model and 14.74% for its temporal variant. The strongest cross-satellite results are obtained by the structured temporal models, although transfer performance remains sensitive to the source-target direction.

C. Modality-Ablation

Table V reports macro F1 for different combinations of hardware, orbital, and RF inputs under the temporal evaluation setting. This experiment evaluates which satellite information sources contribute most to detection performance for the Subsystem-Fusion MLP-T and Transformer-T models.

Across both architectures, RF measurements provide the strongest individual modality, reaching 75.03% macro F1 for the Subsystem-Fusion MLP-T and 82.38% for Transformer-T, while hardware-only and orbit-only inputs perform substantially worse. For Transformer-T, adding hardware to RF increases macro F1 to 85.47%, and the full $H + O + \text{RF}$ configuration reaches 85.63%, indicating that most of the improvement beyond RF alone comes from the hardware features. The relatively small gain from adding orbital information suggests that orbital features provide limited additional discriminative value once hardware and RF measurements are available. A similar pattern is observed for the Subsystem-Fusion MLP-T, where adding hardware or orbital features to RF does not consistently improve performance. Overall, RF measurements contain the majority of the discriminative information in UNSW-IoTSAT, while hardware and orbital features can provide complementary information for the temporal models.

D. Effect of Temporal Window Length

Table VI examines the sensitivity of the hierarchical Transformer to the amount of temporal context using window lengths $L \in \{2, 4, 8, 16\}$. To ensure a controlled comparison, all window lengths are evaluated on the same prediction targets, defined by the observations with sufficient history to construct a complete 16-observation window. Therefore, each model predicts the same target observations while receiving a different amount of preceding context. This matched-endpoint design isolates the effect of temporal history and explains why the $L = 8$ result in this ablation differs slightly from the main experiment, where the 8-observation model was evaluated on

TABLE VI
EFFECT OF TEMPORAL WINDOW LENGTH ON TRANSFORMER-T
PERFORMANCE.

Window Length (L)	Macro F1
2	79.35 $\pm$ 6.41
4	78.22 $\pm$ 5.15
8	84.78 $\pm$ 1.26
16	83.81 $\pm$ 4.67

the larger set of all valid 8-window endpoints. Among the tested settings, $L = 8$ achieves the highest macro F1, reaching 84.78 ± 1.26 , and also exhibits the lowest variability across runs. Although the longer $L = 16$ window attains competitive performance, it is both slightly lower in mean macro F1 and substantially less stable than $L = 8$. Overall, these findings suggest that the hierarchical Transformer benefits from temporal context, but that the benefit saturates beyond approximately eight observations.

E. Effect of Leakage-Resistant Data Partitioning

Table VII evaluates the effect of data partitioning on row-level cyberattack classification. We hypothesize that consecutive observations generated during the same cyberattack occurrence may be highly dependent, such that randomly distributing these observations across training and testing can produce overly optimistic estimates of generalization. To evaluate this effect, we compare the instance-held-out partitioning strategy used throughout the primary experiments with a conventional random-row split. The random-row protocol preserves the same per-class training and test sample counts as the instance-held-out protocol, but does not preserve attack-instance boundaries. All other preprocessing procedures, model configurations, and evaluation settings are held constant.

As predicted, both architectures achieve substantially higher performance under random-row partitioning. The Subsystem-Fusion MLP increases from 84.65% to 92.44% macro F1, while the Transformer increases from 82.09% to 92.26%. Accuracy also increases by 4.23% for the Subsystem-Fusion MLP and 4.39% for the Transformer. These results demonstrate that the evaluation partitioning strategy has a substantial effect on detection performance. When observations from the same attack occurrences can appear across training and testing, the resulting performance may overestimate generalization to previously unseen attack instances. This finding supports the use of instance-held-out partitioning for the primary experiments in this study.

VI. DISCUSSION & LIMITATIONS

The experimental results highlight an important relationship between model architecture and the structure of the available satellite information. At the row level, the Subsystem-Fusion MLP outperforms the Transformer, suggesting that when only a single observation is available, separately encoding subsystem measurements before fusion may be sufficient to capture much of the useful information without requiring attention-based modeling. However, this pattern changes when temporal

TABLE VII
EFFECT OF DATA PARTITIONING ON ROW-LEVEL ATTACK-CLASSIFICATION
PERFORMANCE. Δ DENOTES THE PERCENTAGE POINT INCREASE UNDER
RANDOM-ROW PARTITIONING.

Method	Metric	Instance-Held-Out	Random-Row	Δ
Subsystem-Fusion MLP	Acc.	91.35 $\pm$ 0.10	95.58 $\pm$ 0.13	+4.23%
	F1	84.65 $\pm$ 0.92	92.44 $\pm$ 0.47	+7.79%
Transformer	Acc.	91.18 $\pm$ 0.08	95.57 $\pm$ 0.05	+4.39%
	F1	82.09 $\pm$ 1.02	92.26 $\pm$ 0.46	+10.17%

context is introduced. Providing previous observations does not consistently improve the Random Forest, monolithic MLP, or Subsystem-Fusion MLP, while the hierarchical Transformer improves over its row-level counterpart. These findings suggest that the value of temporal information depends on access to preceding observations and also how the relationships across those observations are modeled. In particular, temporal self-attention in the transformer provides an explicit mechanism for relating representations across the observation window, which may help identify attack behavior that develops over time.

The modality-ablation results further show that detection performance is strongly influenced by the availability of RF measurements. RF features provide the majority of the discriminative information in UNSW-IoTSAT, while hardware and orbital measurements provide additional value for the temporal models. This dependence on RF information should be considered when translating these results to operational satellite systems. Encryption of communication payloads would not necessarily prevent the collection of physical- and link-layer measurements such as received signal strength, SNR, bit-error rate, frequency offset, Doppler shift, or synchronization behavior, since these measurements do not require access to the decoded payload. However, future evaluations should consider partially unavailable RF feature sets to determine how detection performance changes when the complete set of communication measurements cannot be observed.

The proposed architectures present different computational and adaptation trade-offs. The Subsystem-Fusion MLP provides a simpler architecture and achieves the strongest row-level performance, while the hierarchical Transformer achieves the strongest overall performance and maintains relatively low inference latency. Although the MLP may be less computationally expensive to train, the Transformer provides a flexible foundation for future adaptation after deployment. Pretrained transformer-based models can support parameter-efficient adaptation methods through the use of low-rank adaptation [28] in which only a small portion of the model is updated, creating opportunities for future continual- and few-shot-learning approaches that incorporate newly observed attack behavior *without* the need to retrain the complete network. These capabilities were not evaluated in the present study and remain an important direction for future work, particularly in satellite environments where new attacks and changing operating conditions may be encountered over time.

Finally, the results should be interpreted within the scope of the UNSW-IoTSAT dataset. Although the cross-satellite experiment provides a stronger generalization test by withholding an entire satellite during model development, the dataset contains only two satellite nodes while real-world operational environments found across LEO constellations may contain thousands of satellite nodes. The observed transfer performance should be viewed as evidence of cross-satellite generalization within the UNSW-IoTSAT environment rather than proof of generalization to arbitrary satellite platforms. Evaluation across additional spacecraft, communication systems, and more expansive datasets as they become publicly available will be necessary to determine how well these learned attack representations transfer beyond the environments considered in this study.

VII. CONCLUSION

This paper investigated machine- and deep-learning approaches for classifying benign and malicious behavior in LEO satellites using the UNSW-IoTSAT dataset. Through row-level, temporal, leakage-resistant, and cross-satellite evaluation, we examined how model architecture, temporal context, and data partitioning influence cyberattack detection. We introduced structured Subsystem-Fusion MLP and hierarchical multimodal Transformer architectures that preserve hardware, orbital, and RF information, with the temporal Transformer achieving the strongest overall performance. The results further show that random-row partitioning can inflate performance and that strong within-satellite results do not always translate consistently to unseen satellites. Future work will extend this analysis to additional satellite platforms and operational datasets while exploring adaptive learning methods for emerging attack behavior.

REFERENCES

- [1] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff *et al.*, "Satellite communications in the new space era: A survey and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 70–109, 2020.
- [2] F. S. Prol, R. M. Ferre, Z. Saleem, P. Välsuö, C. Pinell, E. S. Lohan, M. Elsanhoury, M. Elmusrati, S. Islam, K. Çelikbilek *et al.*, "Position, navigation, and timing (pnt) through low earth orbit (leo) satellites: A survey on current status, challenges, and opportunities," *IEEE access*, vol. 10, pp. 83 971–84 002, 2022.
- [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*, vol. 25, no. 3, pp. 1604–1652, 2023.
- [4] N. Boschetti, N. G. Gordon, and G. Falco, "Space cybersecurity lessons learned from the viasat cyberattack," in *ASCEND 2022*, 2022, p. 4380.
- [5] K. Stein, G. Francia, E. El-Sheikh, and A. A. Mahyari, "Packet inspection transformer: A self-supervised journey to unseen malware detection with few samples," *IEEE Access*, vol. 13, pp. 196 336–196 354, 2025.
- [6] E. C. P. Neto, S. Iqbal, S. Buffett, M. Sultana, and A. Taylor, "Deep learning for intrusion detection in emerging technologies: a comprehensive survey and new perspectives," *Artificial Intelligence Review*, vol. 58, no. 11, p. 340, 2025.
- [7] J. Wigchert, S. Sciancalepore, and G. Oliveri, "Detection of aerial spoofing attacks to leo satellite systems via deep learning," *Computer Networks*, vol. 269, p. 111408, 2025.
- [8] I. E. Mehr and F. Doyis, "A deep neural network approach for classification of gnss interference and jamming," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 2, pp. 1660–1676, 2024.
- [9] O. Abdelhameed, B. Turnbull, and N. Koroniotis, "Unsw-iotsat: A hybrid testbed-derived dataset for enabling cybersecurity research in the smart satellite domain," *Cyber Security and Applications*, vol. 4, p. 100133, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S277291842600010X>
- [10] O. Abdelhameed, "UNSW-IoTSAT: A ccscs-compliant dataset for iot-based satellite cybersecurity," 2026, gitHub repository. [Online]. Available: <https://github.com/Osama-Abdelhameed/UNSW-IoTSAT>
- [11] A. Sharmin, B. U. Mahmud, N. Nabi, M. Shaima, and M. J. H. Faruk, "Cyber attacks on space information networks: vulnerabilities, threats, and countermeasures for satellite security," *Journal of Cybersecurity and Privacy*, vol. 5, no. 3, p. 76, 2025.
- [12] European Union Agency for Cybersecurity, "Low Earth Orbit (LEO) SATCOM Cybersecurity Assessment," European Union Agency for Cybersecurity, ENISA Report, Feb. 2024, published February 15, 2024. [Online]. Available: <https://www.enisa.europa.eu/publications/low-earth-orbit-leo-satcom-cybersecurity-assessment>
- [13] R. Peled, E. Aizikovich, E. Habler, Y. Elovici, and A. Shabtai, "Evaluating the security of satellite systems," *arXiv preprint arXiv:2312.01330*, 2023.
- [14] J. Willbold, M. Schloegel, M. Vögele, M. Gerhardt, T. Holz, and A. Abbasi, "Space odyssey: An experimental software security analysis of satellites," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1–19.
- [15] A. R. K. Verma, "Cybersecurity in satellite communication networks: Key threats and neutralization measures," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 5667–5692, 2025.
- [16] S. Salim, N. Moustafa, and M. Reisslein, "Cybersecurity of satellite communications systems: A comprehensive survey of the space, ground, and links segments," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 1, pp. 372–425, 2024.
- [17] Y. Yang and L. Zhu, "An efficient way for satellite interference signal recognition via incremental learning," in *2019 international symposium on networks, computers and communications (ISNCC)*. IEEE, 2019, pp. 1–5.
- [18] K. S. J. Yap, "A network intrusion detection system using decision tree machine learning on an istn architecture," Master's thesis, Naval Postgraduate School, Monterey, California, Mar. 2022. [Online]. Available: <https://hdl.handle.net/10945/69725>
- [19] N. Koroniotis, N. Moustafa, and J. Slay, "A new intelligent satellite deep learning network forensic framework for smart satellite networks," *Computers and Electrical Engineering*, vol. 99, p. 107745, 2022.
- [20] A. T. Azar, E. Shehab, A. M. Mattar, I. A. Hameed, and S. A. Elsaid, "Deep learning based hybrid intrusion detection systems to protect satellite networks," *Journal of Network and Systems Management*, vol. 31, no. 4, p. 82, 2023.
- [21] K. Kotowski, C. Haskamp, J. Andrzejewski, B. Ruszczak, J. Nalepa, D. Lakey, P. Collins, A. Kolmas, M. Bartesaghi, J. Martinez-Heras *et al.*, "European space agency benchmark for anomaly detection in satellite telemetry," *arXiv preprint arXiv:2406.17826*, 2024.
- [22] B. Ruszczak, K. Kotowski, D. Evans, and J. Nalepa, "The ops-sat benchmark for detecting anomalies in satellite telemetry," *Scientific Data*, vol. 12, no. 1, p. 710, 2025.
- [23] R. Flood, G. Engelen, D. Aspinall, and L. Desmet, "Bad design smells in benchmark nids datasets," in *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2024, pp. 658–675.
- [24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [25] S. T. Hamidou and A. Mehdi, "Enhancing ids performance through a comparative analysis of random forest, xgboost, and deep neural networks," *Machine Learning with Applications*, p. 100738, 2025.
- [26] S. Cherfi, A. Lemouari, and A. Boulaiche, "Mlp-based intrusion detection for securing iot networks," *Journal of Network and Systems Management*, vol. 33, no. 1, p. 20, 2025.
- [27] M. Udurume, V. Shakhov, and I. Koo, "Comparative analysis of deep convolutional neural network—bidirectional long short-term memory and machine learning methods in intrusion detection systems," *Applied Sciences*, vol. 14, no. 16, p. 6967, 2024.
- [28] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9>