Semantic-Aware Anomaly Detection for Satellite–IoT Networks: A Lightweight Transformer-Based Approach

Security for Space Systems (3S) Conference 2025

November 6, 2025

Junbeom Park[1], Zizung Yoon[2] and Jongsou Park[1]

[1]Department of Computer Engineering, Korea Aerospace University
[2]Department of Smart Drone Engineering, Korea Aerospace University

CONTENTS

1

Introduction

Architectural and Security Challenges

- **Satellite–IoT networks** face severe **architectural** and **resource constraints,** limiting the deployment of conventional security mechanisms.
  - **Resource limitations and restricted access**
    - Hinder timely software updates and end-to-end protection, exposing devices as **vulnerable endpoints.**
  - **Segmented and heterogeneous architectures**
    - Diverse operating systems and communication protocols hinder unified security enforcement across **ground, space,** and **user segments.**
  - **Lack of built-in security features**
    - Many IoT and user-segment devices omit **intrusion detection** or **encryption** due to *hardware* and *cost constraints.*
  - **Absence of runtime anomaly detection**
    - Limits real-time response and **structured threat assessment.**
- **These challenges emphasize the urgent need for lightweight and integrated security mechanisms** specifically tailored to **Satellite–IoT environments.**

Motivation for Semantic-Aware Detection

- **Traditional IDSs** fail to capture **semantic dependencies** across structured packet fields.
  - **Rule-based or statistical IDSs**
    - Restricted to syntactic validation and unable to interpret **contextual relationships** among fields.
  - **Deep learning–based IDSs**
    - Computationally heavy and less robust when packets are **incomplete** or **degraded**.
- **Proposed Semantic-Aware Approach for Satellite–IoT**
  - **Sentence-based packet representation**
    - Converts structured packets into **natural-language-like sentences** preserving **contextual semantics.**
  - **Lightweight DistilBERT model**
    - Performs **semantic inference** with *reduced latency* and *memory usage.*
  - **Scenario-driven dataset design**
    - Constructed with **15 protocol- and security-aware fields** derived from **CSP, CCSDS, MIOTY,** and T**ON-IoT** specifications.
- This approach enables **accurate** and **interpretable anomaly detection,** ensuring practical feasibility for **real-time deployment** in **Satellite–IoT systems.**

2

Related Work

Related Work and Research Gap

❖ Existing studies on Satellite–IoT security

- Rule-based and statistical IDSs focus on **syntax-level anomalies** in telemetry and command traffic.
- They lack **semantic reasoning** and do not adapt well to **multi-segment Satellite–IoT links**.

❖ LLM-based detection approaches

- Language-model-based IDSs capture **contextual dependencies**, improving detection accuracy.
- However, most rely on **synthetic datasets** and overlook **efficiency under resource constraints**.

❖ Identified gaps

- **Lightweight, real-time frameworks** that encode **protocol-level semantics** for Satellite–IoT are still lacking.
- Validation under **realistic, resource-constrained environments** remains limited.

❖ Our contribution

- We propose a **DistilBERT-based semantic anomaly detection approach** tailored for Satellite–IoT networks.
- The model learns **inter-field dependencies** in structured packets, enabling **accurate and efficient detection** even under **missing-field conditions**.

3

Proposed Methodology

End-to-End Process

- **Our detection pipeline** integrates three main stages to achieve **semantic-aware classification**.
- **Packet-to-Sentence Construction**
  - Structured packets parsed into **15 protocol-aware fields** (CSP, MIOTY, CCSDS, TON_IoT).
  - Serialized into **sentences** preserving **inter-field dependencies**.
- **Semantic Inference & Anomaly Classification**
  - **DistilBERT (6-layer)** encodes contextual relations via **WordPiece tokenization**.
  - [CLS] token output classified as **Benign** or **Anomalous**.
- **Attack Type Classification & Logging**
  - Anomalous packets categorized into **Injection**, **Replay**, or **Privilege Abuse**.
  - Logged for **forensic analysis** and response.

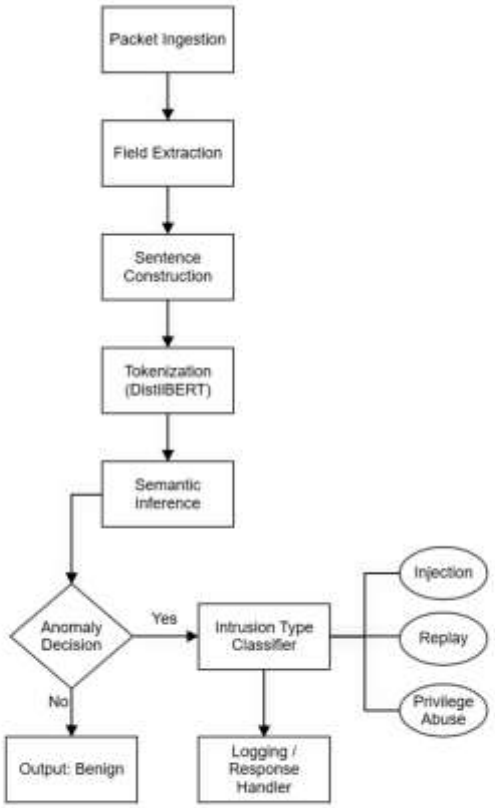

Fig. 1. End-to-end flow diagram of the proposed sentence-based intrusion detection process.

Protocol-Aware Packet Structure (15 Fields)

TABLE I
STRUCTURE AND DESCRIPTION OF THE 15-FIELD UAV–SATELLITE DATASET SCHEMA

Field Name	Description	Example Values
timestamp	Packet generation time in ISO 8601 format	2025-07-01T03:15:20
src / dst	Valid communication nodes from NORMAL_LINKS	gw1→iot, leo→gcs[a]
priority	Message priority determined by msg_type	LOW, MEDIUM, HIGH, CRITICAL
src_port / dst_port	Ports assigned per node from SRC_PORT_MAP / DST_PORT_MAP	gw1:1883, leo:3001[b]
src_region / dst_region	Region code from REGION_MAP	AS→AF, EQ→SP
orbit_class	Orbit category derived from ORBIT_CLASS_MAP	LEO, MEO, N/A
msg_type	Message type based on VALID_MSG_TYPES per src–dst pair	telemetry, data, command, status, ack, alert[c]
payload_type	Field type determined by msg_type	TEMP, SIZE, MOVE, SIGNAL_LOSS, NORMAL[d]
payload	Formatted content generated per payload_type	`TEMP=24.5, command=RESET`
label	Class label for anomaly detection	Normal, Injection, Replay, Privilege Abuse, Jamming, Spoofing
ttl (time-to-live)	TTL value based on src/dst role	64, 128, 200, 255
flags	Control flags by msg_type	ACK, SYN, RST[e]

[a] Examples include (gw1, iot), (rt, gw2), (gcs, meo), based on NORMAL_LINKS.
[b] Port numbers are statically assigned per node based on system design; for example, ground nodes use ports in the 1000 range (e.g., gw1:1883), while space-segment nodes use ports in the 3000 range (e.g., leo:3001).
[c] Allowed message types are predefined for each src–dst pair in the system design.
[d] Mapping: telemetry→{TEMP, HUM, POS, BATT}; data→{COORD, SIZE, DATA_TYPE, REF_ID}; command→{ACTIVATE, MOVE, RESET, ...}; ack→{RECEIVED, EXECUTED}; alert→{ANOMALY_DETECTED, ...}; status→{NORMAL, LOW_BATTERY, ...}.
[e] Flag options include ACK, PSH, ENC, SYN, and RST, as listed in FLAGS_BY_MSGTYPE.

❖ Design Overview

- Defines **15 protocol-aware fields** combining **temporal**, **spatial**, and **semantic** attributes.
- Derived from **CSP**, **MIOTY**, **CCSDS**, and **TON_IoT** standards.

❖ Field Groups

- **Metadata** – node identity and timing (**timestamp**, **src**, **dst**, **ports**).
- **Semantics** – protocol intent and control (**msg_type**, **payload**, **priority**, **flags**).
- **Context** – orbital and security attributes (**orbit_class**, **regions**, **ttl**, **label**).

Model Choice: DistilBERT for Semantic Inference

❖ Motivation

- **Satellite–IoT systems** require on-board inference under strict **CPU, memory, and power constraints**.
- Full-scale LLMs such as **BERT** or **RoBERTa** offer strong contextual reasoning but are **too heavy for real-time embedded deployment**.
- An effective model must preserve **semantic understanding** while operating within **limited computational resources**.

❖ Model Selection: DistilBERT

- **Compact six-layer architecture** retains *BERT-level accuracy* while reducing model size and inference latency.
- **40 % smaller** and **~60 % faster** than BERT, using **≈ 480 MB memory** and **≈ 25 ms inference** per packet on **Jetson Nano / Raspberry Pi 4 with MIOTY**-based sensor inputs.
- Maintains **contextual reasoning across packet fields**, enabling *semantic anomaly detection* in constrained edge environments.
- Lighter **RNN- or CNN-based IDSs** lack this field-level semantic awareness and fail to generalize.

Sentence-Based Representation and Semantic Encoding

❖ Concept Overview

- Structured packets are converted into **sentence-like representations**, allowing **DistilBERT** to capture **contextual semantics** and **inter-field dependencies** beyond conventional IDSs.

❖ Example Sentences

- *Normal:* "Telemetry message from **leo** to **gcs** carrying TEMP=22.5 with priority HIGH and flag ENC at 2025-07-10T08:30:00Z."
- *Privilege Abuse:* "At 2025-07-10T08:45:12Z, node **iot** sent command=RESET to **gcs** with flag SYN, violating access control policies."

❖ DistilBERT Encoding Process

- **Tokenize** → Decompose field–value pairs via **WordPiece** with [CLS]/[SEP] tokens.
- **Encode** → Six-layer Transformer models **contextual relations** among fields.
- **Classify** → [CLS] embedding produces an **anomaly score** and threat label.

❖ Main Observation

- Sentence-based formulation bridges **structured syntax** and **semantic reasoning**, enabling **accurate packet-level anomaly detection** in Satellite–IoT networks.

Semantic Inference and Threat Classification

❖ Main Concept

- **DistilBERT** performs **semantic reasoning** over tokenized packet sentences and classifies them into **four security-relevant categories** reflecting real **Satellite–IoT** behaviors.

❖ Threat Label Descriptions

- **Normal:** Packets follow expected communication flows with valid field values (src, dst, msg_type, payload).
  - *Example: leo → gcs, msg_type = telemetry, payload = TEMP = 22.5, flag = ENC.*
- **Injection:** Packets include malformed or semantically inconsistent payloads that violate protocol or structure rules.
  - *Example: iot → gw1, msg_type = command, payload = CALIBRATE, flag = URG.*
- **Replay:** Previously transmitted timestamps or payloads are reused, disrupting temporal consistency.
  - *Example: uav → gw2, repeating telemetry with payload = POS = 37.4,127.1.*
- **Privilege Abuse:** Low-privilege nodes (e.g., iot) issue control-level commands (e.g., RESET, SHUTDOWN) to high-privilege nodes (e.g., gcs), violating access policies.
  - *Example: iot → gcs, command = SHUTDOWN.*

Classifier Output and Logging

Classification Workflow

- The final [CLS] embedding from **DistilBERT** is passed to a **softmax-based classification head**, producing one of four threat labels that represent operational threat types in Satellite–IoT systems.

Process Summary

- **[CLS] → Softmax:** Computes the most probable threat class from semantic embeddings.
- **Label Assignment:** Classifies as *Normal, Injection, Replay,* or *Privilege Abuse*.
- **Logging:** Appends the predicted label with timestamp and metadata for forensic analysis.
- **Modularity:** The classifier is lightweight and can be retrained as new labels emerge.

Interpretation

- This stage consolidates semantic reasoning into an interpretable classification outcome, forming the bridge between **sentence-level inference** and **system-level event analysis** for situational awareness in Satellite–IoT networks.

4

EXPERIMENTAL RESULTS AND DISCUSSION

Dataset Construction

❖ Dataset & Setup

- **Dataset:** 15-field Satellite–IoT packet dataset (25 K training / 10 K testing).
- **Labels:** *Normal, Injection, Replay, Privilege Abuse.*
- **Sources:** Derived from **CSP, MIOTY, CCSDS, TON-IoT** under unified protocol constraints.
- **Model:** *DistilBERT-base-uncased,* fine-tuned for four-class sentence classification.
- **Training:** AdamW (5×10^{-5} LR), batch 16, 10 epochs, weighted cross-entropy.
- **Metrics:** Accuracy & F1-score (average of five runs).

❖ Test Scenarios

- Evaluation performed on **fully structured packets (5 K)** and **missing-field packets (5 K)** to assess detection robustness.
- Each experiment was repeated five times with fixed train/test splits for consistent evaluation.

❖ Robustness Evaluation Strategy

- Random **2–5 fields per packet** were removed to simulate incomplete telemetry and corrupted payloads.
- DistilBERT maintained **stable accuracy and semantic consistency** under these incomplete-input conditions.

Computational Efficiency and Edge Feasibility

❖ Experimental Setup

- **Hardware:** Intel Core i7-11700 CPU / 64 GB RAM / NVIDIA RTX 3060 (12 GB VRAM).
- **Framework:** Python 3.11 with PyTorch and Hugging Face Transformers.
- **Training:** 10 epochs, batch size = 16, sequence length = 128.
- **Optimizer:** AdamW with weight-decay regularization.

❖ Resource Utilization

- **Training memory:** ≈ 789 MB (including optimizer states)
- **Inference latency:** ≈ 26 ms per packet
- **Model size:** ≈ 66 M parameters (≈ 250–300 MB)
- **Edge devices:** Runs smoothly on Jetson Nano and Raspberry Pi 4B (8 GB RAM)

❖ Performance Interpretation

- **DistilBERT** achieves **real-time inference** with approximately **75 % lower runtime memory** than **BERT-base**, while maintaining **comparable accuracy**.
- **Training** is conducted **offline**, and only **fine-tuned weights** are deployed on **edge devices**.
- The **lightweight architecture** enables **practical, near real-time anomaly detection** for **resource-limited Satellite–IoT environments**.

Performance Comparison on Satellite–IoT Packets

TABLE II
PERFORMANCE COMPARISON OF DETECTION MODELS ON SATELLITE–IOT PACKET CLASSIFICATION.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Snort	48.1	48.0	39.0	34.0
Random Forest	87.6	88.0	87.0	87.0
LSTM	92.8	95.0	91.0	92.3
DistilBERT	99.0	99.0	99.0	98.9

❖ Detection Results and Semantic Insights

- **DistilBERT** captures **cross-field semantic dependencies** that rule-based and traditional ML models cannot learn.
- Detects **contextual anomalies** where individual fields appear valid but their combinations are inconsistent.
- Maintains **high precision–recall consistency**, showing strong generalization across diverse packet structures.
- Confirms the **effectiveness of semantic representations** for **protocol-level anomaly detection** in Satellite–IoT networks.

Scenario-based Robustness

❖ Observations

- **Fast Convergence:** Accuracy surpasses 90 % by epoch 3 under both normal and missing-field conditions.
- **Resilient to Missing Data:** Even with 2–5 fields removed, accuracy stays ≈ 78–80 %, showing strong robustness.
- **Stable Generalization:** Only ≈ 20 % performance gap under incomplete inputs demonstrates semantic retention.

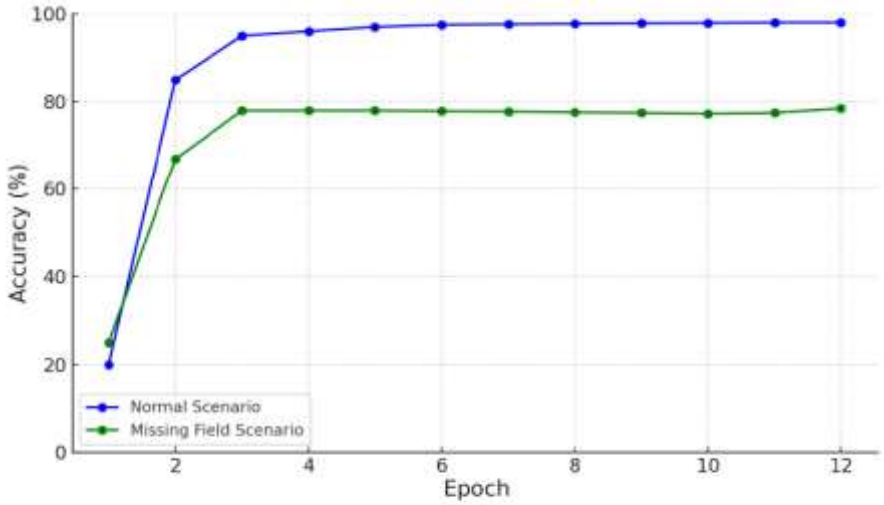

Fig. 2. Validation accuracy trends across epochs for normal and missing-field scenarios in Satellite–IoT packet classification.

❖ Interpretation

- **Contextual encoding** allows field-level redundancy — semantic tokens compensate for missing ones, ensuring reliable packet-level reasoning in lossy Satellite–IoT links.

Scenario-based Robustness

Observations

- **Rapid Convergence:** Both curves reach ≈ 99 % accuracy by epoch 3, confirming efficient fine-tuning.
- **No Overfitting:** Training and validation curves overlap closely, showing strong generalization.
- **Stable Optimization:** Accuracy variance stays within ± 0.5 % after epoch 4, indicating convergence stability.

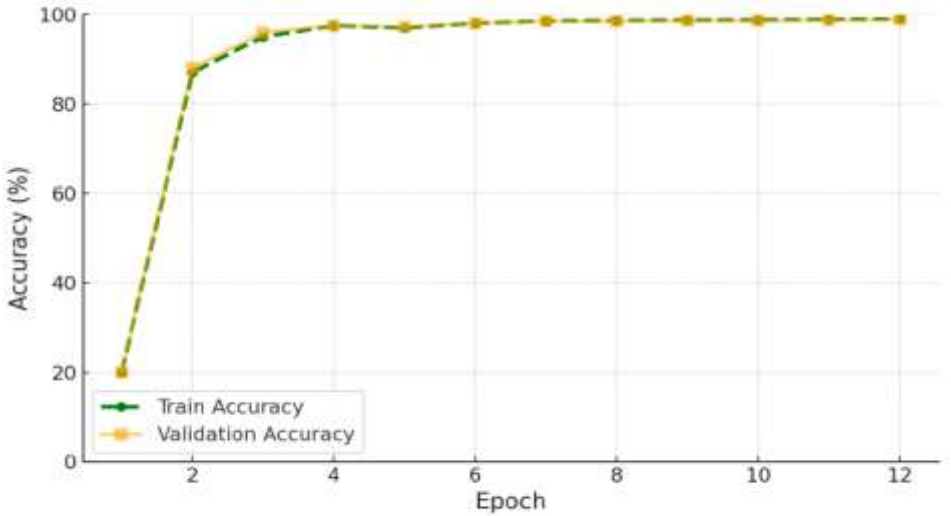

Fig. 3. Training and validation accuracy per epoch under normal packet conditions.

Interpretation

- The alignment between training and validation trends demonstrates stable semantic learning and absence of memorization bias.

Field-Level Attention Analysis

Field-Level Attention

- **flags** show the strongest attention across all classes → control-flag semantics are key anomaly indicators.
- **timestamp** gains higher weight in *Privilege Abuse* and *Injection*, reflecting sequence and timing misuse.
- **src / src_region** receive higher scores under *Injection*, highlighting spoofed-origin detection.

TABLE III
AVERAGE ATTENTION WEIGHTS PER FIELD ACROSS FOUR CLASSIFICATION LABELS

Field Name	Normal	Injection	Privilege Abuse	Replay
flags	**0.0231**	**0.0140**	0.0091	**0.0097**
timestamp	0.0060	0.0102	**0.0105**	**0.0088**
src	0.0080	**0.0103**	0.0098	0.0078
src_region	**0.0092**	0.0087	0.0091	0.0084
payload	0.0049	0.0080	0.0087	0.0083
priority	0.0037	0.0073	**0.0103**	0.0081
orbit_class	0.0049	0.0071	0.0091	0.0076
dst_region	0.0072	0.0069	0.0087	0.0078
src_port	0.0048	0.0070	0.0086	0.0078
msg_type	0.0074	0.0068	0.0082	0.0083
payload_type	0.0032	0.0052	0.0082	0.0079
dst_port	0.0033	0.0057	0.0083	0.0075
ttl	0.0027	0.0050	0.0084	0.0066
dst	0.0025	0.0052	0.0086	0.0062

Semantic Differentiation

- Each attack class triggers a unique attention pattern, proving DistilBERT's context-aware field reasoning.
- Attention aligns with real protocol logic (flags ↔ control, timestamp ↔ replay), confirming transparent and explainable inference.

Experimental Summary & Contributions

❖ Performance Summary

- **99 % accuracy** and **98.9 % F1**, outperforming baseline IDS models.
- Maintained ≈ **80 % accuracy** under missing-field conditions → robust to lossy telemetry.
- **26 ms latency** / **250–300 MB** footprint → real-time feasibility on Satellite–IoT gateways.

❖ Interpretation of Results

- **Semantic encoding > feature-based** methods — captures cross-field dependencies unseen by classical IDS.
- **Attention focus on *flags* and *timestamp*** aligns with actual protocol logic.
- Stable **Precision/Recall ≈ 99 %**, ensuring reliable operational detection.

❖ Main Contributions

- **Proposed sentence-based packet representation** enabling **contextual anomaly reasoning**.
- **Utilized lightweight DistilBERT**, achieving **near-BERT accuracy** with **~75 % lower memory**.
- **Demonstrated interpretable and robust detection**, establishing **groundwork for future temporal-sequence analysis**.

7

Conclusion and Future Work

Conclusion

Summary of Proposed Approach

- **Developed a lightweight semantic anomaly detection approach** for Satellite–IoT networks.
- **Integrated protocol-level semantics** through sentence-based packet representation.

Overall Contribution

- **Demonstrated that semantic encoding enables accurate, explainable, and real-time anomaly detection.**
- **Validated deployability on resource-limited gateways**, bridging model design with operational needs.
- **Established a foundation for mission-aware, context-driven security** in next-generation Satellite–IoT systems.

As shown in previous results, the model **maintained high accuracy and robustness under incomplete telemetry.**

Future Work

❖ Limitation & Temporal Context

- **Limitation:** The current approach operates on **single-packet inference**, without modeling **temporal dependencies** across sequential packets.
- **Future Direction:** Extend the framework to **multi-packet and time-series analysis** to incorporate **session-level context** and achieve **more reliable anomaly detection** over time.

❖ Enhancing Interpretability & Scope

- Integrate with **hierarchical threat modeling** for **system-wide propagation analysis** across communication layers.
- Explore **TTP (Tactics, Techniques, and Procedures)**–based abstraction to connect detected anomalies with **adversarial behaviors and tactics**.

❖ Expanding Coverage & Validation

- Extend semantic reasoning to **cross-layer threats** (e.g., spoofing, signal manipulation, jamming) for **broader coverage** of Satellite–IoT environments.
- Validate **computational feasibility on real onboard processors**, confirming performance under **practical operational constraints**.

Thank you
for your attention.

Junbeom Park

Korea Aerospace University

jbpark@kau.kr