國立臺北科技大學
National Taipei University of Technology

資訊安全碩士學位學程

碩士學位論文

Master of Science in Information Security

Master Thesis

衛星地面站之 TLE 軌道參數欺騙攻擊研究

TLE Data Deception Attacks Against Satellite Ground Stations

研究生：陳雋洋

Researcher: Chun-Yang Chen

指導教授：陳香君博士

Advisor: Shiang-Jiun Chen, Ph.D.

February 2026

國立臺北科技大學
研究所碩士學位論文口試委員會審定書

本校 資訊安全碩士學位學程 研究所 陳雋洋 君

所提論文，經本委員會審定通過，合於碩士資格，特此證明。

學位考試委員會

委　　員：馬 旻 蕙

吳 和 庭

陳 香 君

指導教授：陳香君

所　　長：

中 華 民 國 115 年 1 月 14 日

Abstract

Keywords: Space Situational Awareness (SSA), Two-Line Element (TLE), Starlink, Data Deception Attack, Ground Station Security, Unsupervised Learning

Satellite ground stations rely on Two-Line Element (TLE) data for antenna pointing and orbit propagation, typically operating under a presumption of data integrity. Consequently, most systems lack mechanisms to verify the physical validity of ingested ephemerides. This implicit trust creates a critical vulnerability, allowing adversaries to inject manipulated parameters via Man-in-the-Middle (MitM) attacks without triggering conventional alerts.

This study examines a stealthy forged-into-nominal attack scenario, where an adversary replaces physically anomalous orbital elements with counterfeit nominal values. By maintaining statistical consistency with historical records while violating orbital continuity, the attacker generates phantom targets. These manipulations induce the ground station to compute plausible but deceptive tracking solutions, silently misaligning with the true satellite trajectory and risking catastrophic Loss of Signal (LOS).

To counter this threat, we propose the Orbital Deception Defense Service (ODDS), a framework enforcing both physical and statistical consistency beyond single-threshold anomaly detection. ODDS employs a three-stage unsupervised learning architecture: Gaussian Mixture Models (GMM) first establish a global dynamic baseline from SGP4-propagated orbital behavior; then Isolation Forest (IF) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) detect subtle local anomalies and categorize deviation patterns.

Experimental evaluation using 8,500 Starlink TLE records over three months, augmented with physically plausible adversarial forgeries, shows that ODDS achieves a 77.5% defense rate against high-fidelity attacks. These results confirm that combining temporal-kinematic consistency checks with machine learning substantially increases the difficulty of injecting masked forgeries, establishing ODDS as an effective second line of defense for secure space situational awareness (SSA).

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

With the rapid expansion of the commercial space industry, Low Earth Orbit (LEO) satellite constellations have become a vital part of the global communication infrastructure [1]. Taking SpaceX's Starlink constellation as an example, as of October 2025, more than 8,000 satellites have been deployed, with plans to eventually reach 42,000. This unprecedented scale provides global low-latency, high-coverage communication services, yet it poses significant challenges to Space Situational Awareness (SSA) systems [2]. For satellite ground stations, accurate antenna pointing and communication link establishment critically depend on reliable orbital state information, making the integrity of orbital data a fundamental security requirement.

Currently, the international standard for describing satellite orbital states [3] is the Two-Line Element (TLE) format, which contains key parameters such as inclination, semi-major axis, eccentricity, and the right ascension of the ascending node (RAAN). Ground stations rely heavily on TLE data stored in their repositories to compute antenna pointing angles and establish communication links. However, due to modeling uncertainties and orbital maneuvers, TLE-based propagation inherently tolerates short-term deviations in parameters such as inclination and Right Ascension of the Ascending Node (RAAN), creating a trust gap between reported orbital states and actual physical motion. Ground stations typically presume TLE validity and lack mechanisms to verify the physical correctness of ingested ephemeris; this blind trust creates a critical ground station security vulnerability, where manipulated orbital parameters are processed as legitimate navigation commands and can lead to physical misalignment and tracking failure without triggering conventional alerts [4, 5].

Deception attacks represent a critical threat vector in Cyber-Physical Systems (CPS), where adversaries manipulate sensor data or control signals to mislead system operators while remaining undetected [6]. In the context of satellite ground stations, these attacks exploit the inherent trust gap in TLE data distribution, allowing attackers to inject manipulated orbital parameters that appear legitimate but cause systematic misalignment between ground station tracking systems and actual satellite positions. The sophistication of modern deception attacks lies in their ability to maintain statistical consistency with historical data patterns while violating underlying

physical constraints [7]. In a stealthy attack scenario, an adversary may compromise a ground station's TLE repository and replace physically anomalous orbital elements with counterfeit nominal values. Although the manipulated TLE stream remains statistically consistent with historical data, it violates physical continuity, causing the ground station to generate valid-looking tracking solutions while silently losing alignment with the true satellite trajectory—effectively creating a phantom target that misleads operational systems.

For satellite ground stations, these challenges are further exacerbated in the era of mega-constellations, where TLE data deception attacks confront multifaceted difficulties rooted in data scale, etiological complexity, and model adaptability [8]. Ground stations must process massive volumes of high-velocity, multi-source TLE data streams, while deception attack patterns themselves exhibit complex, abrupt, and spatio-temporally correlated characteristics that can evade traditional detection [9]. The optimal design of stealthy deception attacks aims to maximize the impact on system operations while minimizing the probability of detection, often by carefully crafting attack signals that blend seamlessly with normal operational noise [7]. Consequently, the etiology of inconsistencies in TLE data within ground station repositories is rarely singular, often involving a superposition of factors—such as legitimate satellite maneuvers, attitude drift, space weather disturbances, and malicious spoofing attacks—that renders any rule-based approach ineffective [10]. This complexity is compounded by the fact that deception attacks against networked control systems can be strategically designed to exploit specific vulnerabilities in data processing pipelines, making them particularly difficult to detect through conventional inconsistency detection methods [6]. A more fundamental challenge for ground station operators is the practical infeasibility of acquiring stable and comprehensive labels for TLE deception attacks. This scarcity of ground truth, compounded by TLE data distributions that shift continuously with time and mission parameters, undermines the long-term reliability of supervised or static models for ground station security [11]. The problem is particularly acute in high-dimensional TLE datasets processed by ground stations, where faint deception signals are easily obscured by noise [12]. In this complex and high-risk environment, traditional passive, point-solution defense paradigms for ground stations have become untenable [13].

Given these limitations, there is a critical need for a comprehensive defense solution that

can effectively protect satellite ground stations against TLE data deception attacks. To address this threat, this study investigates TLE data deception attacks against satellite ground stations and proposes the Orbital Deception Defense Service (ODDS) as a defense mechanism. ODDS serves as an independent verification layer for ground stations that enforces cross-domain consistency rather than relying on isolated inconsistency thresholds. Operating as a redundant verification layer independently of the primary TLE pipeline used by ground stations, ODDS remains effective even when the ground station's TLE repository is partially compromised. The system integrates orbital mechanics principles with inconsistency detection techniques, based on physical constraints such as the Simplified General Perturbation Model 4 (SGP4) [14], to identify inconsistencies between reported TLE data and physically plausible orbital evolution. By enforcing physical–statistical consistency, the framework provides a robust defense against TLE data deception attacks targeting ground stations, enhancing the operational resilience and security of satellite ground station operations.

Specifically, the overall architecture enables multi-level identification of TLE data deception attacks targeting ground stations through a cascaded pipeline that progressively refines inconsistency detection from global patterns to local outliers, and finally to systematic suspicious patterns. The pipeline operates through three sequential layers, each building upon the output of the previous stage, providing ground stations with a comprehensive defense mechanism against TLE spoofing.

In the first layer, Gaussian Mixture Models (GMM) [15] construct multi-modal statistical distributions of nominal orbital behaviors derived from SGP4 propagations, establishing a global baseline that captures the natural variation across different orbital planes. Samples that significantly deviate from these learned distributions are flagged as initial suspicious candidates, indicating potential inconsistencies that warrant further investigation.

Building upon the GMM's global perspective, the second layer employs Isolation Forest (IF) [16] to analyze these candidates in the high-dimensional feature space, detecting subtle local outliers that may have been overlooked by the global model. This complementary approach enhances the system's sensitivity to boundary inconsistencies that lie near the decision boundaries of normal orbital patterns, identifying TLE entries that may be statistically isolated but not

necessarily anomalous.

The suspicious candidates identified by the first two layers are then passed to the third layer, where Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [17] performs density-based clustering analysis. This final stage effectively distinguishes genuine, systematic inconsistency patterns from isolated random noise, filtering out spurious detections that lack meaningful patterns while recognizing that isolated TLE entries may represent legitimate but statistically uncommon orbital states.

This cascaded, precision-oriented design prioritizes operational reliability over broad coverage, ensuring that detected inconsistencies are highly credible and actionable. By incorporating SGP4-based physical constraints throughout the pipeline, the framework achieves superior precision while minimizing false positives, recognizing that isolated TLE entries may be legitimate but statistically uncommon, rather than necessarily indicating deception attacks. This makes the system suitable for automated defense systems where false alarms can be costly. Moreover, the trained GMM model from the first layer is further leveraged to generate high-fidelity synthetic TLE samples that preserve the orbital distribution characteristics of real Starlink satellites—providing valuable resources for system testing, model benchmarking, and security research in the SSA domain.

To evaluate the effectiveness of this three-tier architecture in defending ground stations against TLE data deception attacks, comprehensive experiments were conducted on real-world Starlink [18] orbital data. The experimental results demonstrate that the proposed system achieves exceptional performance when analyzing approximately 8,500 daily Starlink orbital records from September to December 2025, simulating the TLE data streams that ground stations typically process.

The main contributions of this work can be summarized as follows:

- TLE Data Deception Attack Detection: The proposed ODDS framework enforces cross-domain consistency between statistical patterns and physical orbital evolution, enabling reliable detection of TLE data deception attacks targeting ground stations. The system can identify attacks that remain statistically consistent with historical data but violate physical continuity, which is critical for protecting ground station operations. This performance

ensures that detected deception attempts are highly credible, making the system suitable for automated ground station defense systems where false alarms can lead to operational disruptions and service interruptions.

- Ground Station Security Enhancement: Operating as an independent verification layer, ODDS provides operational resilience for satellite ground stations by maintaining effectiveness even when primary TLE data sources are partially compromised. The trained GMM model was leveraged to generate synthetic adversarial samples, validating the system's robustness against sophisticated TLE spoofing attacks. This capability demonstrates the framework's effectiveness in enhancing ground station security and supporting secure space situational awareness operations.

The remainder of this thesis is organized as follows. Chapter 2 reviews related literature and existing approaches. Chapter 3 presents the system architecture and methodology. Chapter 4 describes the experimental setup and implementation. Chapter 5 discusses the results and analysis. Finally, Chapter 6 summarizes the contributions and outlines future research directions.

Chapter 2 Related Work

This chapter reviews the main technologies and literature relevant to this research. First, we introduce the TLE data format used to describe satellite orbital information. Next, we discuss the SGP4 and its applications in orbit prediction. Subsequently, we explore the importance of SSA in satellite monitoring and integrity verification. We then review three unsupervised deception-defense methods—GMM, IF, and HDBSCAN—as the foundation for the proposed approach. Finally, we examine advanced orbit prediction techniques and security challenges in modern space operations.

2.1 Two-Line Element (TLE)

TLE Data Format constitute a standardized set of orbital elements used to describe Earth-orbiting spacecraft, with origins tracing back to the orbital tracking system established by the North American Aerospace Defense Command (NORAD) in the 1960s [19]. To date, TLE remain the most widely used data format for global space surveillance and satellite tracking missions. Research by Guo et al. [3] highlights that TLEs provide a critical data foundation for satellite orbit analysis and time-series modeling. Each TLE record comprises three lines of information: the first line contains the satellite's name, while the second and third lines record orbital parameters such as satellite number, epoch time, inclination, eccentricity, and mean motion. These parameters enable orbit propagation through SGP4/SDP4 models to compute a satellite's position and velocity at any given moment.

As the number of LEO satellites rapidly grows, the application scope and processing demands for TLE data continue to expand. Kazemi et al. [20] highlight that modern SSA systems must process tens of thousands of TLE records, making automated monitoring and integrity verification essential. Noetzold et al. [21] also confirm that integrating machine learning techniques can enhance TLE data observability and deception-defense performance.

However, TLE data still faces two major challenges: quality and security. Regarding data quality, common issues include measurement noise, orbit propagation errors, and inconsistent

data update frequencies. On the security front, malicious actors may attempt interference by forging or tampering with TLE data. Graczyk et al. [4] proposed the EphemeriShield system to provide defensive mechanisms against cyber anti-satellite attacks, while Wigchert et al. [5] utilized deep learning to detect spoofing attacks in LEO satellite systems. Overall, TLEs are not only crucial data sources for orbit prediction and space surveillance but are also increasingly central to research in deception defense and space asset protection.

2.2 Simplified General Perturbations 4 (SGP4)

SGP4 is a standard orbit propagation model developed by the U.S. Space Command, specifically designed to process TLE-formatted orbital data [14]. This model, along with its deep-space counterpart SDP4 (Simplified Deep-space Perturbations 4), forms the core computational engine of modern space surveillance systems. SGP4 is primarily applicable to low Earth orbit satellites (orbital period less than 225 minutes), while SDP4 is optimized for deep-space satellites (orbital period greater than 225 minutes) [22].

However, traditional SGP4 models have accuracy limitations, especially in long-term predictions where errors accumulate over time. Wildt et al. [23] conducted a comparative analysis of orbit propagation accuracy, finding that prediction accuracy significantly decreases with increasing propagation time, particularly for near-Earth objects where accuracy is on the order of 100 meters, but errors increase substantially in long-term propagation. To address this, academia has proposed various improvements: Levit et al. [24] improved TLE prediction accuracy through batch least-squares differential correction, achieving a daily prediction error growth rate of approximately 100 meters; Liu et al. [25] proposed a hybrid algorithm combining the simplex method with genetic algorithms to enhance orbit determination accuracy, achieving a 40.25% improvement in 10-day prediction accuracy.

Driven by advancements in deep learning, Acciarini et al. [26] in 2025 proposed an innovative differentiable SGP4 model (∂SGP4), porting SGP4 to a PyTorch-based differentiable framework. This allows it to perform both forward orbit propagation and backward gradient computation simultaneously. This differentiability enables seamless integration of SGP4

with gradient-based methods, including orbit determination, covariance propagation, and various gradient-based optimization applications.

To enable gradient-based learning and parameter refinement, the differentiable SGP4 model reformulates orbit propagation into an optimization problem. The goal is to adjust model parameters such that the propagated orbit closely matches a high-precision numerical reference. This optimization framework is formally defined in Eq. (2.1):

$$\mathcal{P}: \begin{cases} \text{given:} & \vec{x}_0, t \\ \text{find:} & \vec{\theta}_1, \vec{\theta}_{\text{SGP4}}, \vec{\theta}_2 \\ \text{s.t. min:} & J(\vec{x}(\vec{x}_0, t, \vec{\theta}_1, \vec{\theta}_{\text{SGP4}}, \vec{\theta}_2)) = \|\vec{x} - \vec{x}_{\text{HPOP}}\|^2 \end{cases} \tag{2.1}$$

Eq. (2.1) describes the mathematical formulation of the optimization problem $\mathcal{P}$, where $\vec{x}_0$ represents the initial TLE orbital elements and t is the propagation time. The objective is to find the optimal set of parameters $\vec{\theta} = \{\vec{\theta}_1, \vec{\theta}_{\text{SGP4}}, \vec{\theta}_2\}$ that minimize the cost function J, which measures the squared Euclidean distance between the predicted state $\vec{x}$ and the high-precision numerical reference $\vec{x}_{\text{HPOP}}$. Through differentiability, all parameters can be updated based on gradients: $\vec{\theta} \leftarrow \vec{\theta} - \alpha \nabla_{\vec{\theta}} J$. This architecture significantly enhances propagation accuracy while maintaining SGP4's computational efficiency, laying the technical foundation for combining deep learning with orbit integrity verification and maneuver prediction. In terms of integrity verification applications, the SGP4 model provides an important foundation for orbit maneuver detection. Mukundan et al.[27] proposed a simplified maneuver detection method that identifies satellite maneuvers by comparing Keplerian orbital elements obtained from TLEs with those propagated by a simplified perturbation model, capable of detecting maneuvers as small as centimeters per second.

Traditional orbit integrity verification methods often combine physical model validation, such as checking if orbital parameters conform to Kepler's laws [28], or identifying forgeries through orbit propagation residuals. With the development of machine learning techniques, Dráček et al.[29] utilized time series analysis methods to detect manipulated samples from TLE data, providing new solutions to overcome the limitations of traditional SGP4 methods.

2.3 Space Situational Awareness (SSA)

SSA is the systematic capability to monitor and track objects in Earth orbit, crucial for maintaining space environment safety. Kennewell et al. [2] provide a comprehensive overview of SSA, pointing out its core objectives include real-time accurate provision of space object position information and collision risk prediction. Kazemi et al. [20] systematically reviewed orbit determination methods in the SSA domain, emphasizing that the demands for real-time performance and accuracy have reached unprecedented levels in the era of large constellations. Zhang et al. [30] provided a comprehensive review of the development, impact, and monitoring of large LEO constellations.

Addressing the technical challenges posed by large constellations, traditional monitoring methods are often insufficient. Massimi et al. [31] systematically investigated the application of deep learning methods in SSA for large constellations, confirming their significant advantages in space object detection and classification.

In terms of orbit prediction techniques, Lang et al. [18] proposed an unscented batch filtering method for continuously maneuvering Starlink satellites, achieving a Root Mean Square Error (RMSE) of less than 3 kilometers in 24-hour predictions. Wu et al. [32] and Tang et al. [33] explored LSTM-based and federated learning-based orbit prediction techniques, respectively, confirming that machine learning methods can effectively improve prediction accuracy.

In the field of integrity verification, Diro et al. [34] systematically investigated deception-defense techniques in space information networks, identifying key challenges such as scalability, real-time detection, and limited labeled data, and validating the effectiveness of deep learning methods. This provides an important reference for the deception-defense architecture design of this study.

2.4 Deception Attack

Deception attacks constitute a critical threat in CPS by manipulating system data to mislead operators while preserving the appearance of normal operation [6]. Unlike denial-of-service

or jamming attacks, which cause observable disruptions, deception attacks aim to remain statistically inconspicuous while gradually steering the system toward unsafe states. Pang and Liu [35] demonstrated that such attacks can exploit feedback and communication structures in networked control systems to degrade system performance without triggering conventional alarms. Extending this observation to more complex environments, Ding et al. [36] showed that deception-induced biases in stochastic nonlinear systems can closely mimic natural process noise, further complicating detection using traditional security mechanisms.

Theoretical foundations for these threats were formalized by Zhang et al. [7], who introduced the concept of optimal stealthy deception attacks. Their work demonstrated that an adversary can maximize system state deviation while constraining statistical detectability, commonly measured via Kullback–Leibler (K–L) divergence. This result implies that manipulated data can remain statistically indistinguishable from nominal noise, rendering threshold-based or distribution-based detectors ineffective.

Within the SSA domain, these risks are amplified by the heavy reliance on centralized and often unauthenticated orbital data repositories. Most satellite operators lack the observational infrastructure to independently acquire high-quality SSA measurements and therefore depend on a small number of state-operated catalogs for conjunction assessment and orbit maintenance, creating an asymmetric trust relationship. Pavur and Martinovic [37] examined how a dominant SSA provider could exploit this position by manipulating TLE data to alter the perceived identity or behavior of space objects—most notably by disguising active reconnaissance satellites as space debris while still nominally "cooperating" in collision-avoidance data sharing. Building on historical deception cases and real-world catalog data, they simulated such attacks and proposed a machine-learning-based deception detector that operates solely on publicly available TLEs. Their system was able to identify 90 – 98% of simulated deception attempts, including real-world cases such as the 2014-28E incident, demonstrating that even operators without dedicated astrometric hardware can enhance their defenses through independent data-integrity verification.

Synthesizing these insights, an attacker can exploit both optimal stealth principles and the trust-based nature of SSA data distribution to inject counterfeit nominal TLEs that appear fully legitimate to standard filters. Consequently, a ground station may track a phantom target that is

mathematically consistent with historical data yet physically disconnected from the true satellite trajectory. This observation motivates the need for detection mechanisms that move beyond statistical plausibility toward cross-layer physical – statistical consistency, as pursued in this work.

2.5 Unsupervised Pattern Recognition and Filtering

In satellite monitoring and space situational awareness, unsupervised deception defense has become a major research focus due to the scarcity of labeled forgeries. This study employs three complementary techniques—GMM, IF, and HDBSCAN—to build a multi-layer deception-defense framework based on statistical modeling, isolation analysis, and density-based clustering.

2.5.1 Gaussian Mixture Model (GMM)

GMM is a probability distribution-based unsupervised learning method that characterizes the multimodal nature of data through a weighted combination of multiple Gaussian distributions. It has long been widely applied in pattern recognition and integrity verification [15]. With the advancement of machine learning technology, academia continues to improve GMM's modeling capabilities and temporal adaptability. For example, Olivier et al. [38] proposed a Time-Constrained GMM (TCGMM) to enhance clustering performance for multimodal time-series data; Zong et al. [39] combined deep autoencoders with GMM to construct an end-to-end unsupervised deception-defense framework, demonstrating superior fitting capabilities for high-dimensional data distributions.

In selecting unsupervised deception-defense methods, the applicability and limitations of each approach were considered. MacQueen et al. [40] noted that K-means assumes spherical clusters and requires a predefined number of clusters, making it highly sensitive to initial values and scale. Ester et al. [41] indicated that DBSCAN can automatically identify arbitrary-shaped clusters and noise points; however, its stability decreases with significant density variations, and it is sensitive to parameter selection. Zimek et al. [12] conducted a systematic survey on

forgery detection in high-dimensional data, identifying the concentration effect of distances and interference from irrelevant attributes as major challenges affecting detection performance. In contrast, GMM can model multimodal distributions generatively and quantify the consistency of samples with the overall distribution through log-likelihood values, providing comparable forgery scores [42].

In space applications, GMM has been widely used for statistical modeling of satellite operational data. Wei et al. [43] applied it to integrity verification in satellite power systems, while Adam et al. [44] validated its robustness as a statistical baseline model using LEO satellite battery telemetry data. Particularly in the field of TLE maneuver detection, due to factors such as space object collisions, environmental disturbances, and mismatches, forgeries or manipulated samples in TLE data inevitably affect detection accuracy. To address this, Zhang et al. [45] proposed a Robust Gaussian Mixture Model (RGMM), combining robust estimation theory to handle forged or manipulated TLE data.

The core improvement of RGMM lies in robustifying the Expectation-Maximization (EM) algorithm. The iterative process of the EM algorithm includes the E-step (computing responsibilities) and the M-step (estimating parameters). First, a forgery level factor w is introduced to constrain the proportion of Gaussian components in the mixture model. Accordingly, the probability density function of RGMM is defined as shown in Eq. (2.2), where the last Gaussian component is dedicated to modeling forgery behavior:

$$P(x|\theta) = \sum_{k=1}^{K-1} \alpha_k f(x, \mu_k, \sigma_k^2) + w f(x, \mu_K, \sigma_K^2), \quad \sum_{k=1}^{K-1} \alpha_k + w = 1 \tag{2.2}$$

Eq. (2.2) defines the probability density function of the robust mixture model. In this formulation, α_k represents the mixing weight of the k-th Gaussian component, while w is the weight assigned to the forgery component, ensuring that the total sum of weights equals unity.

Second, in the E-step, a robust correction function $L(k)$ is introduced to improve responsibility calculation, thereby reducing the influence of manipulated samples on parameter estimation. Based on this correction mechanism, the posterior responsibility of each sample is computed according to Eq. (2.3):

$$\gamma_{ik} = \frac{\alpha_k f(x_i, \mu_k, \sigma_k^2)}{P(x_i|\theta)} \cdot L(k) \tag{2.3}$$

Where the robust correction function is defined as shown in Eq. (2.4):

$$L(k) = \begin{cases} \frac{P(x_i|\theta)}{\sum_{k=1}^{K-1} \alpha_k f(x_i, \mu_k, \sigma_k^2)} \cdot p_i & k < K \\ \frac{P(x_i|\theta)}{w f(x_i, \mu_K, \sigma_K^2)} \cdot (1 - p_i) & k = K \end{cases} \tag{2.4}$$

In the Eq. (2.5), p_i is the robust factor. To determine this factor, the Institute of Geodesy and Geophysics (I (commonly referred to as IGGIII) is adopted. This scheme classifies samples into three categories—normal, suspicious, and forgery segments—based on their precision range and occurrence probability. Compared to methods like Huber and Danish, the IGGIII scheme can handle all three types of samples simultaneously:

$$p_i = \begin{cases} p_0 & v_i < c_0 \\ p_0 \left(\frac{c_1 - v_i}{c_1 - c_0} \right)^2 & c_0 \leq v_i < c_1 \\ 0 & c_1 \leq v_i \end{cases} \tag{2.5}$$

The normalized residual v_i and the normalization constraint between the robustness factor and the forgery weight are defined in Eq. (2.6):

$$\begin{cases} v_i = \frac{d_{ik}}{\sigma_k} \\ p_0 + w = 1 \end{cases} \tag{2.6}$$

As specified in Eq. (2.6), v_i represents the standardized distance of the i-th sample, where d_{ik} denotes the distance between the i-th sample point and the optimal Gaussian component center μ_k, and σ_k is the standard deviation (the arithmetic square root of the variance σ_k^2). The constant thresholds c_0 and c_1 generally range from 1 to 1.5 and 2.5 to 8, respectively. Furthermore, p_0 represents the proportion of normal samples in the overall GMM, with its value inversely determined by the forgery level factor w.

In the M-step, model parameters $\{\alpha_k, \mu_k, \sigma_k^2\}$ are estimated by maximizing the likelihood

function. After iterative convergence, the K-th forgery component is ignored, and the proportion coefficients of the first $K-1$ components are scaled proportionally. Finally, maneuver detection is performed using a 95% probability threshold. Experimental results show that RGMM significantly improves recall while maintaining precision compared to traditional GMM, demonstrating its potential in handling TLE data quality issues.

Based on GMM's superior modeling capabilities and robust properties, this study employs it as the first-layer global statistical baseline to capture multi-shell and multimodal orbital and telemetry structures, and outputs deception candidates for further analysis by subsequent IF and HDBSCAN layers.

2.5.2 Isolation Forest (IF)

IF proposed by Liu et al. [16] in 2008, is based on the core assumption that "manipulated samples are easier to isolate". It constructs multiple isolation trees by randomly partitioning the feature space, using the path length of a sample as a forgery indicator. The innovation of this method lies in detecting forgeries purely based on the isolation concept, without relying on any distance or density measurements, fundamentally differentiating it from all existing methods. Liu et al. [46] further demonstrated that IF can achieve linear time complexity and small memory requirements through sub-sampling, and effectively handle masking and swamping effects. When dealing with high-dimensional problems containing numerous irrelevant attributes, IF remains robust, operating effectively even when training samples do not contain forgeries.

As deception-defense demands diversify, IF methods continue to evolve and expand. Cao et al. [47] conducted a systematic review of isolation-based deception-defense methods, highlighting that IF and its variants demonstrate excellent performance across various application domains, including high-dimensional data, time series, and stream data scenarios. Building on this, Xu et al. [48] in 2023 proposed Deep IF, combining deep learning with isolation mechanisms to further extend IF's representation learning capabilities in complex high-dimensional data, enabling it to more effectively handle nonlinear relationships and latent features.

In the field of space situational awareness, IF has been proven effective for various satellite system monitoring tasks. Schefels et al. [49] researched pattern detection in time-series satellite

telemetry data, demonstrating that isolation analysis can effectively identify temporal forgery patterns, providing a new technical path for satellite health monitoring. Wang et al. [50] further combined deep learning with IF, proposing a deception-defense framework for satellite telemetry data, enhancing the model's generalization ability through a fake-forgery training strategy. In practice, Li et al. [51] applied IF to real-time Precise Point Positioning (PPP) service quality monitoring, achieving over 95% detection success for orbital deviations greater than 5 cm and clock deviations greater than 0.2 nanoseconds through IF residual integrity verification, and effectively identifying orbital and clock jumps in real-time GNSS products, demonstrating IF's practicality and robustness in high-precision satellite applications.

In summary, IF's core advantages include high computational efficiency, independence from data distribution assumptions, and the ability to capture local isolation forgeries, complementing GMM's global statistical modeling. Therefore, this study adopts IF as the second-layer deception defense to filter deception candidates from GMM for subsequent HDBSCAN analysis.

2.5.3 Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

HDBSCAN, proposed by Campello et al. [17] in 2013, is a hierarchical extension of DBSCAN designed to overcome the reliance of traditional density-based clustering methods on a single density threshold. This approach automatically identifies clusters corresponding to various density thresholds and labels samples that do not belong to any cluster as noise, producing a complete hierarchical density clustering structure.

Algorithm 1: HDBSCAN main steps

1. Compute the core distance w.r.t. $mpts$ for all data objects in X;
2. Compute an MST of G_{mpts}, the Mutual Reachability Graph;
3. Extend the MST to obtain MST_{ext}, by adding for each vertex a "self edge" with the core distance of the corresponding object as weight;
4. Extract the HDBSCAN hierarchy as a dendrogram from MST_{ext}:;
   4.1 For the root of the tree assign all objects the same label (single "cluster");
   4.2 Iteratively remove all edges from MST_{ext} in decreasing order of weights (in case of ties, edges must be removed simultaneously):;
      4.2.1 Before each removal, set the dendrogram scale value of the current hierarchical level as the weight of the edge(s) to be removed;
      4.2.2 After each removal, assign labels to the connected component(s) that contain(s) the end vertex(-ices) of the removed edge(s), to obtain the next hierarchical level: assign a new cluster label to a component if it still has at least one edge, else assign it a null label ("noise");

Algorithm 1 illustrates the execution flow of HDBSCAN. The algorithm constructs a minimum spanning tree (MST) of the mutual reachability graph using core distances, extends it to form MST_{ext} with self-loops, and then removes edges in decreasing order of weights to generate a hierarchy. This process adaptively identifies clusters and noise at different density scales. The core innovation of HDBSCAN lies in providing a complete hierarchical density clustering structure that covers all DBSCAN-like solutions under an infinite range of density thresholds, enabling adaptive handling of different density levels through optimal local cuts. The algorithm's time complexity is $O(dn^2)$ and space complexity is $O(dn)$ (or $O(n^2)$ and $O(n^2)$ respectively when using a distance matrix). Campello et al. [52] further proposed an integrated framework introducing the concept of cluster stability, which provides a globally optimal solution by maximizing overall stability and maintains stable clustering in data with significant density variations.

HDBSCAN has demonstrated effectiveness and robustness across multiple domains. Tran

et al. [53] implemented HDBSCAN on the Apache Spark distributed platform, successfully handling large-scale datasets and demonstrating good scalability. Their work confirms that hierarchical clustering can effectively separate noise and identify data patterns. Logan et al. [54] applied HDBSCAN to unsupervised classification of stars, galaxies, and quasars, exhibiting superior classification accuracy over traditional methods when processing high-dimensional astronomical data. In trajectory analysis, Raj et al. [55] integrated HDBSCAN with LSTM for maritime traffic integrity verification, effectively improving collision prediction accuracy by handling trajectory clustering with varying densities. Similarly, Wang et al. [56] conducted clustering analysis on Automatic Identification System (AIS) vessel trajectories, outperforming DBSCAN and K-means in Silhouette Coefficient and Davies-Bouldin Index evaluations.

For TLE deception-defense tasks, HDBSCAN automatically identifies heterogeneous patterns without predefining cluster numbers and provides multi-level classification. This study adopts HDBSCAN as the third-layer tool to classify deception candidates from GMM and IF into maneuver events, inclination deviations, and other types.

2.6 Advanced Orbit Prediction and Security

Recent advancements in orbit prediction have shifted toward data-driven frameworks that leverage long-term historical orbital measurements to overcome the limitations of snapshot-based methods. Junyu et al. [57] demonstrated that fusing 10-day Starlink TLE sequences enables a more robust capture of temporal dynamics compared to single-epoch propagation. Their results showed that such multi-TLE modeling—enhanced by batch differential correction and high-precision propagation—reduces 5-day prediction errors by approximately 50% relative to standard SGP4, effectively mitigating drift and improving space object surveillance.

However, these performance gains fundamentally depend on the trustworthiness of the underlying space data. Wigchert et al. [58] exposed a critical vulnerability within LEO communication infrastructure: unauthenticated broadcast channels remain susceptible to adversarial spoofing. Through drone-based experimentation with rogue message injection, they empirically validated the feasibility of aerial attack vectors against active constellations. While emerging

PHY-layer detection techniques—such as autoencoder-based integrity verification on signal images—show promising detection performance, current operational systems largely lack such safeguards.

These findings highlight a dual trajectory in modern space operations: as predictive accuracy improves through richer data utilization, the growing security risks associated with data manipulation underline the urgent need for integrated deception defense and defensive mechanisms within spaceborne communication and navigation systems.

Chapter 3 Methodology

This chapter outlines the methodology for the TLE-based deception defense system. The framework consists of a Core Layer that integrates TLE data processing with the Orbital Deception Defense Service (ODDS), Event Management modules for alerts and logging, a Simulation module for synthetic data generation, and an Infrastructure Layer providing storage and computing resources. The subsequent sections detail the design and implementation of these components.

3.1 System Architecture

Figure 3.1 depicts a vertically layered, modular architecture. The core layer couples the TLE data-processing pipeline with the ODDS, whose submodules—a Normality Modeling Engine, Rare Event Detector, and Structural Pattern Analyzer—provide complementary detection capabilities. Event Management modules supply scenario generation, alerting, forecasting, and auditing. Concurrently, the infrastructure layer (PostgreSQL, Ubuntu 22.04) furnishes persistence and runtime support.

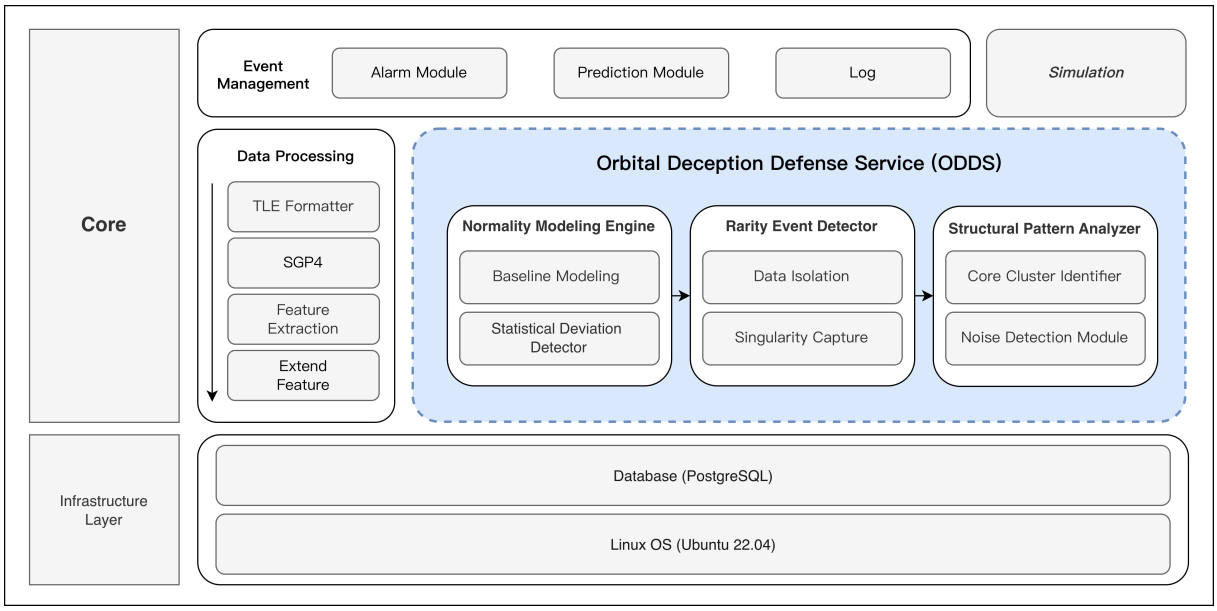

Figure 3.1: Ground Station TLE Deception Defense Architecture

3.1.1 Data Processing

The data processing module transforms raw TLE data, sequentially executing data acquisition, format standardization, orbital parameter estimation, and feature engineering steps, ensuring data quality and processing traceability. The details of each stage module are described as follows:

- TLE Formatter: This module employs a deterministic fixed-field parser to convert raw TLE strings into structured, machine-readable records with column-level accuracy. The TLE format consists of three components: a title line (Line 0) for the satellite name, followed by two standardized data lines, Line 1 and Line 2. The following Starlink TLE excerpt illustrates this structure:

```
STARLINK-1008
1 44714U 19074B 25280.37724496 .00005163 00000+0 36508-3 0 9999
2 44714 53.0493 154.8642 0001368 87.3282 272.7864 15.06415906 325647
```

  Line 1 encodes the catalog number (44714), international designator (19074B), epoch time (25280.37724496, i.e., day 280 of 2025 plus 0.37724496 days), the first derivative of the mean motion (5.163×10^{-5} rev/day^2), the second derivative (0.00000×10^{0} rev/day^3), and the B* drag coefficient (3.6508×10^{-4}). Line 2 provides the classical orbital elements: orbital inclination (53.0493°), right ascension of the ascending node or RAAN (154.8642°), eccentricity (0.0001368), argument of perigee (87.3282°), mean anomaly (272.7864°), mean motion (15.06415906 rev/day), and revolution number (325647). A comprehensive field-by-field interpretation of the TLE format, including all parameters and their physical meanings, is provided in Table A.1 and Table A.2 in Appendix A.

  Unlike flexible regex-based parsers—which are prone to misinterpreting irregular whitespace, merged fields, or missing delimiters commonly seen in high-value entries (e.g., mean motion > 100 rev/day)—the formatter enforces the CCSDS/CelesTrak column indices exactly. This design guarantees bit-wise repeatability: identical inputs always yield identical parsed outputs, providing robustness against common formatting glitches where

adjacent columns collapse together. The column organization is illustrated in Figures 3.2, 3.3, and 3.4, which show the fixed column positions for the satellite name line, Line 1, and Line 2, respectively. Each parameter is extracted from its prescribed column range; for example, in Line 2, the inclination (columns 9−16), RAAN (columns 18−25), and eccentricity (columns 27−33) are parsed directly from their fixed positions, as shown in Fig. 3.4.

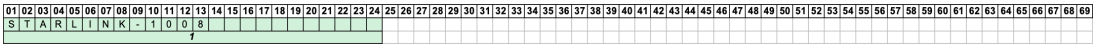

Figure 3.2: TLE Format: Satellite Name Line

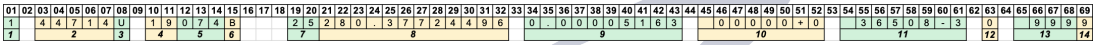

Figure 3.3: TLE Format: First Line Element

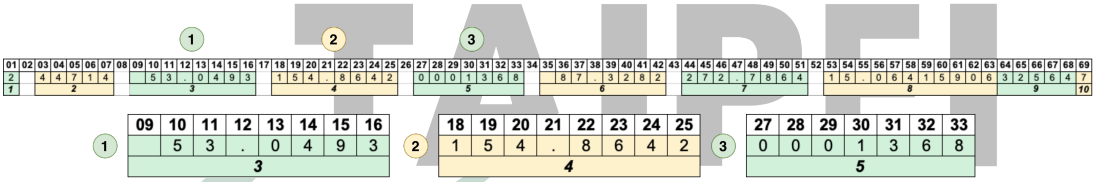

Figure 3.4: TLE Format: Second Line Element

The formatter performs structural normalization exclusively—type casting, restoration of implicit decimal points, epoch alignment to Coordinated Universal Time (UTC), and imputation of missing catalog metadata—without performing value validation or physical plausibility checks. This separation of concerns ensures that the formatter operates purely as a data transformation stage, preserving original TLE semantics while standardizing the representation. Subsequently, bounded-range validation checks are applied as an independent step—constraining inclination to $[0, 180°]$ and mean motion to $[10, 20]$ revolutions per day—to enable early detection of format corruption or forged/manipulated entries. Analytical and physical transformations are intentionally deferred to downstream modules, ensuring a clear separation between parsing, validation, and feature extraction stages.

- SGP4 Propagation: This module employs the SGP4 algorithm to estimate the satellite's three-dimensional position and velocity vectors from TLE parameters. The system first

converts the mean motion from the TLE (units: revolutions per day) to angular velocity (units: radians per second):

$$n = n_0 \times \frac{2\pi}{86400} \quad \text{(rad/s)} \tag{3.1}$$

where n_0 is the mean motion value from the TLE (rev/day), and $86,400$ is the number of seconds per day.

Subsequently, the system computes the semi-major axis from the angular velocity using Kepler's third law:

$$a = \left(\frac{\mu}{n^2}\right)^{1/3} \quad \text{(km)} \tag{3.2}$$

where $\mu = 398,600.4418$ is the Earth's standard gravitational parameter (units: km^3/s^2). The orbital period is then derived directly from the angular velocity:

$$T = \frac{2\pi}{n} \quad \text{(s)} = \frac{2\pi}{n \times 60} \quad \text{(min)} \tag{3.3}$$

Additionally, the system calculates the perigee and apogee radii:

$$r_p = a(1 - e) \quad \text{(perigee radius)} \tag{3.4}$$

$$r_a = a(1 + e) \quad \text{(apogee radius)} \tag{3.5}$$

where e is the eccentricity. The perigee and apogee altitudes are obtained by subtracting the Earth's radius $R_E = 6,378.137$ km from the respective radii. This step supplements the orbital dynamics information that cannot be directly obtained from the TLEs alone, providing complete orbital physical parameters for subsequent feature engineering.

- Feature Extraction: This stage reconstructs the classical orbital elements (COEs) derived from SGP4 propagation, including the semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω, and mean anomaly M. Together, these parameters provide a compact yet physically interpretable representation

of the orbital state.

Two families of secondary indicators are subsequently derived. First, the perigee and apogee altitudes are computed from the orbital radii via Eqs. (3.4)–(3.5) by subtracting the Earth's mean radius. Second, the orbital period is analyzed for physical consistency. The theoretical period (T_{theory}) is computed as:

$$T_{\text{theory}} = 2\pi\sqrt{\frac{a^3}{\mu}} \quad \text{(s)} \tag{3.6}$$

This theoretical value is then compared against the observed period (T_{obs})—derived from the TLE's mean motion—through the Kepler residual:

$$r_{\text{Kepler}} = |T_{\text{obs}} - T_{\text{theory}}| \quad \text{(min)} \tag{3.7}$$

which quantifies deviations from ideal Keplerian dynamics. Elevated residuals indicate possible measurement noise or impulsive maneuvers that perturb the orbit from its nominal behavior.

Samples that fall outside the nominal Starlink operational envelope are flagged by range-violation indicators, computed using Eq. (3.8):

$$\text{is\_out\_of\_range\_a} = \begin{cases} 1 & \text{if } a < a_{\text{min}} \text{ or } a > a_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where $a_{\text{min}} = 6,915$ km and $a_{\text{max}} = 6,945$ km correspond to the 540–570 km altitude band published for the Starlink Gen-1 53° shells (Earth radius $6,378$ km plus altitude). The same logic is consistently applied to other monitored parameters (e.g., period and inclination), ensuring that all data provided to downstream detection modules remain within validated operational ranges.

- Feature Extension: To amplify the forgery-discrimination power, the pipeline augments

the base COEs with five temporal-derivative features. Each derivative is estimated via the backward-difference operator in Eq. (3.9), which captures the instantaneous slope of the corresponding orbital parameter and thus reveals abrupt trends that static features cannot encode:

$$\left.\frac{dp}{dt}\right|_{t_i} = \frac{p(t_i) - p(t_{i-1})}{t_i - t_{i-1}} \quad \text{(units/day)} \tag{3.9}$$

where t_i denotes the timestamp of the i-th observation. For the RAAN, the wrap-around rule in Eq. (3.10) is applied to resolve the 360° discontinuity inherent to angular measurements:

$$\Delta\Omega = \begin{cases} \Omega_i - \Omega_{i-1} - 360^\circ & \text{if } \Omega_i - \Omega_{i-1} > 180^\circ \\ \Omega_i - \Omega_{i-1} + 360^\circ & \text{if } \Omega_i - \Omega_{i-1} < -180^\circ \\ \Omega_i - \Omega_{i-1} & \text{otherwise} \end{cases} \tag{3.10}$$

All features are subsequently standardized via Z-score scaling to obtain zero-mean, unit-variance representations:

$$z = \frac{x - \mu}{\sigma} \tag{3.11}$$

where μ and σ are the sample mean and standard deviation of each feature, respectively. This normalization mitigates disparities in physical magnitudes—for instance, semi-major axis values on the order of 6.7×10^3 km versus eccentricities near 10^{-3}—and ensures that distance- or density-based detectors operate on comparably scaled inputs.

3.1.2 Orbital Deception Defense Service (ODDS)

ODDS operates as an integrated three-layer pipeline designed to dissect orbital deception through a progressive filtration strategy. Rather than functioning as isolated modules, the layers form a coarse-to-fine categorization chain: the first layer establishes a global statistical baseline to flag deviations; the second layer isolates rare singularities within those deviations; and the third layer resolves structural clusters to assign semantic meaning. This cascading design ensures that high-dimensional orbital features are systematically distilled into coherent, physically interpretable decisions. The design and implementation of these layers are detailed below.

- Normality Modeling Engine: This module constructs the statistical baseline of nominal orbital behavior and derives reference thresholds that guide downstream deviation assessments. It identifies samples that deviate from the learned multimodal distribution and passes these flagged candidates—along with their calibrated decision cutoffs—to the Rarity Event Detector and Structural Pattern Analyzer for subsequent, finer-grained analysis.

  – Baseline Modeling: To capture the multi-shell structure of Starlink, we fit a GMM to the feature vectors $\mathrm{x} \in \mathbb{R}^d$. Each component carries its own weight, mean, and covariance, and the parameters are estimated with Expectation-Maximization (EM). The model size is automatically selected—$K = 7$ minimizes the Bayesian Information Criterion (BIC) among the candidate set $\{2, \ldots, 11\}$, with the Akaike Information Criterion (AIC) used as a sanity check. Full covariance matrices are retained to preserve inter-feature correlations, and the EM procedure is capped at 200 iterations with a 10^{-3} convergence tolerance to ensure stable yet efficient training. Table 3.1 summarizes these configuration choices.

Table 3.1: GMM Configuration Summary

Parameter	Value	Description
Number of components (K)	7	Selected via BIC over $\{2, \ldots, 11\}$ to capture multi-shell structure without overfitting.
Model selection criteria	BIC (primary), AIC (secondary)	Complexity-penalized metrics to balance fit quality and parsimony.
EM iterations	200 (max)	Upper bound ensuring convergence while limiting runtime.
EM tolerance	10^{-3}	Stop when log-likelihood improvement falls below this threshold.
Covariance type	Full matrices	Preserve inter-feature correlations critical to orbital dynamics.

  – Statistical Deviation Detector: Each sample receives a forgery score equal to the negative log-likelihood assigned by the learned GMM. Observations embedded in dense, high-probability regions therefore attain low scores, whereas samples located in sparse regions inherit large scores. Table 3.2 summarizes this interpretation. The decision threshold is set to the 10th percentile of the score distribution within the current batch. This percentile aligns with operational telemetry, which shows that approximately 10% of daily observations exhibit measurable deviations, and yielded

the most favourable F1-score/recall trade-off during validation. Samples falling below this boundary are treated as statistically significant departures from the nominal orbital manifold and are passed downstream as high-priority deception candidates.

Table 3.2: Interpretation of Negative Log-Likelihood Scores

NLL magnitude	Interpretation
Small NLL	Sample is embedded in a high-probability region of the GMM; behaviour is consistent with nominal traffic.
Large NLL	Sample occupies a low-probability region; behaviour is atypical and warrants escalation.

- Rarity Event Detector: Building on the deception candidates surfaced by the GMM baseline, this stage performs localized isolation analyses to recover boundary cases and infrequent high-risk events that global statistics tend to overlook.
  - Data Isolation: The stage employs the IF algorithm, which leverages ensembles of randomly generated decision trees to identify locally isolated points in the high-dimensional feature space. The IF constructs $t = 100$ isolation trees, each grown from a randomly sampled subset of the training data and recursively partitioned until a sample is isolated or the maximum depth is reached. Because forgeries or manipulated samples are both "few and different," they tend to be isolated earlier than nominal samples, yielding shorter average path lengths across the trees. The final forgery score normalizes these path lengths by the expected depth of a random sample and maps the result to the $[0, 1]$ interval—values nearer to one indicate samples that are quickly isolated and therefore more likely to be forged or manipulated. The expected forgery proportion is set to 10%, and the decision threshold is calibrated accordingly to reflect operational requirements.
  - Singularity Capture: The IF is particularly adept at detecting samples whose local density sharply contrasts with their neighbours, without imposing parametric distributional assumptions. Its computational complexity of $O(\log n)$ makes it well suited for large-scale datasets. By operating exclusively on the candidates flagged by the preceding layer, this stage reduces computational overhead, sharpens detec-

tion precision, and compensates for the boundary forgeries, orbital maneuvers, or measurement artefacts that the GMM baseline might miss.

- Structural Pattern Analyzer: This stage performs structured analysis on the deception candidates passed forward by the first two layers, separating systematic deception patterns from incidental noise to enhance interpretability and operational relevance.

  – Core Cluster Identifier: The stage applies HDBSCAN to perform density-based clustering. The configuration sets `min_cluster_size` to 10 so that any extracted dense cluster must contain at least ten samples, and `min_samples` to 5 so that local density estimates rely on the five nearest neighbors. Prediction data is enabled to support the assignment of subsequently arriving samples. Table 3.3 summarizes the role of these hyperparameters.

Table 3.3: Key HDBSCAN Hyperparameters

Parameter	Value	Purpose
`min_cluster_size`	10	A cluster must contain at least 10 points to be considered stable, which filters out spurious groups caused by noise.
`min_samples`	5	Density estimates use the five nearest neighbours, striking a balance between sensitivity to small structures and robustness to noise.

The algorithm constructs a hierarchical cluster structure by computing the mutual-reachability distance $d_{\text{mreach}}(p, q)$ between pairs of points, defined as:

$$d_{\text{mreach}}(p, q) = \max\{\text{core}_k(p), \text{core}_k(q), d_{\text{Euclidean}}(p, q)\} \tag{3.12}$$

where $\text{core}_k(p)$ denotes the core distance of point p (the distance to its k-th nearest neighbour, with $k =$ `min_samples`), and $d_{\text{Euclidean}}(p, q)$ is the standard Euclidean distance. This metric ensures that points in sparse regions have larger mutual-reachability distances, effectively emphasizing dense regions in the clustering process. A minimum-spanning tree (MST) on the mutual-reachability graph yields the density hierarchy, where edge weights represent the minimum density threshold required to connect

two points. The algorithm then extracts the most stable branches—those that persist across multiple density thresholds—as final clusters. Samples assigned to a cluster receive positive integer labels that represent the associated deception archetype.

– Noise Detection Module: Samples that cannot be assigned to any stable cluster are labelled as noise (label: -1) by HDBSCAN, indicating isolated forgeries rather than systematic patterns. The system further infers deviation types based on physical features (inclination, period, perigee altitude, eccentricity, etc.) through a rule-based inference engine. This engine classifies manipulated samples into semantic categories such as deorbiting/disposal (characterized by decreasing perigee altitude and increasing eccentricity), failure or loss of control (indicated by rapid inclination changes or orbit degradation), inclination deviation (marked by significant inclination shifts relative to nominal values), or orbital maneuver (identified by controlled semi-major axis adjustments). This semantic classification enhances the interpretability and practical value of the results, providing structured guidance for subsequent threat assessment and response.

3.1.3 Event Management

The Event Management layer functions as the operational bridge of the detection pipeline, converting analytical inference into actionable system intelligence. It ensures that detected forgeries or manipulated samples are not only flagged but also contextualized, forecasted, and rigorously recorded for future auditing.

- Alarm Module: This component serves as the immediate guardian of data integrity, triggering alerts when the *current* TLE data is identified as compromised. It encapsulates the diagnostic results—such as high forgery scores or recognized error patterns—into standardized payloads. By disseminating these warnings via unified interfaces (e.g., RESTful APIs), it ensures that downstream systems are instantly notified to suspend operations or reject the invalid orbital data.

- Prediction Module: Focusing on temporal degradation, this module forecasts *when* the or-

bital data is likely to diverge from nominal behavior. By utilizing Exponentially Weighted Moving Average (EWMA) to extrapolate deviation trends, it estimates the remaining validity window of the TLEs. This allows operators to anticipate the onset of deception or data degradation and schedule TLE updates proactively, rather than reacting only after data quality has already deteriorated.

- Log System: Ensuring full system observability, this module maintains a granular, structured audit trail (JSON) of the entire data lifecycle. It captures specific inputs, model states, execution latencies, and operator decisions for every processing step. This comprehensive logging guarantees that all automated actions and human interventions are reproducible, facilitating precise root cause analysis and forensic reconstruction.

3.1.4 Simulation

A controlled simulation environment is essential for quantitatively evaluating deception defense performance, particularly given the scarcity of labeled forgeries in real-world satellite telemetry. By establishing a framework with explicit ground truth, the system facilitates the rigorous calculation of precision, recall, and F1-scores. The simulation workflow proceeds through a structured sequence comprising data synthesis, threshold optimization, output validation, and robustness stress testing.

The process commences with Forgery Data Generation, which establishes the foundation for controlled experimentation. This phase integrates statistical modeling with orbital mechanics through two primary mechanisms. First, a GMM fits the nominal orbital parameters, where the BIC determines the optimal component count—validated by the AIC—while full covariance matrices preserve inter-feature dependencies. Second, the system introduces Controlled Perturbation by injecting specific deviations, such as semi-major axis shifts or inflated Keplerian residuals. Crucially, a strict filtering layer ensures physical plausibility by rejecting any samples violating orbital constraints (e.g., inclination limits), thereby maintaining high fidelity in the synthetic dataset.

Following data generation, the workflow transitions to Threshold Optimization. Since un-

supervised models output continuous forgery scores rather than binary labels, establishing effective decision boundaries is critical. This phase employs Proportion Control to regulate the ratio between nominal and perturbed data, simulating diverse contamination scenarios ranging from rare-event detection to high-noise environments. Simultaneously, Threshold Scanning systematically evaluates potential cut-offs across the score distribution, identifying the operating point that maximizes detection metrics relative to the known ground truth.

With decision boundaries established, the framework proceeds to Classification Calibration. This phase bridges the gap between unsupervised model outputs and semantic forgery definitions. Through Label Mapping, the system aligns generated clusters or forgery scores with the binary ground truth labels created during the synthesis phase. A subsequent Matching Test verifies this alignment, ensuring that detected forgeries correspond to actual injected perturbations rather than statistical noise within the nominal distribution.

Finally, the detection stack undergoes a Robustness Stress Test to characterize its operational limits. This phase evaluates algorithm resilience under adverse conditions through Signal Attenuation, where the magnitude of injected perturbations is progressively reduced to identify the sensitivity floor (minimum detectable signal). Furthermore, testing against extreme edge cases defines the boundaries of algorithmic performance, providing a comprehensive profile of system reliability against subtle adversarial spoofing or environmental noise.

3.1.5 Infrastructure Layer

The infrastructure layer furnishes the foundational services required for system operation, ensuring that non-functional requirements—stability, scalability, maintainability, and security—meet production-grade standards.

- Database (PostgreSQL): Serving as the unified persistence layer, the relational database stores both structured and semi-structured artefacts, including raw TLE records, processed feature vectors, model outputs, deception adjudications, and operational logs. Carefully designed schemas and time-series indexing enable efficient temporal queries, historical replay, and long-horizon trend analysis. ACID-compliant transaction management pre-

serves data integrity, while master−replica replication and backup policies deliver high availability in alignment with audit and compliance obligations.

- Operating System (Ubuntu 22.04): The platform offers a stable, container-friendly runtime that supports Docker-based deployment for consistent environments across hosts. Integrated resource monitoring, version control, and security hardening uphold uniform standards for horizontal scalability and fault tolerance. Observability is delivered through coordinated log aggregation, metrics collection, and distributed tracing, enabling rapid diagnostics and performance tuning. Security posture is further reinforced by the principle of least privilege and network isolation policies.

3.2 Experimental Design

This section outlines the experimental protocol, encompassing the construction of the single-day Starlink dataset and the synthetic TLE generation mechanism. It defines the parameterized deception scenarios and the associated semi-supervised labeling workflow, concluding with the quantitative metrics used to evaluate detection efficacy and operational utility. The methodology prioritizes data transparency, scenario reproducibility, and the alignment of evaluation criteria with practical monitoring requirements.

3.2.1 Dataset

To accurately capture the nominal operational behavior of the Starlink constellation, this study employs a Single-day Constellation Snapshot strategy. This approach is necessitated by the highly dynamic nature of the LEO environment, where satellite ephemerides are strictly time-sensitive. Contrary to the assumption of static Keplerian orbits, TLEs represent a ”best fit” snapshot that degrades over time due to three primary physical factors. First, at an altitude of approximately 550 km, satellites experience atmospheric drag, causing continuous velocity loss and orbital decay. Second, gravitational perturbations—arising from Earth’s oblateness (J_2 term) and luni-solar forces—induce precession of the orbital plane. Third, Starlink satellites

frequently execute active maneuvers using Hall thrusters for station-keeping and autonomous collision avoidance, rendering previous orbital predictions immediately obsolete.

These factors create a continuous ”Orbit Maintenance Loop,” where the validity of a TLE is confined to a short window around its specific Epoch. As the time delta from the epoch increases, propagation residuals—representing the divergence between the SGP4-predicted state and the actual satellite position—accumulate exponentially. Consequently, relying on multi-day temporal data without correction introduces significant noise. By isolating all satellites within the same daily snapshot, this study minimizes these cumulative residuals, ensuring that the dataset reflects the current physical state of the constellation—preserving high temporal fidelity—rather than artifacts of environmental drift or stale data.

The dataset is constructed from Starlink TLEs published by CelesTrak [59]. It is pertinent to note that CelesTrak archives mirror the official High-Accuracy Catalog maintained by the 18th Space Defense Squadron (18 SDS). While independent physical verification (e.g., via ground-based radar or optical tracking) is outside the scope of this work, these records constitute the global *de facto* standard for unclassified orbital data. Thus, they are accepted as the *ground truth* for nominal operations, serving as the trusted baseline against which forgeries and adversarial spoofing attacks are evaluated.

The final preprocessed dataset comprises 8,451 validated records. Key statistical characteristics are summarized in Table 3.4. The constellation operates within the LEO regime, with perigee and apogee altitudes oscillating between 500 and 650 km. Orbital periods are tightly clustered around a mean of approximately 90 minutes (95.6 $\pm$ 2.5 min). The data exhibits a distinct multi-modal inclination distribution (spanning 43°–98°), reflecting the constellation's multi-shell architecture, while eccentricity values remain near-zero ($e \approx 0$), indicating predominantly circular orbits. In terms of data integrity, the dataset maintains a missing-value ratio of $< 0.1\%$ post-processing.

Preliminary analysis via a GMM suggests a baseline forgery rate of approximately 10% (determined via a 10th-percentile threshold). This proportion aligns with expected operational deviations in large-scale constellations, confirming the dataset's suitability for benchmarking unsupervised deception defense algorithms reliant on residual analysis.

Table 3.4: Dataset Statistics (Starlink Single-day Snapshot)

Dimension	Statistic	Value	Description
Sample Size	Count	8,451	Validated TLE entries
Inclination	Range (Mode)	43°–98° (∼53°)	Multi-shell configuration
Period	Mean ± SD	95.6 ± 2.5 min	LEO operational regime
Eccentricity	Median (IQR)	2×10^{-4} ($1–4 \times 10^{-4}$)	Near-circular orbits
Data Quality	Validity Rate	$> 99.9\%$	High integrity
Baseline Noise	Est. Forgery Rate	$\sim 10\%$	Based on GMM (10th pct)

3.2.2 Deception Scenario Design

Robust validation of an autonomous deception detector requires performance verification across both nominal and off-nominal orbit regimes. To stress-test the detector's sensitivity to dynamic deviations and its specificity against distributional forgeries or manipulated samples, this study designs two primary physical deviation scenarios as shown in Fig. 3.5: (A) Abnormal Maneuver and (B) Inclination Deviation. Crucially, all off-nominal tracks are processed through the same pipeline as nominal data, ensuring that any detected deviation is attributed solely to the injected perturbation.

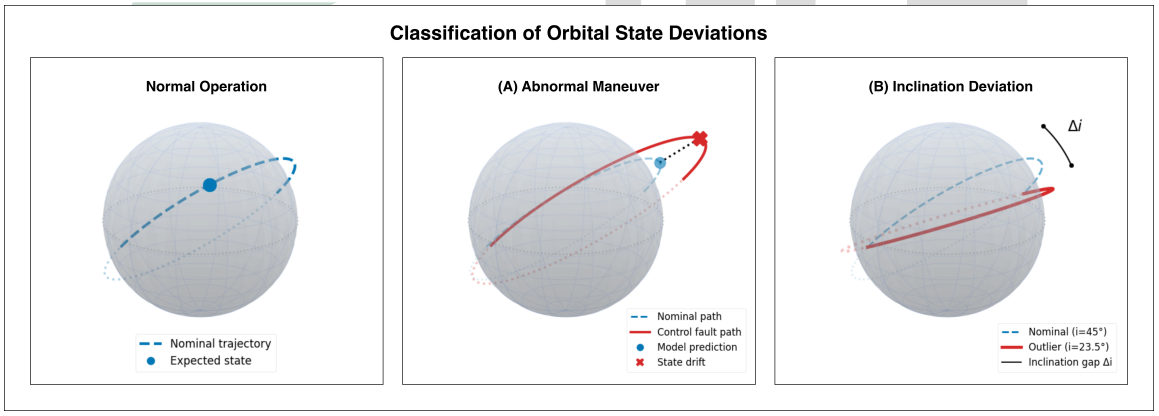

Figure 3.5: Classification of Orbital State Deviations

- Normal Operation:

  The baseline trajectory (dashed line) represents the ground-truth behavior of a satellite maintaining its station-keeping window. The Expected state (blue dot) serves as the reference point for the nominal residual profile.

- (A) Abnormal Maneuver:

This scenario simulates an unmodeled thrust event, producing a Control fault path (solid red line). Mathematically, the semi-major axis a is perturbed and the mean motion n is recomputed via Kepler's third law ($n^2a^3 = \mu$). The deviation is captured as a State drift (red "X"), representing the Euclidean distance between the Model prediction (blue dot) and the actual TLE observation:

$$\boldsymbol{\delta} = \mathrm{r}_{obs}(t_1) - \mathrm{r}_{pred}(t_1)$$

This tests the detector's sensitivity to temporal inconsistencies in the dynamic feature subspace.

- (B) Inclination Deviation:

  Distributional deviations occur when orbits are dynamically valid but operationally non-compliant. In this case, the inclination i is shifted from the Nominal ($i = 45°$) to an Outlier ($i = 23.5°$), creating a significant Inclination gap Δi. This scenario challenges the classifier's decision boundary in the geometric subspace, testing robustness against mission-noncompliant out-of-distribution samples.

3.2.3 Deception Attack

While the previous sections define deviations at the physical level, this section introduces the *data-layer* adversarial framework. We transition from a purely observational perspective to an active-threat model, assuming an adversary with Man-in-the-Middle (MitM) capabilities or unauthorized write access to the TLE distribution channel. In this context, the adversary does not merely corrupt data; they synthesize forged TLE records that are semantically consistent and structurally valid. By carefully modulating specific orbital elements, the attacker aims to manipulate the detector's residual vector $\boldsymbol{\delta}$ to induce either operational blindness or systemic panic.

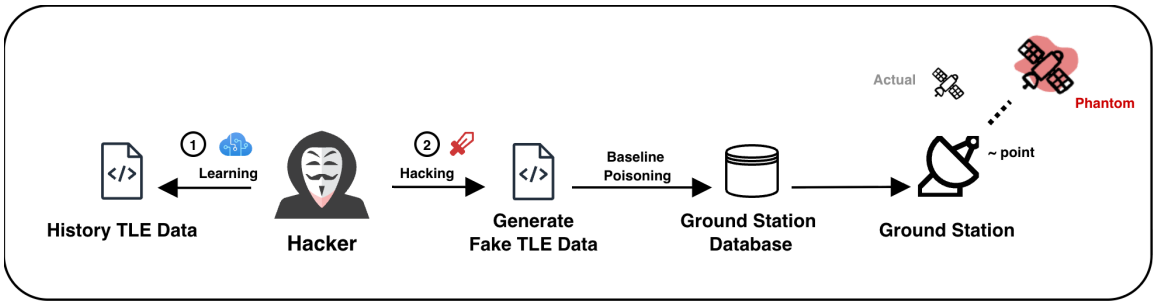

Figure 3.6: Adversarial TLE Deception Attack Workflow

Fig. 3.6 illustrates the dual-strategy workflow designed to exploit the detector's logic through the following two deceptive modes:

Scenario A In this mode, the adversary seeks to "cloak" a genuine Abnormal Maneuver. Although the physical satellite has undergone a significant drift (e.g., $|\delta_a| > 100$ km), the attacker forges a TLE by reverting the dynamic elements (a, n) back to their nominal constellation centroids. To bypass temporal consistency checks, the adversary re-calibrates the mean anomaly M and RAAN Ω, thereby enforcing temporal continuity between the pre-maneuver state and the forged observation. By artificially nullifying the state drift $\boldsymbol{\delta}$, the attacker induces a false negative, ensuring the critical propulsion event remains undetected by the autonomous monitor.

Scenario B In contrast, this mode cloaks a genuine Inclination Deviation by re-mapping the anomalous geometric state back into the nominal shell distribution. As illustrated in Fig. 3.7 (right), when the actual state exhibits an operationally noncompliant shift—characterized by a drift from the nominal inclination $i \approx 53°$ and a concurrent rise in eccentricity e—the adversary perturbs these geometric elements to synthesize a forged TLE that resides within the nominal statistical cluster. To evade energy-consistency filters that monitor orbital period stability, the attacker preserves the original (a, n) pair while re-calculating the argument of perigee ω (and associated phase terms) to maintain the spatial continuity of the ground track. By artificially compressing the geometric residual, the detector is driven into a false negative, effectively masking a high-priority orbital deviation as benign station-keeping behavior.

Fig. 3.7 summarizes their impact on the classification logic and highlights which feature subspaces (dynamic vs. geometric) are preferentially targeted by the adversary.

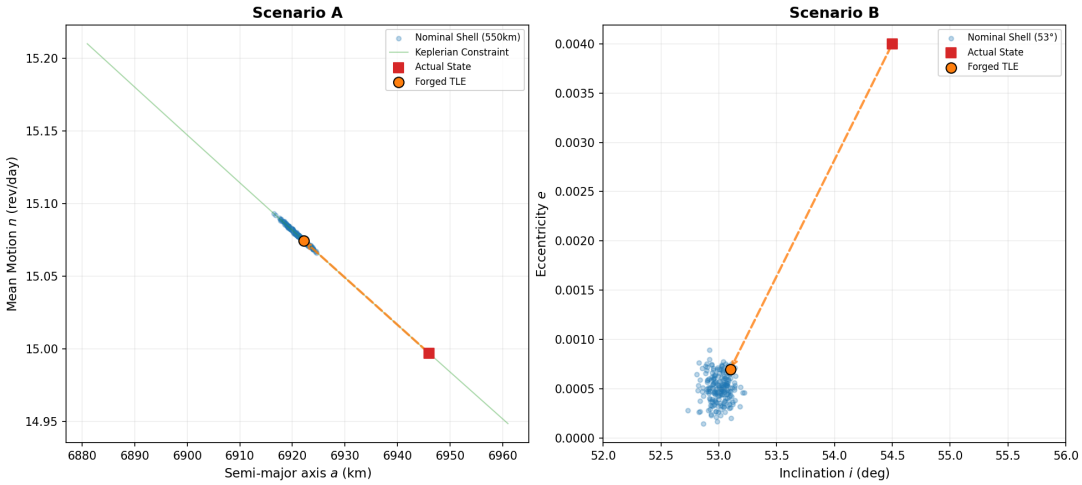

Figure 3.7: Deception Attack Strategies Overview

Chapter 4 Implementation

This chapter details the implementation of the proposed deception defense framework. The implementation covers the experimental environment setup, data processing pipeline, multi-stage deception defense system, event management, and simulation capabilities.

4.1 Environment Setup

The experimental setup focuses on the computational infrastructure and software environment established for evaluating the proposed deception defense system. This foundation ensures that all experiments are conducted in a reproducible and controlled environment, enabling fair comparisons and reliable interpretation of results. The following subsections detail the hardware specifications and software dependencies required for system execution.

4.1.1 Hardware Specification

All experiments in this study were conducted on a standard workstation; the detailed hardware specifications are listed in Table 4.1. The experimental platform utilizes the Apple Silicon architecture (M1 Pro chip), featuring a Unified Memory Architecture that efficiently facilitates machine learning model training and inference. The 8-core CPU and 14-core GPU configuration provides substantial parallel processing capabilities, which are particularly beneficial for training complex models such as GMM on large-scale TLE datasets. The operating system employed is macOS Sonoma 14.5.0, which ensures a stable execution environment and provides comprehensive support for the Python scientific computing ecosystem.

Table 4.1: Hardware Specifications

Component	Specification
Machine	MacBook Pro 14” (M1 Pro Chip)
Processor	Apple M1 Pro (8-core CPU, 14-core GPU)
Operating System	macOS Sonoma 14.5.0
Memory (RAM)	16 GB Unified Memory
Storage	Solid State Drive (SSD)

4.1.2 Software Specification

The experimental workflow is implemented in Python 3.8+, integrating a comprehensive suite of open-source libraries. `scikit-learn` (v1.0+) serves as the core engine for unsupervised learning algorithms, including GMM, IF, and HDBSCAN. Data manipulation and scientific computing tasks are handled by `pandas`, `numpy`, and `scipy`, while `matplotlib` and `seaborn` facilitate the visualization of orbital plots and evaluation metrics. To ensure strict reproducibility of the reported results, a fixed random seed (`random_state=42`) is applied to all stochastic processes, including initialization and data splitting. The complete software configuration is detailed in Table 4.2.

Table 4.2: Third-Party Software List

Component	Version	Purpose	License
Python	3.8+	Primary Development Language	PSF License
scikit-learn	1.0+	ML Algorithms (GMM, IF, HDBSCAN)	BSD 3-Clause
pandas	1.3+	Data Manipulation & Structuring	BSD 3-Clause
numpy	1.21+	Numerical Computing & Array Ops.	BSD 3-Clause
matplotlib	3.4+	Basic Visualization	PSF-based
seaborn	0.11+	Statistical Data Visualization	BSD 3-Clause
scipy	1.7+	Scientific Computing & Statistics	BSD 3-Clause

4.2 Data Processing

Algorithm 2 details the implementation workflow for the data processing pipeline, transforming raw TLE files into a standardized feature matrix suitable for deception defense.

The outer loop in Lines 1–6 iterates through each TLE file in the data directory. For each file, raw lines are read in Line 2 and grouped into triplets corresponding to individual satellite records in Line 3. The inner loop in Lines 4–5 processes each triplet: physical parameters are extracted using the fixed-field parsing strategy detailed in Chapter 3 in Line 5, ensuring column-level accuracy, and appended to the orbital data collection.

After all files are processed, classical orbital elements are extracted to form the raw feature

matrix X_{raw} in Line 7. Temporal-derivative features are computed in Line 8 using backward-difference operators to capture rate-of-change indicators. The combined feature set undergoes Z-score standardization in Line 9, producing the standardized matrix X that serves as input to downstream deception defense modules. The algorithm returns the standardized feature matrix in Line 10.

Algorithm 2: Data Processing

Input : Raw TLE Data Files

Output: Processed Orbital Feature Matrix X

```
1  foreach TLE_File in Data_Directory do
2      Lines ← Read(TLE_File);
3      Triplets ← Group Lines Into Triplets(Lines);
4      foreach Triplet in Triplets do
5          PhysParams ← Parse TLE Physics(Triplet);
6          OrbitalData.Append(PhysParams);
7  X_raw ← Extract Orbital Elements(OrbitalData);
8  X_derived ← Compute Rate Features(X_raw);
9  X ← Standardize(X_raw, X_derived);
10 return X;
```

4.3 Multi-Stage Deception Defense

This section presents the implementation details of the multi-stage deception defense pipeline, which operates as a cascading filtration system that progressively refines deception candidates through three complementary stages. Each stage addresses different aspects of forgery identification, from global statistical deviations to local rarity events and structural pattern recognition.

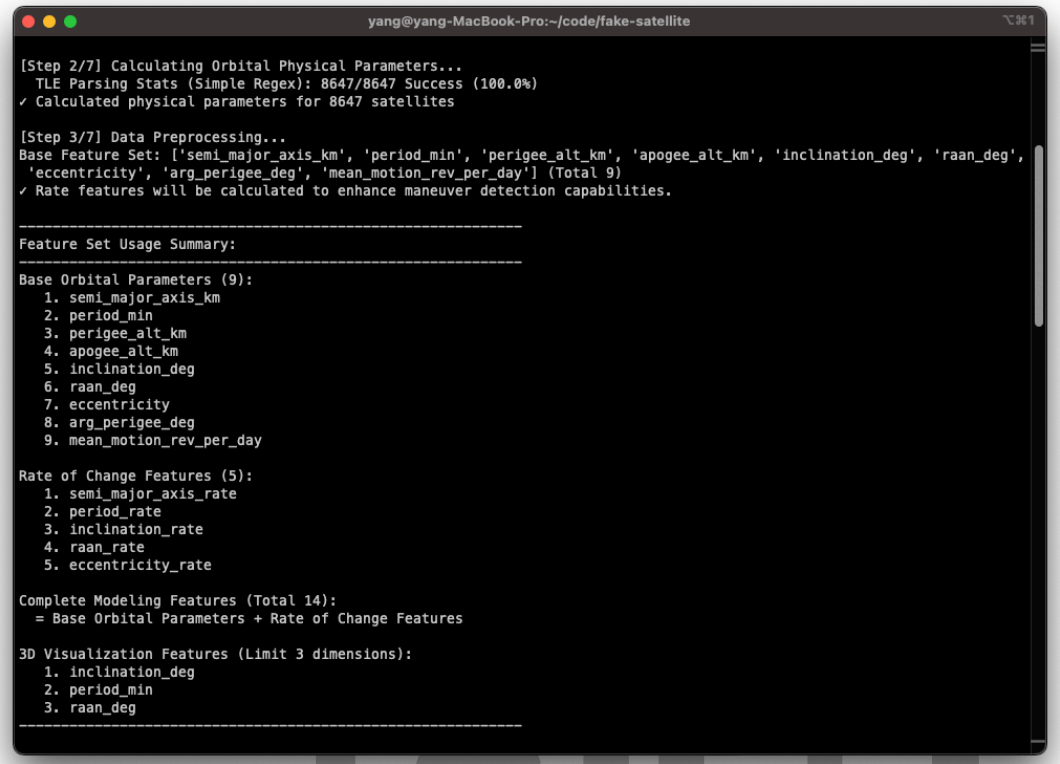

Figure 4.1: Data Processing Results

4.3.1 Normality Modeling Engine

Algorithm 3 delineates the execution workflow of the Normality Modeling Engine. Accepting the standardized feature matrix X and component count $K = 7$ as inputs, the procedure initiates with Gaussian Mixture Model (GMM) training in Line 1. The model is instantiated using the specific hyperparameter configuration—including convergence tolerance and iteration limits—summarized in Table 3.1.

In Line 2, forgery quantification is performed by computing the negative log-likelihood for each sample based on the trained GMM. The decision threshold, τ, is subsequently established in Line 3 as the 10th percentile of the score distribution. Following the initialization of the candidate subset $\mathcal{A}$ (Line 4), the algorithm iterates through the dataset in Lines 5–8. Samples yielding scores below τ are flagged as forgeries or manipulated samples and aggregated into $\mathcal{A}$.

Lines 9–10 enforce physical constraint validation, where flagged satellites are cross-referenced against constellation-specific operational bounds. Finally, the results are exported to a structured

format in Line 11, and the algorithm returns the forgery scores, threshold, and the identified deception candidate subset.

Algorithm 3: Normality Modeling Engine

Input : Feature Matrix X, Number of Components $K = 7$

Output: Anomaly Scores s, Threshold τ, Anomaly Subset $\mathcal{A}$

1 $GMM \leftarrow$ TrainGaussianMixture(X, K);
2 s $\leftarrow$ GMM.ScoreSamples(X);
3 $\tau \leftarrow$ Percentile(s, 10%);
4 $\mathcal{A} \leftarrow \emptyset$;
5 **foreach** *Sample* i ***in*** X **do**
6 **if** $s_i < \tau$ **then**
7 $\mathcal{A}$.Append(i);
8 MarkAsAnomaly(i);
9 **foreach** *Satellite* ***in*** $\mathcal{A}$ **do**
10 EvaluatePhysicalRules(Satellite);
11 SaveResults($\mathcal{A}$);
12 **return** s, τ, $\mathcal{A}$;

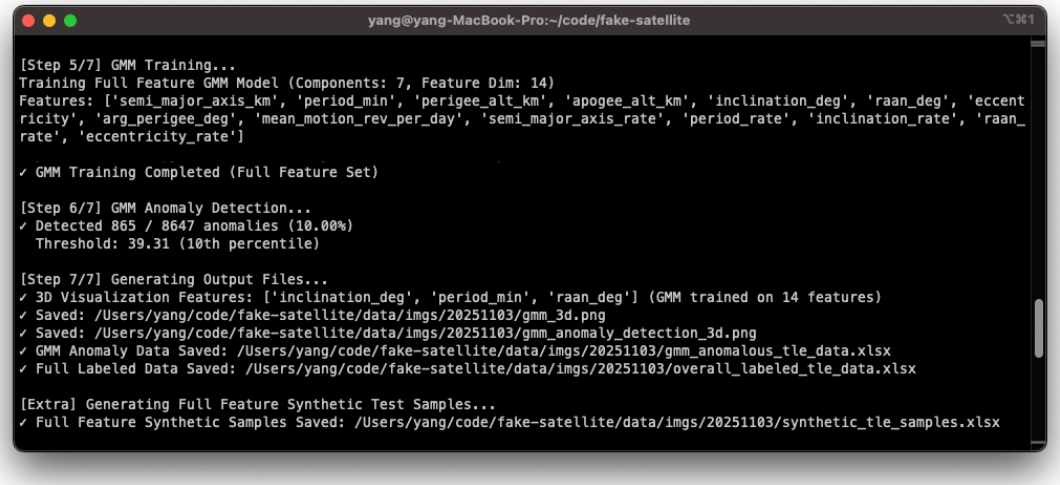

Figure 4.2: GMM Model Training Process

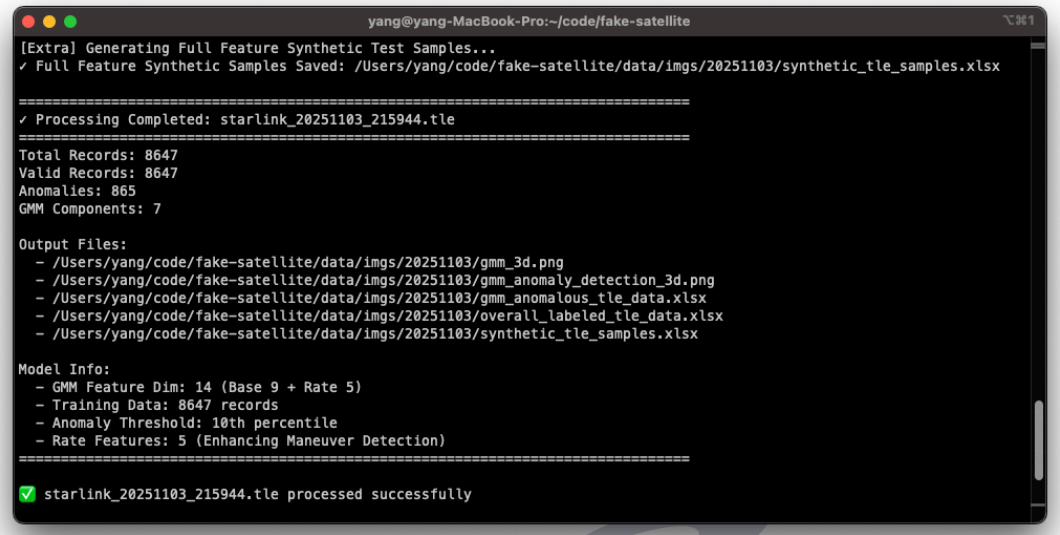

Figure 4.3: GMM Model Training Results

4.3.2 Orbital-Constraint Validation

Algorithm 4 delineates the physics-based deterministic validation mechanism, which serves as a complement to the statistical GMM by enforcing rigid astrodynamic constraints.

Lines 1–5 focus on the theoretical reconstruction of orbital dynamics via Kepler's Third Law. The Earth's standard gravitational parameter, μ, is initialized in Line 1. To ensure system robustness against corrupted telemetry, the algorithm performs input validation on the semi-major axis in Lines 2–3, returning a null indicator for invalid data. The theoretical period is subsequently derived in Lines 4–5, converting the result into minutes for consistency with the feature space.

Lines 7–14 handle the retrieval and evaluation of operational boundaries. Constellation-specific thresholds for the semi-major axis and orbital period—as defined in the system configuration—are loaded in Lines 7–10. The algorithm then validates the observed telemetry against these bounds in Lines 11–14, generating boolean flags that indicate any excursions from the nominal Starlink Gen-1 operational envelope.

The final validation metrics are synthesized in Lines 16–18. The Keplerian residual, r_{min}, is computed as the absolute deviation between the observed and theoretical periods, quantifying the

adherence to ideal orbital mechanics. Simultaneously, the range violation flags are consolidated, ensuring all parameters remain within the expected physical constraints.

Algorithm 4: Physics-Based Deterministic Validation

Input : Semi-major axis a_{km}, Observed period T_{obs}

Output: Theoretical period T_{theory}, Keplerian residual r_{min}, Range violation flags (out_a, out_t)

$\mu \leftarrow 398600.4418$;

if a_{km} *is Invalid* ***or*** $a_{km} \leq 0$ then

  return *Null*;

$T_{seconds} \leftarrow 2\pi\sqrt{a_{km}^3/\mu}$;

$T_{theory} \leftarrow T_{seconds}/60$;

$a_{bounds} \leftarrow$ RetrieveConfigBounds("semi_major_axis_km");

$t_{bounds} \leftarrow$ RetrieveConfigBounds("period_min");

$out_a \leftarrow (a_{km} < a_{bounds}.\text{min}) \vee (a_{km} > a_{bounds}.\text{max})$;

$out_t \leftarrow (T_{obs} < t_{bounds}.\text{min}) \vee (T_{obs} > t_{bounds}.\text{max})$;

$T_{theory} \leftarrow$ CalculateKeplerianPeriod(a_{km});

$r_{min} \leftarrow |T_{obs} - T_{theory}|$;

$(out_a, out_t) \leftarrow$ AssessConstraints(a_{km}, T_{obs});

return $T_{theory}, r_{min}, (out_a, out_t)$;

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	name	epoch_timestam	major_axis	period_mi	rigee_alt	logee_alt	clination_d	raan_deg_t	perigee_	eccentricit	motion_rev	eoretical_t	kepler_residual_min	is_out_of_range_a	is_out_of_range_T	is_anomaly	anomaly_score
2	STARLINK-1008	3.90277E+12	6925.376	95.5927	546.2932	548.1839	53.0544	37.8105	79.2826	0.000137	15.06391	95.5927	7.10543E-14	FALSE	FALSE	TRUE	-39.05381778
3	STARLINK-1010	2.64233E+12	6648.957	89.92699	267.2307	274.4103	53.0413	322.5987	65.4426	0.00054	16.01299	89.92699	5.68434E-14	TRUE	TRUE	TRUE	-7.135565123
4	STARLINK-1011	5.04134E+12	6689.91	90.7591	308.7506	314.7956	53.0483	18.5277	329.7236	0.000452	15.86618	90.7591	5.68434E-14	TRUE	FALSE	TRUE	-9.539570045
5	STARLINK-1012	4.28497E+12	6925.387	95.59293	546.38	548.1196	53.0551	37.6121	91.859	0.000126	15.06388	95.59293	8.52651E-14	FALSE	FALSE	TRUE	-39.14564263
6	STARLINK-1017	3.96513E+12	6898.689	95.0407	519.5411	521.5638	53.0522	34.2531	272.7231	0.000147	15.1514	95.0407	7.10543E-14	FALSE	FALSE	TRUE	-26.24150341

Figure 4.4: Orbital Constraint Validation Results

4.3.3 Rarity Event Detector

Algorithm 5 outlines the architectural logic of the Rarity Event Detector, which executes a localized isolation analysis on the deception candidates previously identified by the Normality Modeling Engine.

The procedure commences in Line 1 by projecting the GMM-identified subset $\mathcal{A}$ onto a consistent feature space using the pre-fitted scaler $\mathcal{S}$. This ensures that the subsequent isolation mechanism operates on the same distributional scale as the global model. In Line 2, an IF ensemble—configured with 100 estimators and a contamination factor $\eta = 0.1$—is deployed to partition the feature space. This step calculates forgery scores based on normalized path lengths, effectively filtering out noise and retaining only those samples that exhibit distinct isolation characteristics ('refined deception candidate subset' $\mathcal{A}_{iso}$).

To transition from individual point detection to pattern recognition, density-based clustering is applied in Line 3. Using a minimum cluster size of 10, the algorithm aggregates spatially proximate anomalies into semantic clusters. Finally, Line 4 yields both the refined subset and the resulting cluster set $\mathcal{C}$, establishing the structural foundation for the subsequent pattern analysis stage.

Algorithm 5: Rarity Event Detector

Input : GMM Anomaly Subset $\mathcal{A}$, Scaler $\mathcal{S}$

Output: Refined Anomaly Subset $\mathcal{A}_{iso}$, Cluster Set $\mathcal{C}$

1 $\mathrm{X}_{scaled} \leftarrow$ StandardizeFeatures$(\mathcal{A}, \mathcal{S})$;

2 $\mathcal{A}_{iso} \leftarrow$ ExecuteIsolationForest$(\mathrm{X}_{scaled}, \eta = 0.1)$;

3 $\mathcal{C} \leftarrow$ PerformDensityClustering$(\mathcal{A}_{iso})$;

4 return $\mathcal{A}_{iso}, \mathcal{C}$;

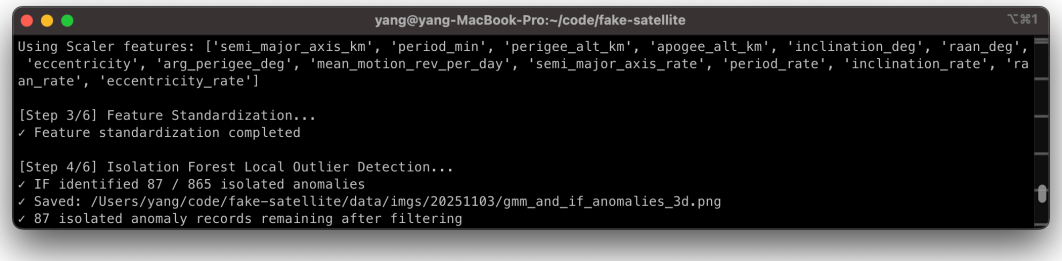

Figure 4.5: IF Training Process

4.3.4 Structural Pattern Analyzer

The Structural Pattern Analyzer transforms raw cluster assignments into semantically interpretable categories based on physical orbital characteristics. Algorithm 6 delineates this classification workflow, which bridges the gap between unsupervised clustering and operational threat assessment by applying rule-based inference to cluster-level statistics.

The iterative process commences in Line 1, looping through each cluster c within the set $\mathcal{C}$. Note that this input set is derived from the preceding density-based detection stage using the hyperparameter configurations summarized in Table 3.3. For every iteration, Line 2 computes a vector of centroidal statistics, μ_c, aggregating key orbital parameters—including inclination, period, perigee/apogee altitudes, and eccentricity—to represent the collective physical behavior of the cluster.

The hierarchical decision logic is executed in Lines 3–16. First, Line 3 evaluates inclination anomalies, prioritizing clusters where the mean inclination deviates from the constellation's nominal band $[Inc_{\text{min}}, Inc_{\text{max}}]$. Subsequently, Lines 6–7 identify deorbit or retirement events by detecting simultaneous decay in period and perigee altitude (below 96% of nominal minimums). High eccentricity values ($> 130\%$ of the limit) are checked in Lines 8–9, flagging potential loss of station-keeping. Lines 10–11 distinguish orbit maneuvers via altitude deviations within a normal period range, while severe geometric distortions are identified in Lines 12–13. Clusters failing to match these criteria are defaulted to an "Other Anomaly" category in Lines 14–15, and the determined label is formally assigned in Line 16.

Upon completion of the classification loop, the system synthesizes the refined deception candidate subset, cluster associations, and semantic labels into a unified result set $\mathcal{R}$ in Line 17. The final classification result is returned in Line 18 to facilitate downstream analysis.

Algorithm 6: Structural Pattern Analyzer

Input : Refined Anomaly Subset $\mathcal{A}_{iso}$, Cluster Set $\mathcal{C}$, Normal Baselines $\mathcal{B}$

Output: Anomaly Classification Result $\mathcal{R}$

foreach *Cluster* $c \in \mathcal{C}$ do
    $\mu_c \leftarrow$ ComputeClusterCentroids(c);
    if $\mu_{inc} \notin [Inc_{min}, Inc_{max}]$ then
        $L_c \leftarrow$ "Inclination Anomaly";
    else if $(\mu_T < 0.96T_{min}) \wedge (\mu_{h_p} < 0.96h_{p,min})$ then
        $L_c \leftarrow$ "Deorbit / Retired";
    else if $\mu_e > 1.3e_{max}$ then
        $L_c \leftarrow$ "Malfunction / Uncontrolled";
    else if $\mu_T \in \mathcal{B}_T \wedge (\mu_{h_p} < 0.98h_{p,min} \vee \mu_{h_a} > 600)$ then
        $L_c \leftarrow$ "Orbit Maneuver";
    else if $\mu_{h_a} > 1.5\mu_{h_p} \vee \mu_{h_p} < 0.92h_{p,min}$ then
        $L_c \leftarrow$ "Orbit Shape Anomaly";
    else
        $L_c \leftarrow$ "Other Anomaly";
    AssignSemanticLabel(c, L_c);
$\mathcal{R} \leftarrow$ CompileAnalysisResults$(\mathcal{A}_{iso}, \mathcal{C}, L)$;
return $\mathcal{R}$;

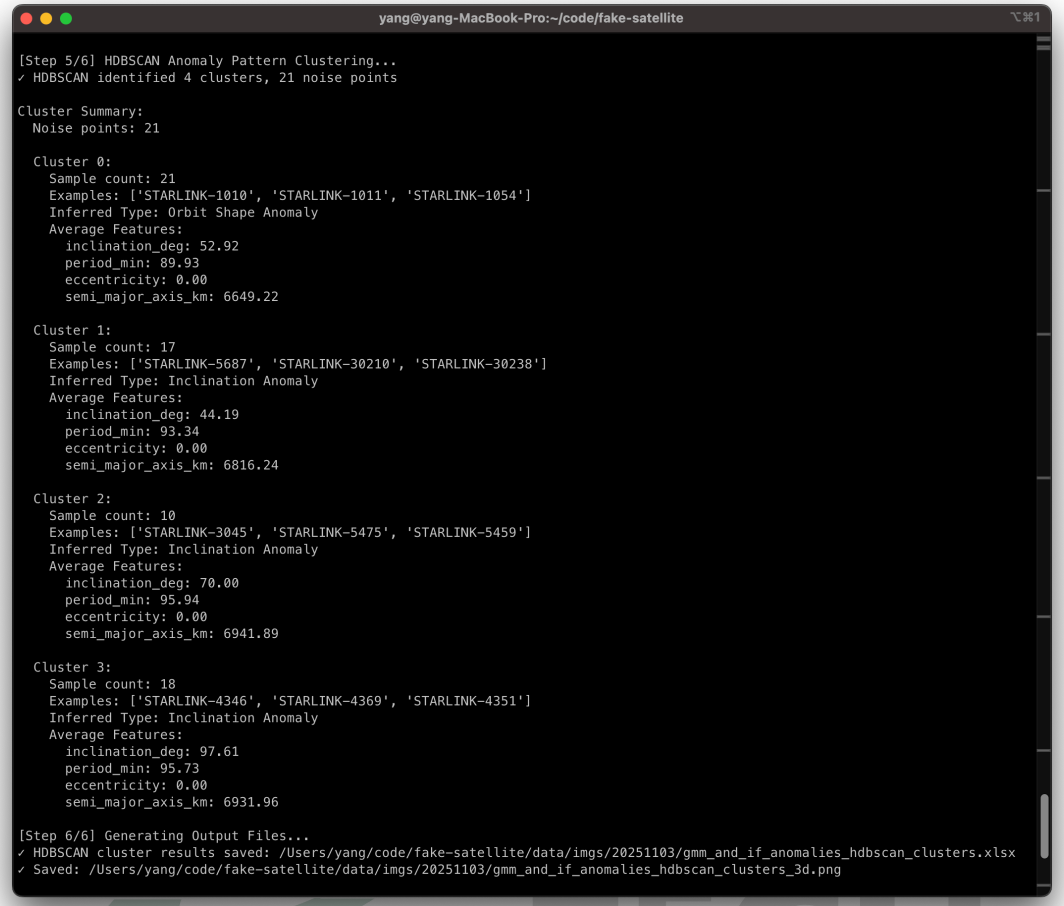

Figure 4.6: HDBSCAN Clustering Results

4.4 Orbital Pattern Learning Module

While the structural classification identifies categories of anomalies, operational requirements often necessitate a mechanism to reconcile these deviations with historical norms. The Baseline-Driven Feature Correction Module achieves this by establishing a moving lookback window—specifically the previous 10 observation folders —as a localized manifold for normality. This module ensures that anomalous samples are not merely discarded but are instead projected back into the historically validated ”nominal” subspace through a deception mechanism that replaces anomalous orbital parameters with values from the baseline cluster.

The deception process is implemented through Algorithm 7, which follows a sequence of data standardization, hybrid probabilistic scoring, and feature substitution.

Algorithm 7: TLE Data Deception via Baseline Feature Substitution

Input : Baseline dataset $\mathcal{D}_{base}$ from prev10 folders, target dataset $\mathcal{X}_{target}$, threshold $\tau = 0.5$

Output: Deceived dataset $\mathcal{X}_{deceived}$ with anomalies replaced by baseline neighbors

1 Extract feature triplet $\mathrm{X}_{base} = [i, T, \Omega]$ from $\mathcal{D}_{base}$ and X_{target} from $\mathcal{X}_{target}$;

2 Standardize: $\mathrm{X}_{base}^{scaled}, \mathrm{X}_{target}^{scaled} \leftarrow \mathrm{StandardScaler}(\mathrm{X}_{base}, \mathrm{X}_{target})$;

3 Compute 1-NN distances: $d_{NN}, \mathrm{indices}_{NN} \leftarrow \mathrm{NearestNeighbors}(\mathrm{X}_{base}^{scaled}, \mathrm{X}_{target}^{scaled})$;

4 Compute GMM scores: $P_{gmm}(\mathrm{x}) \leftarrow \mathrm{GaussianMixture}(\mathrm{X}_{base}^{scaled}, \mathrm{X}_{target}^{scaled})$;

5 Combine: $P_{fake}(\mathrm{x}) \leftarrow 0.6 \cdot P_{gmm}(\mathrm{x}) + 0.4 \cdot (1 - \exp(-d_{NN}))$;

6 Flag anomalies: is_fake $\leftarrow P_{fake}(\mathrm{x}) > \tau$;

7 foreach *sample* x_i *where is_fake*$[i] =$ *True* do

8 $\quad j^* \leftarrow \mathrm{indices}_{NN}[i]$;

9 $\quad$ Replace: $\mathrm{x}_i[i, T, \Omega] \leftarrow \mathcal{D}_{base}[j^*][i, T, \Omega]$;

10 return $\mathcal{X}_{deceived}$;

Lines 1–2 of Algorithm 7 extract the core feature triplet—inclination (i), period (T), and RAAN (Ω)—from both the baseline $\mathcal{D}_{base}$ and the target $\mathcal{X}_{target}$, then apply standardization so that Euclidean distances are computed on a uniform scale.

Lines 3–6 implement the hybrid scoring mechanism. Line 3 computes 1-NN distances and the baseline neighbor indices $\mathrm{indices}_{NN}$; Line 4 obtains the GMM-based forgery score $P_{gmm}(\mathrm{x})$ from the baseline density. Line 5 combines the two scores with weight $w_{gmm} = 0.6$, and Line 6 flags any sample with $P_{fake}(\mathrm{x}) > \tau$ as a forgery or manipulated sample. This balances density-based (GMM) and distance-based (1-NN) sensitivity.

The loop in Lines 7–9 performs feature substitution: for each flagged sample x_i, Line 8 selects the nearest baseline index $j^* = \mathrm{indices}_{NN}[i]$, and Line 9 replaces the orbital parameters (i, T, Ω) with the values from $\mathcal{D}_{base}[j^*]$. The deceived sample thus lies within the historically validated nominal cluster, masking transient or unverified orbital deviations.

The effectiveness of this deception is validated through 3D displacement visualization, as shown in Fig. 4.7. The ”Original Detection” plot highlights anomalous points with red ”X”

markers and visualizes the correction path via dark orange dashed lines connecting original positions to corrected positions, with green arrows (quivers) indicating the displacement direction. The corresponding ”After Displacement” plot confirms that all target points have been successfully moved into the baseline cluster region, with originally normal points shown in blue and deceived points (originally anomalous) shown in teal. This systematic feature substitution facilitates the generation of datasets that are statistically consistent with historical behavior while concealing actual orbital anomalies.

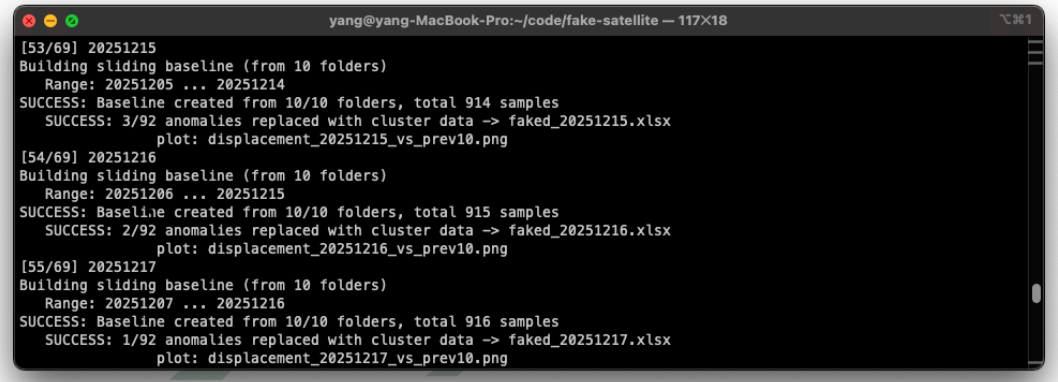

Figure 4.7: Orbital Pattern Learning Process

4.5 Event Management

Bridging analytical inference and operational intelligence, the Event Management layer converts detection results into actionable alerts, forecasts, and audit records.

4.5.1 Alarm Mechanism

Algorithm 8 defines the alarm mechanism that transforms detection results into actionable security responses. The procedure processes the deception classification result $\mathcal{R}$ and generates severity-graded alarms for each identified anomaly.

Lines 1–2 initialize the alarm queue and a deduplication tracker. The main loop (Lines 3–12) evaluates each candidate in $\mathcal{R}$ to determine the alarm priority based on semantic classifi-

cation: critical priority for ”Malfunction / Uncontrolled” events, medium priority for orbital maneuvers, and low priority for orbit shape deviations.

To ensure system efficiency, Lines 8–10 implement deduplication within a time window Δt. If no recent alarm for the same satellite exists, Lines 11–13 construct a JSON payload containing the metadata, dispatch it via RESTful API endpoints, and record the event in the audit log.

Algorithm 8: Alarm Mechanism

Input : Anomaly Classification Result $\mathcal{R}$, Time Window Δt
Output: Alarm Queue $\mathcal{Q}$

1 $\mathcal{Q} \leftarrow \emptyset$;
2 $\mathcal{T} \leftarrow$ InitializeDeduplicationTracker();
3 foreach *Anomaly* $a \in \mathcal{R}$ do
4 if a.*label* = *”Malfunction / Uncontrolled”* then
5 $priority \leftarrow$ ”Critical”;
6 else if a.*label* $\in$ {*”Orbit Maneuver”*, *”Inclination Anomaly”*} then
7 $priority \leftarrow$ ”Medium”;
8 else
9 $priority \leftarrow$ ”Low”;
10 if $\neg$*IsRecentlyNotified*(a.*satellite_id*, $\mathcal{T}$, Δt) then
11 $payload \leftarrow$ ConstructJSONPayload($a, priority$);
12 DispatchAlarm($payload$, API_Endpoint);
13 $\mathcal{T}$.Record(a.satellite_id, CurrentTime());
14 return $\mathcal{Q}$;

4.5.2 Prediction Mechanism

Algorithm 9 describes the prediction mechanism that forecasts future orbital states and identifies potential anomalies. The procedure operates on historical TLE sequences to project trajectories over a prediction horizon H using statistical smoothing techniques.

Lines 1–2 prepare the input by extracting temporal sequences of orbital elements for each satellite. Instead of computationally expensive model training, the mechanism adopts an Exponentially Weighted Moving Average (EWMA) approach defined by the smoothing factor α. The main loop starting at Line 3 iterates through each satellite. Inside the loop, Line 4 applies EWMA smoothing to the historical data to reduce noise, and Line 5 extrapolates the smoothed trend to forecast the trajectory over the horizon H.

Lines 6–7 evaluate the predicted trajectories (X_{pred}) against operational thresholds to compute risk scores. If a risk score exceeds the threshold θ_{risk}, a proactive alert is generated in Line 8, enabling preventive measures before anomalies manifest.

Algorithm 9: Prediction Mechanism (EWMA-Based)

Input : Historical TLE Sequences $\mathcal{H}$, Prediction Horizon H, Smoothing Factor α, Risk Threshold θ_{risk}
Output: Predicted Trajectories $\mathcal{P}$, Risk Assessments $\mathcal{S}$

1 $\mathcal{S}_{seq} \leftarrow$ ExtractTemporalSequences($\mathcal{H}$);
2 $\mathcal{P} \leftarrow \emptyset$;
3 foreach *Satellite* $s \in \mathcal{S}_{seq}$ do
4 $\quad X_{smoothed} \leftarrow$ ApplyEWMA(s, α);
5 $\quad X_{pred} \leftarrow$ ExtrapolateTrend($X_{smoothed}, H$);
6 $\quad risk\_score \leftarrow$ EvaluateRisk(X_{pred}, OperationalThresholds);
7 $\quad$ if $risk\_score > \theta_{risk}$ then
8 $\quad\quad$ GenerateProactiveAlert($s, risk\_score, X_{pred}$);
9 $\quad \mathcal{P}$.Append(X_{pred});
10 return $\mathcal{P}, \mathcal{S}$;

4.5.3 Logging Mechanism

Algorithm 10 outlines the logging mechanism that records system events and detection results for audit and analysis. The procedure captures metadata at each stage of the deception defense pipeline.

Lines 1–2 initialize the log storage and define log level hierarchies (DEBUG, INFO, WARNING, CRITICAL). The main logging loop in Lines 3–10 processes events from the detection pipeline. For each event, Lines 4–6 construct a structured log entry containing timestamp, event type, metadata (input sources, model configurations, detection scores, classification results), and performance metrics.

Lines 7–8 apply log level filtering and retention policies, ensuring only relevant entries are stored while managing storage requirements. Line 9 persists the log entry to queryable storage, and Line 10 optionally forwards critical events to external monitoring platforms for real-time aggregation and visualization.

Algorithm 10: Logging Mechanism

Input : System Events $\mathcal{E}$, Log Level L, Retention Policy $\mathcal{R}$
Output: Log Entries $\mathcal{L}$

$\mathcal{L} \leftarrow \emptyset$;
$LogLevels \leftarrow \{\text{DEBUG}, \text{INFO}, \text{WARNING}, \text{CRITICAL}\}$;
foreach *Event* $e \in \mathcal{E}$ do
    $timestamp \leftarrow \text{CurrentTime}()$;
    $metadata \leftarrow \text{ExtractMetadata}(e)$;
    $log\_entry \leftarrow \text{ConstructLogEntry}(timestamp, e.\text{type}, metadata)$;
    if *ShouldLog*$(log\_entry, L, \mathcal{R})$ then
        $\mathcal{L}$.Append(log_entry);
        PersistToStorage(log_entry);
        if $log\_entry.level = CRITICAL$ then
            ForwardToMonitoring(log_entry);
return $\mathcal{L}$;

4.6 Simulation

The simulation module establishes a comprehensive pipeline to validate the deception defense framework. It consists of three distinct phases: (1) synthetic data generation and forgery injection, (2) threshold optimization via parameter scanning, and (3) robustness stress testing under varying attack intensities.

Algorithm 11 outlines the synthetic data generation process that leverages the trained GMM model to produce high-fidelity samples representing nominal orbital behaviors.

Lines 1–2 initialize the output collection and load the trained GMM model. The main generation loop in Lines 3–9 produces N synthetic samples. For each sample, Line 4 selects a GMM component proportionally to its mixture weight π_k. Line 5 samples a feature vector from the corresponding multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Line 6 applies the inverse standardization transformation to convert the synthetic features back to physical orbital elements. Line 7 formats the result as a standard TLE record with realistic epoch timestamps, and Line 8 appends metadata for ground-truth validation.

Algorithm 11: Synthetic TLE Data Generation

Input : Trained GMM Model M, Number of Samples N, Scaler $\mathcal{S}^{-1}$

Output: Synthetic TLE Dataset $\mathcal{D}_{syn}$, Ground-Truth Labels $\mathcal{L}_{gt}$

1 $\mathcal{D}_{syn} \leftarrow \emptyset$;

2 $M \leftarrow$ LoadTrainedGMM();

3 for $i = 1$ ***to*** N do

4 $\quad k \leftarrow$ SelectComponent(M.weights);

5 $\quad \mathrm{x}_{syn} \leftarrow$ SampleFromGaussian($M.\boldsymbol{\mu}_k, M.\boldsymbol{\Sigma}_k$);

6 $\quad \mathrm{x}_{physical} \leftarrow \mathcal{S}^{-1}(\mathrm{x}_{syn})$;

7 $\quad TLE_{record} \leftarrow$ FormatTLE($\mathrm{x}_{physical}$, epoch, catalog_num);

8 $\quad \mathcal{D}_{syn}$.Append(TLE_{record}, metadata);

9 return $\mathcal{D}_{syn}, \mathcal{L}_{gt}$;

Algorithm 12 describes the forgery injection mechanism ("Controlled Perturbation"). This step enables systematic evaluation of detection sensitivity by introducing mathematically defined deviations.

Lines 1–2 initialize the forgery pattern library. The injection loop in Lines 3–8 processes each sample. Line 4 selects a forgery type from predefined patterns (e.g., inclination shifts, eccentricity increases). Lines 5–6 apply the perturbation function $\delta(\cdot)$ to modify features. Line 7 updates the ground-truth label, ensuring that the subsequent evaluation phase has precise references for classification calibration.

Algorithm 12: Forgery Injection (Controlled Perturbation)

Input : Nominal Samples $\mathcal{D}_{nom}$, Anomaly Patterns $\mathcal{P}$, Injection Ratio ρ

Output: Anomaly-Injected Dataset $\mathcal{D}_{anom}$, Ground-Truth Labels $\mathcal{L}_{gt}$

1 $\mathcal{D}_{anom} \leftarrow \emptyset$;

2 $N_{inject} \leftarrow \lfloor |\mathcal{D}_{nom}| \times \rho \rfloor$;

3 for $i = 1$ ***to*** N_{inject} do

4 $\quad sample \leftarrow \mathcal{D}_{nom}[i]$;

5 $\quad pattern \leftarrow$ SelectAnomalyPattern($\mathcal{P}$);

6 $\quad \mathrm{x}_{perturbed} \leftarrow$ ApplyPerturbation($sample, pattern.\delta(\cdot)$);

7 $\quad label \leftarrow pattern.$type;

8 $\quad \mathcal{D}_{anom}$.Append($\mathrm{x}_{perturbed}, label, pattern.$params);

9 return $\mathcal{D}_{anom}, \mathcal{L}_{gt}$;

To bridge the gap between continuous forgery scores and binary classification, Algorithm 13 implements the *Threshold Optimization* and *Classification Calibration* phases.

This process involves calculating forgery scores (e.g., Negative Log-Likelihood for GMM or Reconstruction Error for Autoencoders) for the mixed dataset. Lines 3–7 perform *Threshold Scanning*, iterating through percentile-based thresholds τ from the validation set. For each candidate threshold, Line 5 classifies samples, and Line 6 computes precision, recall, and F1-score. Line 8 selects the optimal threshold τ_{opt} that maximizes the F1-score. Finally, Line 9 executes the *Robustness Stress Test*, outputting the calibrated performance metrics on the test set to verify the system's defense capability against the specific forgery patterns injected in Phase 2.

Algorithm 13: Threshold Optimization and Robustness Evaluation

Input : Injected Dataset $\mathcal{D}_{anom}$, Labels $\mathcal{L}_{gt}$, Model $\mathcal{M}$

Output: Optimal Threshold τ_{opt}, Performance Metrics Φ

S $\leftarrow$ CalculateAnomalyScores($\mathcal{D}_{anom}, \mathcal{M}$)

$T_{candidates} \leftarrow$ GeneratePercentiles(S, $[90, \ldots, 99.9]$) $\tau_{opt} \leftarrow 0$, Score$_{max} \leftarrow 0$;

for $\tau \in T_{candidates}$ do

    $\hat{y} \leftarrow \mathbb{I}(\text{S} > \tau)$ $F1_\tau \leftarrow$ ComputeF1($\mathcal{L}_{gt}, \hat{y}$);

    if $F1_\tau > \mathit{Score}_{max}$ then

        Score$_{max} \leftarrow F1_\tau$;

        $\tau_{opt} \leftarrow \tau$;

$\Phi \leftarrow$ EvaluateRobustness($\mathcal{D}_{anom}, \mathcal{L}_{gt}, \tau_{opt}$) return τ_{opt}, Φ;

Chapter 5 Results and Analysis

This chapter presents a comprehensive evaluation of the proposed ODDS. Owing to the inherent class imbalance characteristic of satellite telemetry data, conventional metrics such as accuracy and F1-score prove inadequate for performance characterization. Accordingly, this study emphasizes Precision and the False Positive Rate (FPR) as the primary indicators of operational reliability and system robustness.

5.1 Moving Baseline Strategy

A moving baseline strategy is utilized to assess stability in dynamic environments, separating transient variances from persistent forgeries or deception via a sliding historical window. Based on evidence that extended windows (e.g., 10 days) yield superior accuracy in satellite orbit prediction [57], this study investigates performance across two temporal scales: a 5-day (short-term) and a 10-day (long-term) baseline.

5.1.1 5-Day Baseline

The 5-day baseline configuration provides high sensitivity to rapid state transitions, making it suitable for detecting immediate operational deviations. Figure 5.1 presents the monthly forgery rate trend for December 2025, revealing the temporal dynamics of deception defense under this short-term baseline. The system exhibits a mean forgery rate of 2.94%, with notable peaks indicating periods of heightened forgery activity. The highest forgery rate occurs on Day 26, reaching approximately 10.5%, while multiple days (Days 5, 10, 11, 22, 28, and 31) show zero deception candidates, indicating stable operational periods.

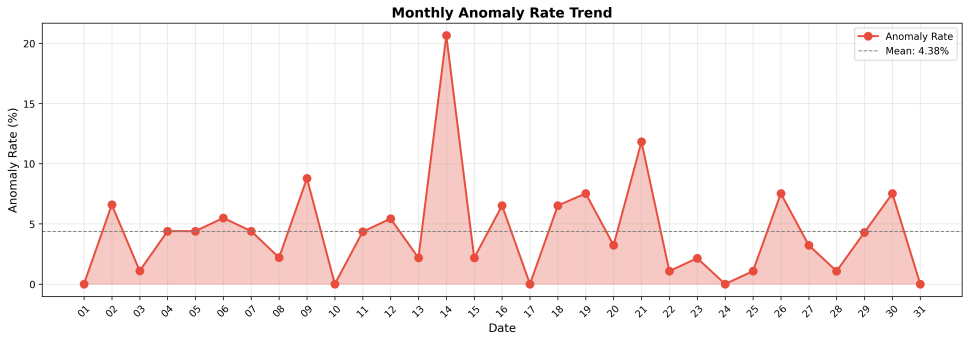

Figure 5.1: Monthly Forgery Rate: 5-day Sliding Baseline

Figure 5.2 illustrates detailed detection results obtained using a 5-day window over December 15–17, 2025, demonstrating the system's responsiveness to immediate state changes:

- December 15: Two deception candidates detected ($N = 92$).
- December 16: Six deception candidates detected, capturing a pronounced deviation in the telemetry stream.
- December 17: Zero deception candidates detected, indicating a rapid return to the nominal operational state.

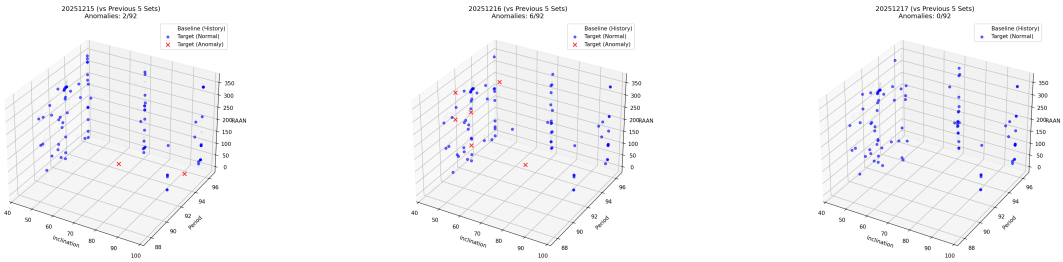

Figure 5.2: Detection Results: 5-day Sliding Window (Sensitivity).

5.1.2 10-Day Baseline

The 10-day baseline configuration extends the temporal context, functioning as a low-pass filter that effectively smooths out transient noise. Figure 5.3 displays the monthly forgery rate trend for December 2025 under this long-term baseline. The system demonstrates a mean

forgery rate of 4.38%, which is higher than the 5-day baseline, reflecting the broader temporal context that captures more persistent forgeries. The highest forgery rate occurs on Day 14, reaching approximately 20.8%, indicating a significant operational deviation that persisted across the extended baseline window. Days with zero deception candidates (Days 1, 10, 17, 24, and 31) represent periods of sustained nominal operation.

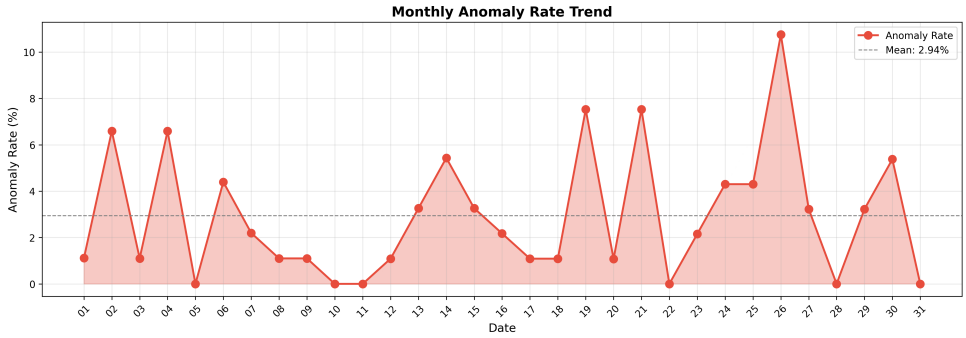

Figure 5.3: Monthly Forgery Rate: 10-day Sliding Baseline

Figure 5.4 presents detailed detection results obtained using a 10-day window over December 15–17, 2025, illustrating the system's stability-oriented detection behavior:

- December 15: Three deception candidates detected.
- December 16: Two deception candidates detected (in contrast to six detected by the 5-day model).
- December 17: One deception candidate detected.

Comparison of these baseline configurations reveals a fundamental trade-off: the 5-day window provides high responsiveness to immediate events, whereas the 10-day window offers stability by filtering out ephemeral variations and focusing on persistent trends. This observation aligns with findings in satellite orbit prediction research, where extended temporal windows (10-day baselines) have been shown to improve prediction accuracy and capture more persistent patterns compared to shorter windows [57]. The monthly trend analysis further demonstrates that the 10-day baseline captures a broader range of forgeries or deception candidates (mean rate 4.38% vs. 2.94%), while the 5-day baseline exhibits more pronounced daily fluctuations.

Optimal window selection depends on specific mission requirements, balancing rapid response against robust trend monitoring.

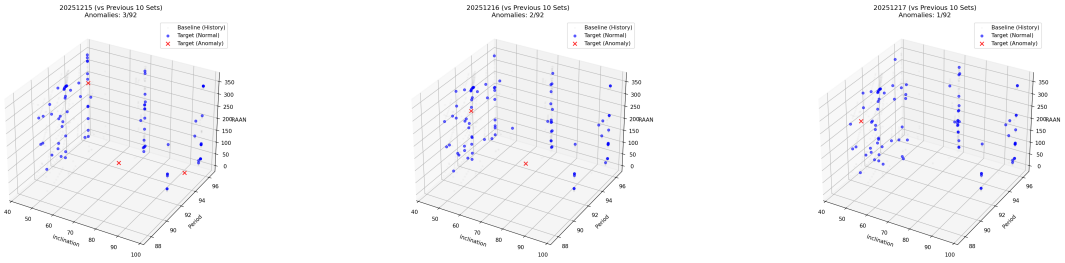

Figure 5.4: Detection Results: 10-day Sliding Window (Stability).

5.2 Orbital Parameter Displacement Analysis

To evaluate the framework's vulnerability to adversarial manipulation, this section investigates the displacement of orbital parameters under a TLE data deception attack. Figures 5.5, 5.6, and 5.7 visualize the systematic shift of satellite states within a 3D feature space comprising Inclination (i), Period (P), and Right Ascension of the Ascending Node ($RAAN$). By utilizing the 10-day sliding window as a *Baseline (History)*, the experiment demonstrates the attacker's attempt to navigate the target satellite into a falsified but statistically plausible region.

In the "Original Detection" phase (left panels), the attack identifies the Target (Normal) satellite and calculates a displacement vector, visualized as the orange dashed Route. This path represents the intended trajectory from the legitimate orbital state to the Corrected Position (Green Circle). Observation on December 15 reveals a significant initial displacement that moves the target away from its dense historical cluster. By December 16 and 17, the attack stabilizes the satellite in these "deceived" coordinates. The "After Displacement" phase (right panels) confirms that the Target Deception (Teal dots) successfully resides in a region of the parameter space that is distinct from its original position while attempting to blend into the fringes of the historical baseline.

The analysis indicates that the deception is a targeted relocation rather than a stochastic injection of noise. By maintaining the falsified data within a physically plausible range—particu-

larly evident in the $RAAN$ and Period dimensions—the attack aims to bypass simple threshold-based detectors. Such a persistent shift in the orbital feature space represents a critical threat to SSA systems, as it can induce track loss, identity mis-association, and erroneous collision warnings, ultimately forcing operators to act upon a compromised reality.

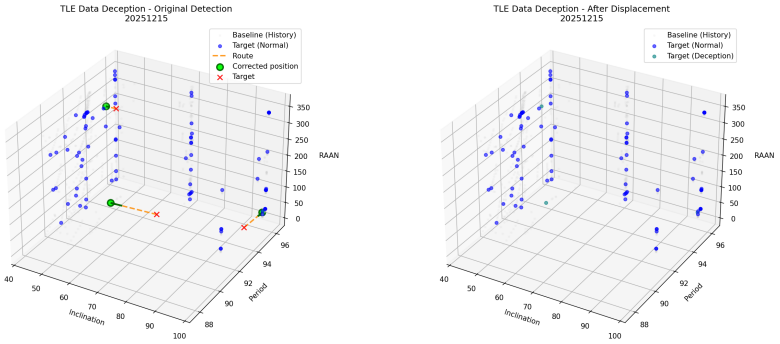

Figure 5.5: Deception Results: Dec 15, 2025

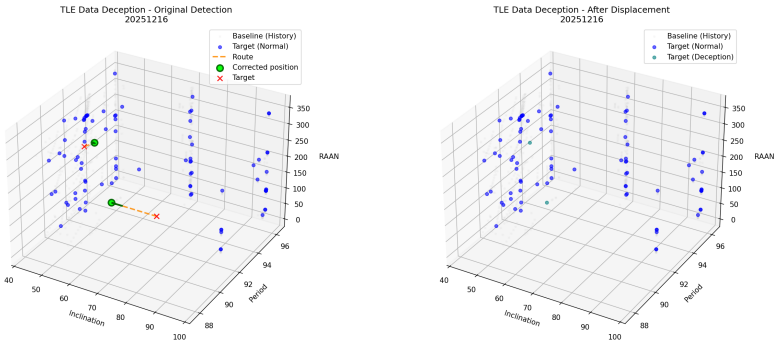

Figure 5.6: Deception Results: Dec 16, 2025

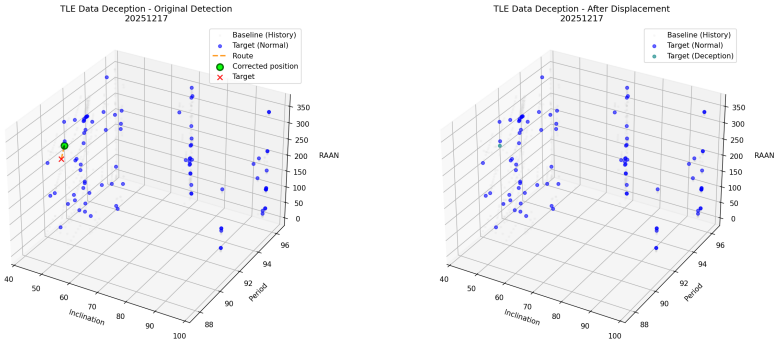

Figure 5.7: Deception Results: Dec 17, 2025

5.3 Defense Evaluation

The viability of ODDS as a cybersecurity defense framework is further validated through its adversarial resilience. This section presents the quantitative performance of the ”Forged-into-Nominal” experiment and discusses its operational implications for TLE deception attack mitigation.

5.3.1 Quantitative Results of Deception Defense

The effectiveness of ODDS was tested by attempting to ”wash” forged or manipulated signals into nominal-looking telemetry to evade detection. The aggregate benchmarking results are summarized in Table 5.1.

Table 5.1: Deception and Defense Benchmarking Summary

Metric	Value
Total Samples Processed	6,186
Forgeries Replaced (Attack Attempts)	120
Successfully Detected by ODDS (Defense)	93
Evaded Re-detection (Evasion)	27
Evasion Rate (Attack Success)	22.5%
Defense Rate (Defense Recall)	77.5%

The experiment shows that while the attacker achieved a 22.5% evasion rate by meticulously crafting TLE parameters to match the nominal distribution, ODDS intercepted 77.5% of these high-fidelity forgeries. In an adversarial context, this 77.5% defense rate demonstrates that the multi-layered consistency checks in ODDS—particularly the GMM’s temporal constraints—remain sensitive to the underlying physical-statistical mismatch even when data appears statistically normal in isolation.

5.3.2 TLE Deception Attack Defense

A critical vulnerability in the current space ecosystem is the unauthenticated transmission of the TLE format. Without cryptographic verification, TLE data is inherently susceptible to Man-in-the-Middle (MitM) attacks. If a ground station ingests spoofed data, the SGP4 propagator will generate incorrect look angles, leading to physical antenna misalignment and a persistent Loss of Signal (LOS). ODDS serves as a pre-ingestion validation layer, blocking forged or manipulated data before it contaminates the tracking loop. From an adversarial perspective, ODDS significantly raises the Work Factor of an attack: to bypass the system, an attacker must craft parameters that satisfy both the learned probability distribution and kinematic consistency (GMM), while simultaneously avoiding isolation as forgeries or manipulated samples (IF). This dual-domain requirement increases the complexity and resources necessary for a successful breach, thereby enhancing the operational resilience of the ground segment.

The forged-into-nominal scenario represents an attempt to mask off-nominal maneuvers as benign behavior. Despite these masking efforts, ODDS retains a discriminative capability because its GMM layer prioritizes temporal consistency between successive propagated states over simple feature histograms, while the IF and HDBSCAN layers identify residual patterns that nominal process noise cannot explain. These findings confirm that ODDS does not rely solely on static distributions but leverages cross-epoch and cross-domain consistency checks to detect ”washed” forged or manipulated behavior, providing a robust second line of defense for satellite telemetry.

Chapter 6 Conclusion and Future Work

This chapter summarizes the key findings and technical contributions of this thesis, emphasizing its significance in safeguarding satellite ground stations against orbital deception. By addressing the security vulnerabilities within TLE-based tracking infrastructures, this work establishes a critical framework for trusted orbital data assurance in an increasingly adversarial space domain.

6.1 Conclusion

This thesis conducted a rigorous investigation into *TLE data deception attacks targeting satellite ground stations*. Given that modern ground infrastructures rely on TLE data for mission-critical operations—including antenna pointing and orbit propagation—the integrity of these parameters is a paramount security concern. This research exposed how adversaries can exploit orbit prediction uncertainties to inject counterfeit yet physically plausible information to mislead ground-based systems.

The primary contributions and validated capabilities of this research are summarized below:

1. Adversarial Threat Modeling

   This work formally characterized the manipulation of orbital element streams within ground station pipelines. By identifying critical vulnerabilities in the TLE ingestion workflow, we demonstrated that the lack of cryptographic authentication in legacy formats facilitates Man-in-the-Middle (MitM) injections, potentially leading to a catastrophic Loss of Signal (LOS).

2. Stealthy Deception Synthesis

   A key technical contribution is the development of a "forged-into-nominal" deception module. By "washing" manipulated orbital deviations into historically consistent patterns, we evaluated ODDS against high-fidelity forgeries. In the experiment, ODDS achieved a 77.5% defense rate (attackers 22.5% evasion), demonstrating that temporal-kinematic

consistency checks effectively reduce the evasion space of spoofing attacks that bypass conventional threshold-based detection.

3. ODDS Defense Validation

   This thesis introduced ODDS, a multi-layered machine learning framework leveraging GMM, Isolation Forest, and HDBSCAN. Experimental results confirm that even against sophisticated forgeries, ODDS effectively maintains defense by prioritizing temporal-kinematic consistency. This architecture significantly elevates the Work Factor for adversaries, forcing them to solve complex multi-objective optimization problems to maintain orbital plausibility.

In summary, this research provides a security-oriented foundation for defending against orbital deception. By functioning as a pre-ingestion validation layer, the proposed system ensures that tracking loops remain resilient even when the data source is compromised, contributing to the broader mission of ensuring trustworthy space operations.

6.2 Future Work

To further advance the protection of space infrastructures against adaptive deception threats, future research will focus on three strategic directions:

- Context-Aware Adaptation

  Future iterations of ODDS could incorporate adaptive learning rates to better distinguish between legitimate maneuvers and adversarial spoofing. Integrating context-aware sensors would allow the system to reduce evasion rates by cross-referencing telemetry with independent physical observations (e.g., optical or ranging data).

- Federated Resilience

  Deploying lightweight defense models at the edge of satellite-ground architectures can enhance resilience without exposing sensitive datasets. A federated learning approach across multiple ground stations would enable a collective intelligence to identify global spoofing campaigns while preserving the data privacy of individual operators.

- Zero-Trust Infrastructure

  Moving beyond passive defense, a zero-trust verification architecture should be explored. This entails combining cryptographic integrity with continuous behavioral consistency checks, ensuring that no orbital update is trusted by default. Such an ecosystem would mitigate risks arising from insider threats and supply-chain compromises.

These directions aim to transition satellite ground station security from reactive monitoring toward an active, deception-aware, and intrinsically secure orbital defense ecosystem.

References

[1] Carmen Pardini and Luciano Anselmo. “Evaluating the impact of space activities in low earth orbit”. In: *Acta Astronautica* 184 (2021), pp. 11–22. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2021.03.030. URL: https://www.sciencedirect.com/science/article/pii/S0094576521001430.

[2] John A Kennewell and Ba-Ngu Vo. “An overview of space situational awareness”. In: *Proceedings of the 16th International Conference on Information Fusion*. 2013, pp. 1029–1036.

[3] Zhixin Guo et al. *SpaceTrack-TimeSeries: Time Series Dataset towards Satellite Orbit Analysis*. 2025. arXiv: 2506.13034 [astro-ph.EP].

[4] Rafal Graczyk, Marcus Voelp, and Paulo Esteves-Verissimo. *EphemeriShield – defence against cyber-antisatellite weapons*. 2021. arXiv: 2101.12620 [cs.CR]. URL: https://arxiv.org/abs/2101.12620.

[5] Jos Wigchert, Savio Sciancalepore, and Gabriele Oligeri. *Detection of Aerial Spoofing Attacks to LEO Satellite Systems via Deep Learning*. 2024. arXiv: 2412.16008 [cs.CR].

[6] Zhong-Hua Pang et al. “Security of networked control systems subject to deception attacks: a survey”. In: *International Journal of Systems Science* 53.16 (2022), pp. 3577–3598. DOI: 10.1080/00207721.2022.2143735. eprint: https://doi.org/10.1080/00207721.2022.2143735. URL: https://doi.org/10.1080/00207721.2022.2143735.

[7] Qirui Zhang et al. “Optimal Stealthy Deception Attack Against Cyber-Physical Systems”. In: *IEEE Transactions on Cybernetics* 50.9 (2020), pp. 3963–3972. DOI: 10.1109/TCYB.2019.2912622.

[8] Ning Ruan et al. “Edge AI for Earth Observation”. In: *IEEE Internet Computing* 29.3 (2025), pp. 31–40. DOI: 10.1109/MIC.2025.3587325.

[9] Min Wang, Donghua Zhou, and Maoyin Chen. “Anomaly Monitoring of Nonstationary Processes With Continuous and Two-Valued Variables”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53.1 (2023), pp. 49–58. DOI: 10.1109/TSMC.2022.3167838.

[10] Seif-Eddine Benkabou et al. “Local Anomaly Detection for Multivariate Time Series by Temporal Dependency Based on Poisson Model”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.11 (2022), pp. 6701–6711. DOI: 10.1109/TNNLS.2021.3083183.

[11] Takehisa Yairi et al. “A Data-Driven Health Monitoring Method for Satellite Housekeeping Data Based on Probabilistic Clustering and Dimensionality Reduction”. In: *IEEE Transactions on Aerospace and Electronic Systems* 53.3 (2017), pp. 1384–1401. DOI: 10.1109/TAES.2017.2671247.

[12] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. “A survey on unsupervised outlier detection in high-dimensional numerical data”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 5.5 (2012), pp. 363–387. DOI: https://doi.org/10.1002/sam.11161. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sam.11161. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11161.

[13] L.A. Zhang, K. Langeland, and J. Tran. *Artificial Intelligence and Machine Learning for Space Domain Awareness: Characterizing the Impact on Mission Effectiveness*. Research report (Rand Corporation). RAND Corporation, 2025. ISBN: 9781977414472. URL: https://books.google.com.tw/books?id=Nb4J0QEACAAJ.

[14] David Vallado and Paul Crawford. “SGP4 Orbit Determination”. In: Aug. 2008. ISBN: 978-1-62410-001-7. DOI: 10.2514/6.2008-6770.

[15] D.A. Reynolds and Richard Rose. “Robust text-independent speaker identification using Gaussian Mixture speaker models”. In: *Speech and Audio Processing, IEEE Transactions on* 3 (Feb. 1995), pp. 72 –83. DOI: 10.1109/89.365379.

[16] Fei Tony Liu, Kai Ting, and Zhi-Hua Zhou. “Isolation Forest”. In: Jan. 2009, pp. 413 –422. DOI: 10.1109/ICDM.2008.17.

[17] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. “Density-Based Clustering Based on Hierarchical Density Estimates”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172. ISBN: 978-3-642-37456-2. DOI: 10.1007/978-3-642-37456-2_14.

[18] Anqi Lang and Yu Jiang. “Orbit Determination for Continuously Maneuvering Starlink Satellites Based on an Unscented Batch Filtering Method”. In: *Sensors* 25.13 (2025). ISSN: 1424-8220. DOI: 10.3390/s25134079. URL: https://www.mdpi.com/1424-8220/25/13/4079.

[19] Emilian Croitoru. “Satellite Tracking Using Norad Two-Line Element Set Format”. In: *Scientific Research and Education in the Air Force* (2016). DOI: 10.19062/2247-3173.2016.18.1.58.

[20] Sajjad Kazemi et al. “Orbit determination for space situational awareness: A survey”. In: *Acta Astronautica* 222 (2024), pp. 272–295. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2024.06.015. URL: https://www.sciencedirect.com/science/article/pii/S0094576524003308.

[21] Darlan Noetzold et al. “Enhancing Infrastructure Observability: Machine Learning for Proactive Monitoring and Anomaly Detection”. In: *Journal of Internet Services and Applications* 15 (Oct. 2024), pp. 508–522. DOI: 10.5753/jisa.2024.4509.

[22] Wei Dong and Zhao Chang-yin. “An Accuracy Analysis of the SGP4/SDP4 Model”. In: *Chinese Astronomy and Astrophysics* 34.1 (2010), pp. 69–76. ISSN: 0275-1062. DOI: https://doi.org/10.1016/j.chinastron.2009.12.009. URL: https://www.sciencedirect.com/science/article/pii/S0275106209001404.

[23] Christopher F Wildt. “Accuracy in orbital propagation: A comparison of predictive software models”. In: (2017).

[24] Creon Levit and William Marshall. “Improved orbit predictions using two-line elements”. In: *Advances in Space Research* 47.7 (2011), pp. 1107–1115. ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2010.10.017. URL: https://www.sciencedirect.com/science/article/pii/S0273117710006964.

[25] Jinghong Liu et al. “Improving Orbit Prediction of the Two-Line Element with Orbit Determination Using a Hybrid Algorithm of the Simplex Method and Genetic Algorithm”. In: *Aerospace* 12.6 (2025). ISSN: 2226-4310. DOI: 10.3390/aerospace12060527. URL: https://www.mdpi.com/2226-4310/12/6/527.

[26] Giacomo Acciarini, Atılım Güneş Baydin, and Dario Izzo. “Closing the gap between SGP4 and high-precision propagation via differentiable programming”. In: *Acta Astronautica* 226 (2025), pp. 694–701. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2024.10.063. URL: https://www.sciencedirect.com/science/article/pii/S0094576524006374.

[27] Arvind Mukundan and Hsiang-Chen Wang. “Simplified Approach to Detect Satellite Maneuvers Using TLE Data and Simplified Perturbation Model Utilizing Orbital Element Variation”. In: *Applied Sciences* 11.21 (2021). ISSN: 2076-3417. DOI: 10.3390/app112110181. URL: https://www.mdpi.com/2076-3417/11/21/10181.

[28] Yun-Ju Chiu. “From Circular to Elliptical: Exploring the Sequence and Related Illustrations of Kepler’s Laws of Planetary Motion”. Chinese. In: *Chinese Physics Education* 20.1 (2019), pp. 1–13. ISSN: 1998-7544. DOI: 10.6212/CPE.201907_20(1).0001.

[29] Jiri Silha Dracek Frantisek Dracek and Roman Durikovic. “Anomaly Detection from TLE Data”. In: *2023 Communication and Information Technologies (KIT)*. 2023, pp. 1–6. DOI: 10.1109/KIT59097.2023.10297051.

[30] Jingrui Zhang et al. “LEO Mega Constellations: Review of Development, Impact, Surveillance, and Governance”. In: *Space: Science and Technology* 2022 (July 2022), pp. 1–17. DOI: 10.34133/2022/9865174.

[31] Federica Massimi, Pasquale Ferrara, and Francesco Benedetto. “Deep Learning Methods for Space Situational Awareness in Mega-Constellations Satellite-Based Internet of Things Networks”. In: *Sensors* 23.1 (2023). ISSN: 1424-8220. DOI: 10.3390/s23010124. URL: https://www.mdpi.com/1424-8220/23/1/124.

[32] WU Xiaohe. “Research on satellite orbit prediction technology based on LSTM”. In: *Journal of Cybersecurity* 2.4, 18 (2024), pp. 18–28. DOI: 10.20172/j.issn.2097-3136.240402. URL: https://www.journalofcybersec.com/EN/10.20172/j.issn.2097-3136.240402.

[33] Jiayi Tang et al. “Federated-Learning-Based Strategy for Enhancing Orbit Prediction of Satellites”. In: *Mathematics* 13.8 (2025). ISSN: 2227-7390. DOI: 10.3390/math13081312. URL: https://www.mdpi.com/2227-7390/13/8/1312.

[34] Abebe Diro et al. “Anomaly detection for space information networks: A survey of challenges, techniques, and future directions”. In: *Computers and Security* 139 (2024), p. 103705. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2024.103705. URL: https://www.sciencedirect.com/science/article/pii/S0167404824000063.

[35] Zhong-Hua Pang and Guo-Ping Liu. “Design and Implementation of Secure Networked Predictive Control Systems Under Deception Attacks”. In: *IEEE Transactions on Control Systems Technology* 20.5 (2012), pp. 1334–1342. DOI: 10.1109/TCST.2011.2160543.

[36] Derui Ding et al. “Security Control for Discrete-Time Stochastic Nonlinear Systems Subject to Deception Attacks”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.5 (2018), pp. 779–789. DOI: 10.1109/TSMC.2016.2616544.

[37] James Pavur and Ivan Martinovic. “On Detecting Deception in Space Situational Awareness”. In: *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. ASIA CCS ’21. Virtual Event, Hong Kong: Association for Computing Machinery, 2021, pp. 280 – 291. ISBN: 9781450382878. DOI: 10.1145/3433210.3453081. URL: https://doi.org/10.1145/3433210.3453081.

[38] J.F. Olivier and T.M. Louw. “Time-constrained Gaussian mixture model for clustering multi-modal chemical process data”. In: *IFAC-PapersOnLine* 58.25 (2024). 3rd Control Conference Africa CCA 2024, pp. 108–113. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2024.10.246. URL: https://www.sciencedirect.com/science/article/pii/S2405896324020214.

[39] Bo Zong et al. “Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection”. In: *International Conference on Learning Representations*. 2018. URL: https://api.semanticscholar.org/CorpusID:51805340.

[40] J. MacQueen. “Some Methods for Classification and Analysis of Multivariate Observations”. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Held June 21–July 18, 1965 and December 27, 1965–January 7, 1966. Berkeley, CA: University of California Press, 1967, pp. 281–297. URL: https://digitalassets.lib.berkeley.edu/math/ucb/text/05_1_1967.pdf.

[41] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[42] Angela A. Sodemann, Matthew P. Ross, and Brett J. Borghetti. “A Review of Anomaly Detection in Automated Surveillance”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42 (2012), pp. 1257–1272. URL: https://api.semanticscholar.org/CorpusID:15466712.

[43] MU Jingjing HE Zhangming ZHOU Xuanying WEI Juhui WANG Jiongqi. “Anomaly Detection for Satellite Power System Based on Gaussian Mixture Model”. In: *Aerospace control and application* 48.4 (2022), pp. 104–114. ISSN: 1674-1579. DOI: 10.3969/j.issn.1674-1579.2022.04.013.

[44] Ahmed Adam et al. “A Comparative Analysis of Anomaly Detection Techniques for Battery Telemetry Data in Low Earth Orbit Remote Sensing Satellites”. In: *International Journal of Prognostics and Health Management* 16 (Sept. 2025). DOI: 10.36001/ijphm.2025.v16i2.4273.

[45] Haoyue Zhang, Chunmei Zhao, and Zhengbin He. “Robust Gaussian Mixture Model for Maneuver Detection Using TLE Data”. In: *Geomatics and Information Science of Wuhan University* (2024). DOI: 10.13203/j.whugis20230360. URL: http://ch.whu.edu.cn/article/doi/10.13203/j.whugis20230360.

[46] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation-Based Anomaly Detection”. In: *ACM Trans. Knowl. Discov. Data* 6.1 (Mar. 2012). ISSN: 1556-4681. DOI: 10.1145/2133360.2133363. URL: https://doi.org/10.1145/2133360.2133363.

[47] Yang Cao et al. “Anomaly Detection Based on Isolation Mechanisms: A Survey”. In: *Machine Intelligence Research* 22.5 (Sept. 2025), pp. 849–865. ISSN: 2731-5398. DOI: 10.1007/s11633-025-1554-4. URL: http://dx.doi.org/10.1007/s11633-025-1554-4.

[48] Hongzuo Xu et al. “Deep Isolation Forest for Anomaly Detection”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.12 (Dec. 2023), pp. 12591–12604. ISSN: 2326-3865. DOI: 10.1109/tkde.2023.3270293. URL: http://dx.doi.org/10.1109/TKDE.2023.3270293.

[49] Clemens Schefels, Leonard Schlag, and Markus Steinbach. “To Catch Them All: A Generic Approach for Pattern Detection in Time Series Satellite Telemetry Data”. In: Jan. 2021.

[50] Yakun Wang et al. “A Deep Learning Anomaly Detection Framework for Satellite Telemetry with Fake Anomalies”. In: *International Journal of Aerospace Engineering* 2022 (Jan. 2022), pp. 1–9. DOI: 10.1155/2022/1676933.

[51] Xingxing Li et al. “Quality monitoring of real-time PPP service using isolation forest-based residual anomaly detection”. In: *GPS Solutions* 28.3 (2024), p. 118. ISSN: 1521-1886. DOI: 10.1007/s10291-024-01657-z. URL: https://doi.org/10.1007/s10291-024-01657-z.

[52] Ricardo J. G. B. Campello et al. “Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection”. In: *ACM Transactions on Knowledge Discovery from Data* 10.1 (2015), pp. 1–51. DOI: 10.1145/2733381.

[53] Tat-Huy Tran, Tuan-Dung Cao, and Thi-Thu-Huyen Tran. “HDBSCAN: Evaluating the Performance of Hierarchical Clustering for Big Data”. In: *Soft Computing: Biomedical and Related Applications*. Ed. by Nguyen Hoang Phuong and Vladik Kreinovich. Cham: Springer International Publishing, 2021, pp. 273–283. ISBN: 978-3-030-76620-7. DOI: 10.1007/978-3-030-76620-7_24. URL: https://doi.org/10.1007/978-3-030-76620-7_24.

[54] C. H. A. Logan and S. Fotopoulou. “Unsupervised star, galaxy, QSO classification: Application of HDBSCAN”. In: *Astronomy & Astrophysics* 633 (2020). Section: Numerical methods and codes, A154. DOI: 10.1051/0004-6361/201936648. URL: https://doi.org/10.1051/0004-6361/201936648.

[55] Nitish Raj and Prabhat Kumar. “Leveraging HDBSCAN, LSTM and R-DTW for Proactive Detection and Collision Prediction in Maritime Traffic”. English. In: *Defence Science Journal* 75.4 (2025), pp. 490–497. URL: https://www.proquest.com/scholarly-journals/leveraging-hdbscan-lstm-r-dtw-proactive-detection/docview/3228705193/se-2.

[56] Shih-Ming Wang et al. “Application of Three-Dimensional Hierarchical Density-Based Spatial Clustering of Applications with Noise in Ship Automatic Identification System Trajectory-Cluster Analysis”. In: *Applied Sciences* 15.5 (2025). ISSN: 2076-3417. DOI: 10.3390/app15052621. URL: https://www.mdpi.com/2076-3417/15/5/2621.

[57] Junyu Chen and Chusen Lin. “Research on Enhanced Orbit Prediction Techniques Utilizing Multiple Sets of Two-Line Element”. In: *Aerospace* 10.6 (2023). ISSN: 2226-4310. DOI: 10.3390/aerospace10060532. URL: https://www.mdpi.com/2226-4310/10/6/532.

[58] Jos Wigchert, Savio Sciancalepore, and Gabriele Oligeri. “Detection of Aerial Spoofing Attacks to LEO Satellite Systems via Deep Learning”. In: *Computer Networks* 269 (2025), p. 111408. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2025.111408. URL: https://www.sciencedirect.com/science/article/pii/S1389128625003755.

[59] T. S. Kelso. *Celestrak: Satellite Two-Line Element Sets*. https://celestrak.org/NORAD/elements/starlink.txt. Accessed: 2025-10-01. 2025.

Appendix A: TLE Format Field Description

Tables A.1 and A.2 provide a field-by-field breakdown of the Two-Line Element (TLE) format using the STARLINK-1008 example.

Table A.1: TLE Field Definitions: Name and Line 1

Field	Example	Description
Satellite name	STARLINK-1008	Unique identifier following constellation name and satellite number.
Line number	1	TLE line identifier (first data line).
Catalog number	44714	NORAD satellite catalog number; unique 5-digit identifier assigned sequentially.
Classification	U	Security code: U (Unclassified), C (Classified), S (Secret).
International designator	19074B	Format YYNNNPPP: year (19=2019), launch # (074), piece (B).
Epoch time	25280.37724496	Reference time: YYDDD.DDDDDDDD format (25=2025, day 280, 0.377≈9:03 UTC). Elements valid only at this moment.
First derivative of mean motion	.00005163	$\dot{n}/2$ in rev/day^2. Positive = decay (drag), negative = raising.
Second derivative of mean motion	00000+0	Second derivative / 6 in rev/day^3; higher-order perturbations. Usually near zero.
B* drag coefficient	36508-3	SGP4 drag parameter (3.6508×10^{-3}). $B^* = \frac{1}{2}\rho_0 \frac{C_D A}{m}$ where ρ_0 = reference density, C_D = drag coeff., A = area, m = mass. Larger values = stronger drag, faster decay.
Element set number	9999	Revision number; incremented with each update.

Table A.2: TLE Field Definitions: Line 2

Field	Example	Description
Line number	2	TLE line identifier (second data line).
Catalog number	44714	Repeated NORAD catalog number for verification; must match Line 1 to ensure data integrity.
Inclination	53.0493	Orbital inclination i (0°–180°): angle between orbital and equatorial planes. $i = \cos^{-1}(h_z/\|\mathrm{h}\|)$ from $\mathrm{h} = \mathrm{r} \times \mathrm{v}$.
RAAN	154.8642	Right ascension of ascending node Ω (0°–360°): angle from vernal equinox to ascending node (equator crossing S→N). Defines orbital plane orientation.
Eccentricity	0001368	Orbital eccentricity e (0.0001368). $e = \sqrt{1 + \frac{2Eh^2}{\mu^2}}$ or $e = \frac{r_a - r_p}{r_a + r_p}$. Ranges 0 (circle) to <1 (ellipse).
Argument of perigee	87.3282	Argument of perigee ω (0°–360°): angle from ascending node to perigee in orbital plane.
Mean anomaly	272.7864	Mean anomaly M (0°–360°) at epoch. $M = n(t - t_0)$; relates to true anomaly via Kepler's equation: $M = E - e \sin E$.
Mean motion	15.06415906	Mean motion n (rev/day). From Kepler's third law: $n^2 a^3 = \mu$ ($\mu = 3.986 \times 10^{14}$ m^3/s^2). Higher n = lower altitude ($n \approx 15.04 \approx 550$ km LEO).
Revolution number	325647	Orbit count since launch/reference epoch. Useful for deception attack detection.